nVISION-i
USER MANUAL

di-soric

nVision-i Users Guide

Release 24 .2

di-soric GmbH & Co. KG

Nov 06, 2024

CONTENTS

nVision-i Benutzerhandbuch

Release 24.2.2971

Disclaimer of liability:

No liability is accepted for printing errors and mistakes that occurred during the creation of the document. The information and details in this document can be changed at any time.

INTRODUCTION

nVision-i is the di-soric development software for vision sensor devices.

There are currently two product lines:

- the **ID-600** is a stationary code reader,

- the **CS-60** is a general purpose vision sensor.

nVision-i has valuable and powerful tools for image processing and inspection. It is used to configure solutions on a di-soric vision sensor.

nVision-i helps you with a variety of vision sensor based tasks needed in the industry:

- Check the position of a part to guide handling systems or to put a vision tool in the right position.

- Identify a part based on marks, shape or other visual aspects.

- Verify a part to check if it has been built or assembled correctly.

- Measure a part's dimensions.

- Inspect a part for defects.

1.1 Program Editions

nVision-i has two main components: the **nVision-i Designer** is the rapid application development system and the **nVision-i Runtime** is the runtime part.

The **nVision-i Designer** provides a graphical user interface and a graphical programming system.

The **nVision-i Runtime** provides the machine vision functionality in a modular way and a facility to execute graphical programs. The **nVision-i Runtime** comes in two versions:

- running on a **CS-60/ID-600** vision sensor device

- running on a PC as a **CS-60/ID-600** vision sensor emulator

The **nVision-i Designer** is the program you use to create applications. It talks directly to a vision sensor (or an emulator on a PC), where the application is executed.

The vision sensor has functions for basic image processing, image analysis, blob analysis, identification, measurement and template matching.

1.2 Requirements

The **nVision-i Designer** runs on Windows 7 or higher. The **nVision-i Designer** is a 64bit application.

The **nVision-i Runtime** runs on the vision sensor device as well as on a PC in emulator mode.

System

1 GHz (Giga-Hertz) or faster, two or more cores, 64 bit prozessor or SoC (system on a chip)

Memory

RAM 4 GB (Giga-Bytes) or more

Disk

64 GB (GIga-Bytes) or more available disk space

Display

HD (high-definition, 720p) display, 9" monitor or bigger, 8 bit per color channel

Operating System (OS)

Microsoft Windows 10 or 11

1.3 Licensing

The **nVision-i Designer** works in connection with a vision sensor hardware. Both the **nVision-i Designer** as well as the vision sensor use licensed software components. The nVision-i license is provided free of charge. License keys are unique and can only be used once.

1.3.1 nVision-i

If no previous version has been installed and licensed, the license is checked when nVision-i is started for the first time.
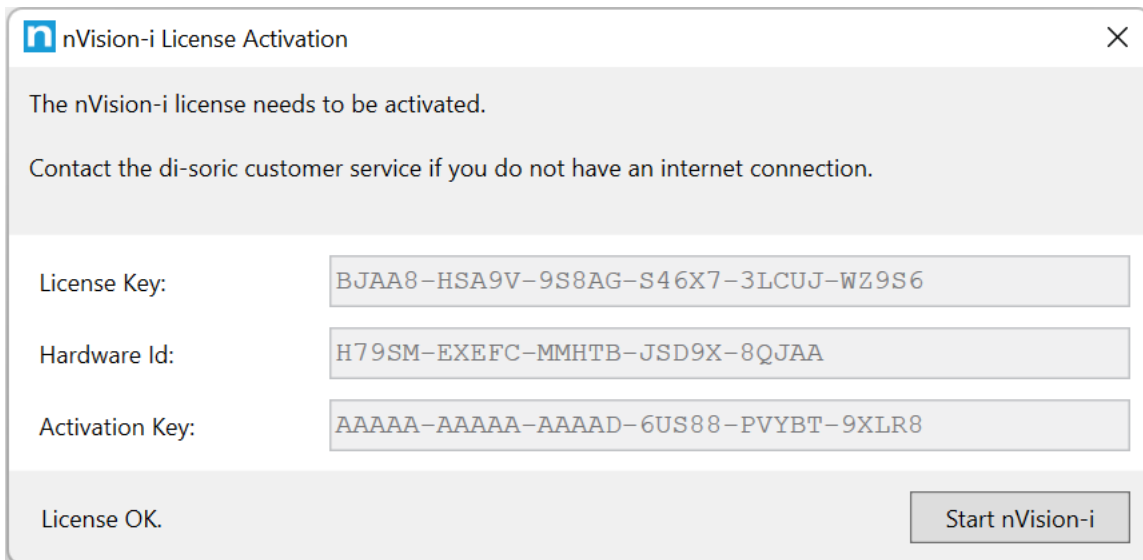
Fig. 1: The license activation dialog of nVision-i.

If the licensing was successful, nVision-i can be started. Otherwise an error message is displayed.

If you do not have Internet access or the firewall settings prevent a connection, the software can be activated by phone. In these cases, we recommend that you contact our customer service. Customer service will give you instructions and provide data for manual license activation (in this case you need to know your hardware ID).

After successful activation, nVision-i can be started.

You may receive a message from the Windows Firewall when you start nVision-i:

Fig. 2: Firewall message from nVision-i.

You must allow access via the firewall so that nVision-i can communicate with the **CS-60/ID-600** sensors.

CS-60 Vision Sensor

Several tool sets are available for the **CS-60**.

The **Basic** tool set contains the location tools **Locate Area**, **Locate Edge** and **Locate Shape**, the detection tools **Detect Brightness**, **Detect Contrast** and **Detect Area Pixels**, and the counting tools **Count Areas**, **Count Edges** and **Count Shapes**.

Optional tool sets are the **Measure** and the **ID** sets and the **ID-Pro** extension.

The **Measure** tool set contains the measurement tools **Measure Angle**, **Measure Circle**, **Measure Distance**, **Measure Point to Point** and **Measure Point to Line**.

The **ID** tool set contains the barcode tools **Locate Barcode**, **Detect Barcode** and **Count Barcodes**.

The **ID-Pro** (Direct Part Marking) extension offers extended and improved recognition of **DPM** barcodes as well as verification of 2D codes.

The **CS-60** vision sensor comes with a pre-installed license for the **Basic** toolset. Licenses for the **Measure** and/or **ID** toolsets and the **ID-Pro** extension can be purchased and installed with the **nVision-i** software.

ID-600 Stationary Code Reader

One tool set is available for the **ID-600**.

The **ID** tool set contains the barcode tools **Locate Barcode**, **Detect Barcode** and **Count Barcodes**.

The **ID-Pro** (Direct Part Marking) extension offers extended and improved recognition of **DPM** barcodes as well as verification of 2D codes.

The **ID-600** staionary code reader comes with a pre-installed license for the **ID** toolset. A license for the **ID-Pro** extension can be purchased and installed with the **nVision-i** software.

License Management

The current license of a **CS-60/ID-600** vision sensor can be displayed with the software **nVision-i**.
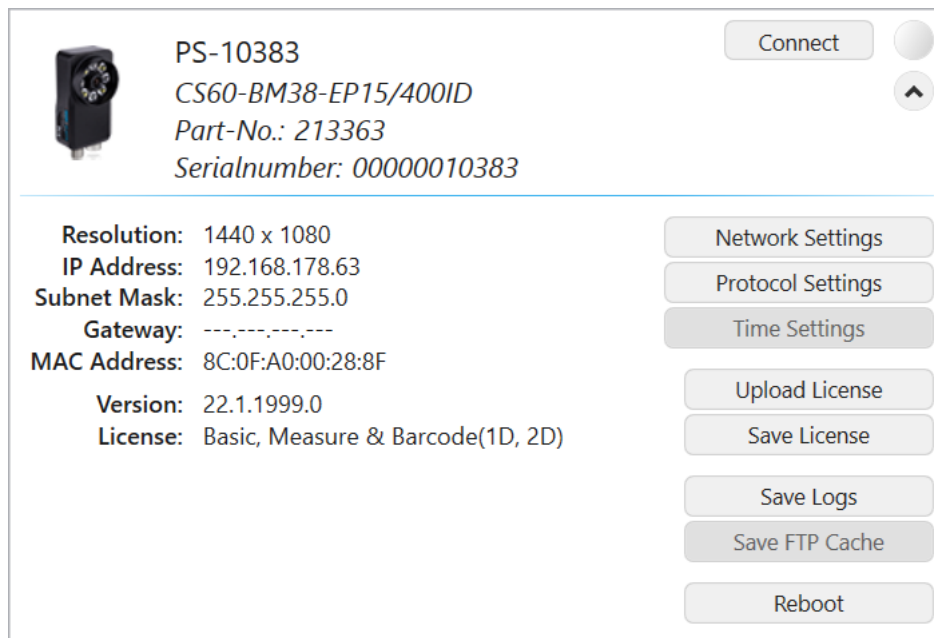
Fig. 3: Extended information about the **CS-60/ID-600** vision sensor.

A license file can be uploaded to the sensor using the **Upload License** button.

With the **Save License** button, a license file can be downloaded from the sensor and saved on the PC (helpful when ordering an upgrade license).

A license is tied to the serial number of a **CS-60/ID-600** vision sensor.

Here is an example of the content of a license file:

```
<?xml version="1.0" encoding="utf-8"?>
    <License xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi=
→"http://www.w3.org/2001/XMLSchema-instance">
```

```
    <LicenseKey>BNAA8-ASA9V-9S8AG-2M67C-X7L76-KXKTZ</LicenseKey>
    <LicenseActivationKey>AAAAA-AAAAA-AAAAH-STSNH-2SJES-446AA</
↪LicenseActivationKey>
    <HardwareId>HBNFJ-SAEFC-K84TB-JSSWJ-JLAEA</HardwareId>
</License>
```

All three values are encrypted. The **LicenseKey** contains the activation of the tools and modules, the **HardwareId** contains the encrypted serial number and the **LicenseActivationKey** contains the activation code of the license server. All three keys must match for the license to be valid. Due to the dependence of the license from the serial number, a license cannot be transferred between sensors.

1.4 Installation

The **nVision-i** installer comes in a 64bit version. The **nVision-i** installer checks for the necessary prerequisites (VC++ runtime), installs them when missing, and also installs the **nVision-i** software.
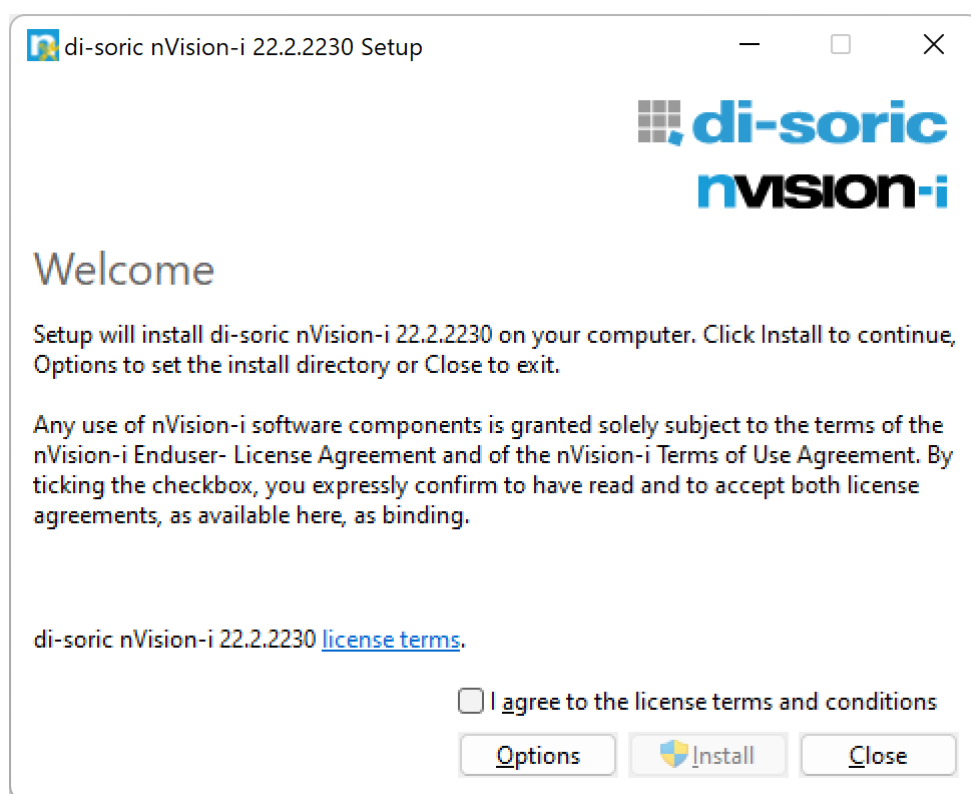
Fig. 4: The welcome screen of the setup program.

The installer welcome screen. Read the license terms, confirm "I agree to the license terms and conditions" and click **Install** to start the installation.
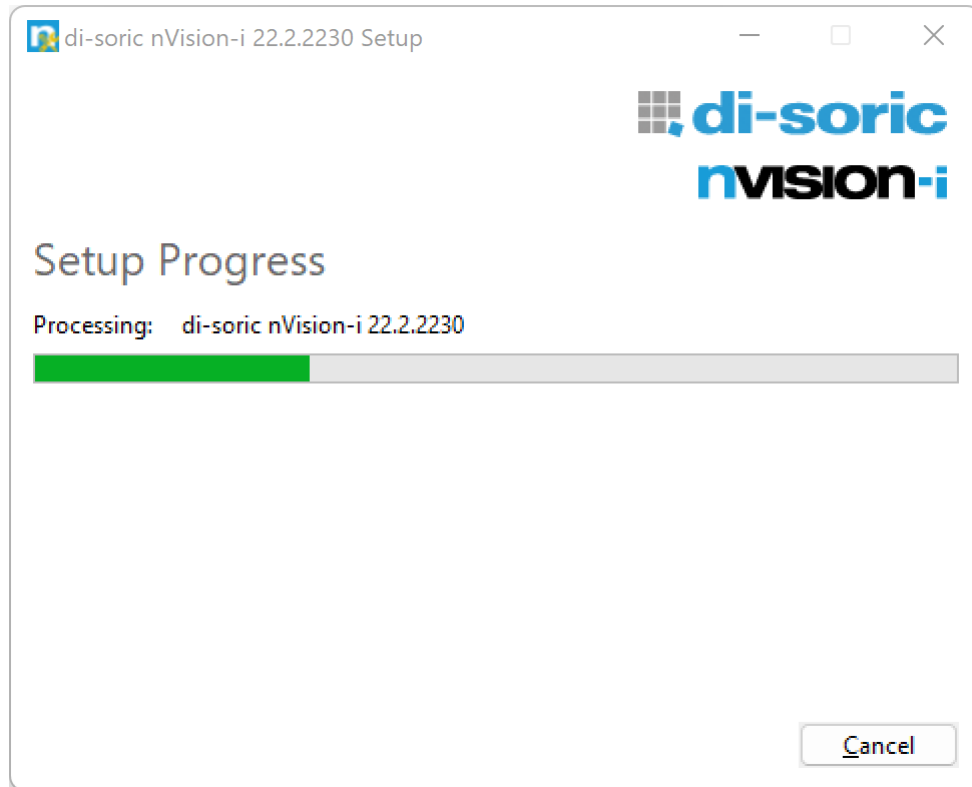
Fig. 5: The progress screen of the setup program.

The installer copies the necessary files to the installation folder (`%PROGRAM_FILES%\di-soric Imaging\nVision-i\`).
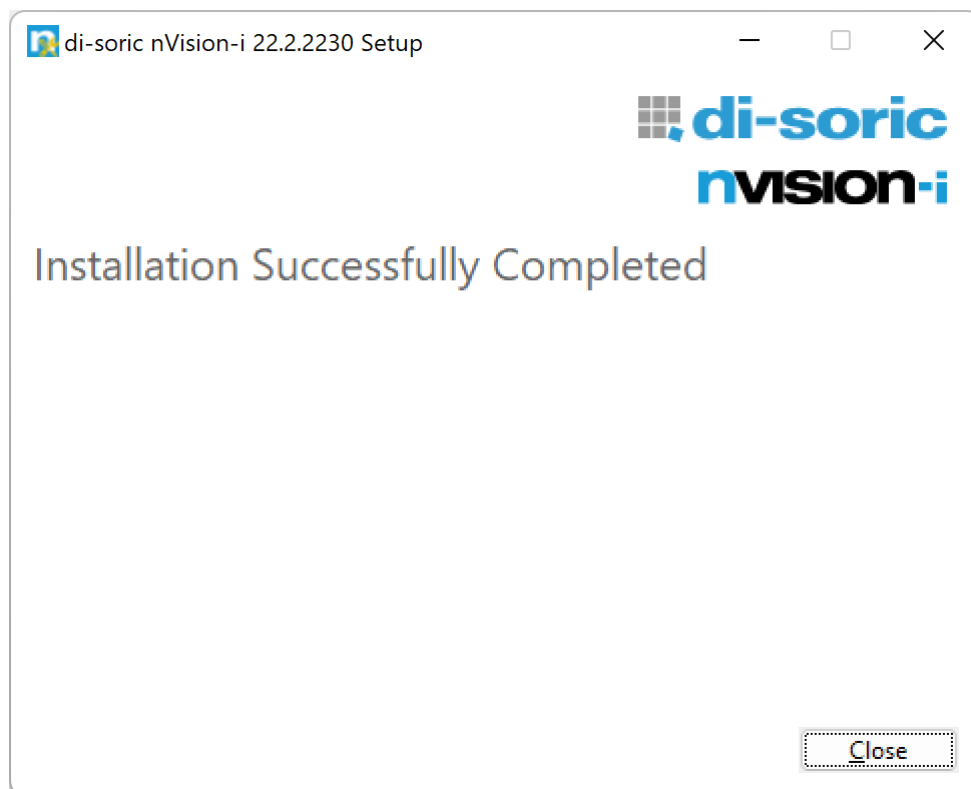
Fig. 6: The finshed screen of the setup program.

When the installation is finished, the confirmation is displayed. Eventually, you need to reboot your PC.

You can install several versions of **nVision-i** side by side. The licensing information is kept intact and valid for all installed versions.

Optionally, on the installer welcome screen, you can click **Options** in order to change the install location to a specific directory.
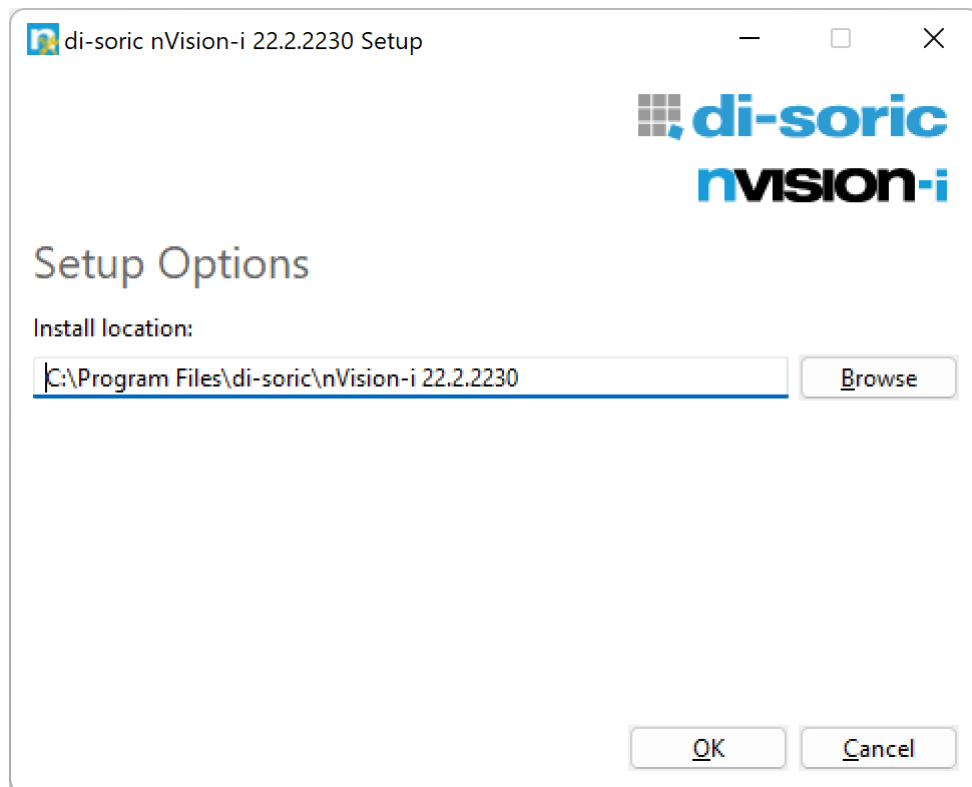
Fig. 7: The options screen of the setup program.

The installer is multilingual and supports German, English, Spanish, French, Italien, Korean and Chinese. If the installer is run without an option, it tries to use the language set for the user or the system. If the language is not supported, English will be used. The language can also be selected using the command line option `-lang xxxx`, e.g.

```
nVision-i-Setup_*_x64.exe -lang 1031
```

The following countries and languages are supported:

Id	BCP 47 Code	Beschreibung
1028	zh-TW	Chinese - Taiwan
1031	de-DE	German - Germany
1033	en-US	English - United States
1034	es-ES	Spanish - Spain
1036	fr-FR	French - France
1040	it-IT	Italian - Italy
1042	ko-KR	Korean
2052	zh-CN	Chinese - China
2055	de-CH	German - Switzerland
2057	en-GB	English - United Kingdom
2058	es-MX	Spanish - Mexico
2060	fr-BE	French - Belgium
2064	it-CH	Italian - Switzerland

continues on next page

Table 1 – continued from previous page

Id	BCP 47 Code	Beschreibung
3076	zh-HL	Chinese - Hong Kong SAR
3079	de-AT	German - Austria
3081	en-AU	English - Australia
3082	es-ES	Spanish - Spain
3084	fr-CA	French - Canada
4100	zh-SG	Chinese - Singapore
4103	de-LU	German - Luxembourg
4105	en-CA	English - Canada
4106	es-GT	Spanish - Guatemala
4108	fr-CH	French - Switzerland
5124	zh-MO	Chinese - Macao SAR
5127	de-LI	German - Liechtenstein
5129	en-NZ	English - New Zealand
5130	es-CR	Spanish - Costa Rica
5132	fr-LU	French - Luxembourg
6153	en-IE	English - Ireland
6154	es-PA	Spanish - Panama
6156	fr-MC	French - Monaco
7177	en-ZA	English - South Africa
7178	es-DO	Spanish - Dominican Republic
7180	fr-029	French - West Indies
8201	en-JM	English - Jamaica
8202	es-VE	Spanish - Venezuela
8204	fr-RE	French - Reunion
9225	en-029	English - Caribbean
9226	es-CO	Spanish - Colombia
9228	fr-CG	French - Congo
10249	en-BZ	English - Belize
10250	es-PE	Spanish - Peru
10252	fr-SN	French - Senegal
11273	en-TT	English - Trinidad
11274	es-AR	Spanish - Argentina
11276	fr-CM	French - Cameroon
12297	en-ZW	English - Zimbabwe
12298	es-EC	Spanish - Ecuador
12300	fr-CI	French - Cote Ivoire
13321	en-PH	English - Philippines
13322	es-CL	Spanish - Chile
13324	fr-ML	French - Mali
14345	en-ID	English - Indonesia
14346	es-UY	Spanish - Uruguay
14348	fr-MA	French - Morocco
15369	en-HK	English - Hong Kong SAR
15370	es-PY	Spanish - Paraguay

Table 1 – continued from previous page

Id	BCP 47 Code	Beschreibung
15372	fr-HT	French - Haiti
16393	en-IN	English - India
16394	es-BO	Spanish - Bolivia
17417	en-MY	English - Malaysia
17418	es-SV	Spanish - El Salvador
18441	en-SG	English - Singapore
18442	es-HN	Spanish - Honduras
19466	es-NI	Spanish - Nicaragua
20490	es-PR	Spanish - Puerto Rico
21514	es-US	Spanish - United States
58378	es-419	Spanish - Latin America
58380	fr-015	French - North Africa

1.5 Configuration

CS-60/ID-600 sensors will be shipped with the IP address preset: `192.168.3.15`.

The network and fieldbus configuration of the sensors can be carried out within nVision-i.
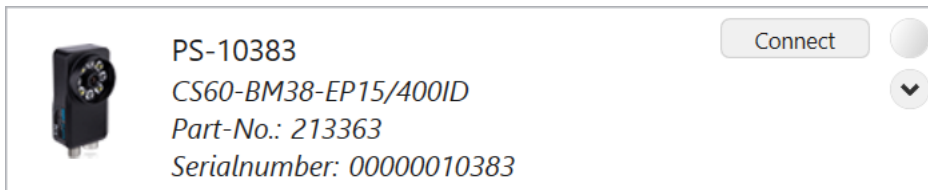
Click the arrow button to see more information about the selected device, such as camera resolution, ethernet configuration, industrial protocols and license information. If a sensor is available, the LED at the right is off.

Click the **Connect** button to establish a connection to the device. If a device is connected, the button changes to **Disconnect**. The LED changes to blue to indicate the connection.

If a sensor is connected to another nVision-i instance, the LED is yellow to indicate that you cannot connect to this sensor.

At the top right of the Devices column are two buttons to start the emulator and refreshes the device list on demand.
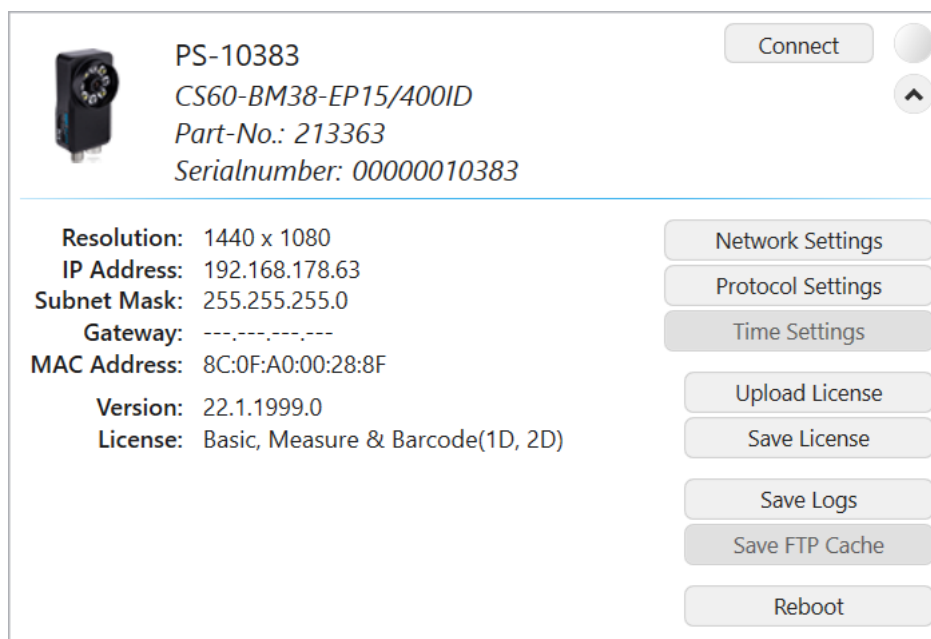
Fig. 8: **CS-60/ID-600**, extended settings

Click on the **Network Settings** button to configure the IP (Internet Protocol) settings of the device:
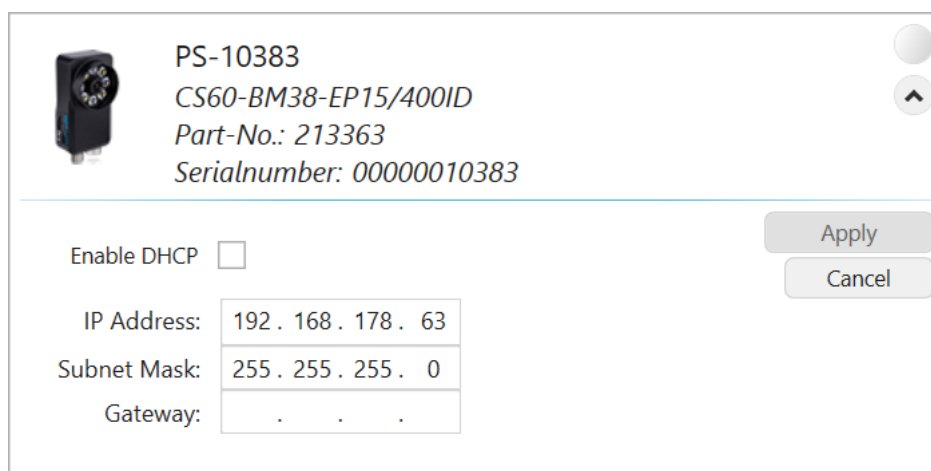
Fig. 9: **CS-60/ID-600**, network settings

Enable DHCP enables or disables the DHCP mode (i.e. the device will get it's IP address from a DHCP server in the network). If not in DHCP mode, you can set or change the **IP Address**, **Subnet Mask** and **Gateway**.

Apply applies the settings you made, **Cancel** quits without changing any setting.

Click the **Protocol Settings** button to configure optional communication protocols.

Fig. 10: **CS-60/ID-600**, protocol settings

Enable Profinet enables or disables the Profinet subsystem of the sensor. You can choose between **CS-60/ID-600 Mode** and **C4G Mode** to select the specific communication mode.

Enable EtherNet/IP enables or disables the EtherNet/IP subsystem of the sensor. This setting is only visible on models with EtherNet/IP support.

Enable Web-Server enables or disables the built-in **CS-60/ID-600** webserver.

Enable FTP-Client enables or disables the FTP/SFTP functionality. The **FTP Cache Size** specifies the number of files (image and text) that can be maximally cached, if the FTP/SFTP server is not available. If the maximum number of files is reached room is made by first deleting the oldest OK files and then deleting the oldest NOK files.

The digital inputs can be used for **job change** or for **linking**.

The digital outputs can each be operated in **PNP (Active High)** or **NPN (Active Low)** mode.

Apply applies the settings you made, **Cancel** quits without changing any setting.

Click the **Time Settings** button to set the **CS-60/ID-600** clock.
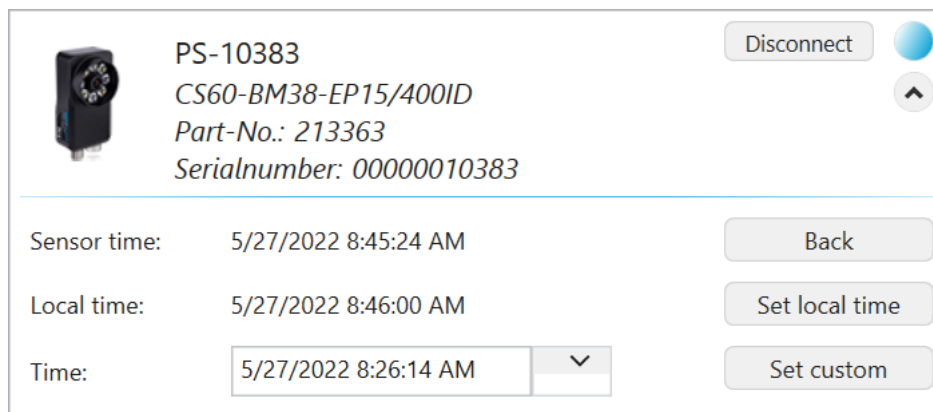
Fig. 11: **CS-60/ID-600**, time settings

Set local time sets the **CS-60/ID-600** clock to the PC time.

Set custom sets the **CS-60/ID-600** clock to the entered time.

Use **Back** to leave the dialog.

Use the **Upload License** and **Save License** buttons to upload license files to the **CS-60/ID-600** or download license files from the **CS-60/ID-600** to the PC.

Use the **Save Logs** button to download the error logs from the **CS-60/ID-600** to the PC.

1.6 Firmware Update

If the firmware of the sensors does not match, a warning is displayed, when you try to connect to the sensor. If the sensor firmware is not updated, you cannot connect.

Clicking **Change to Version** starts the firmware updater.
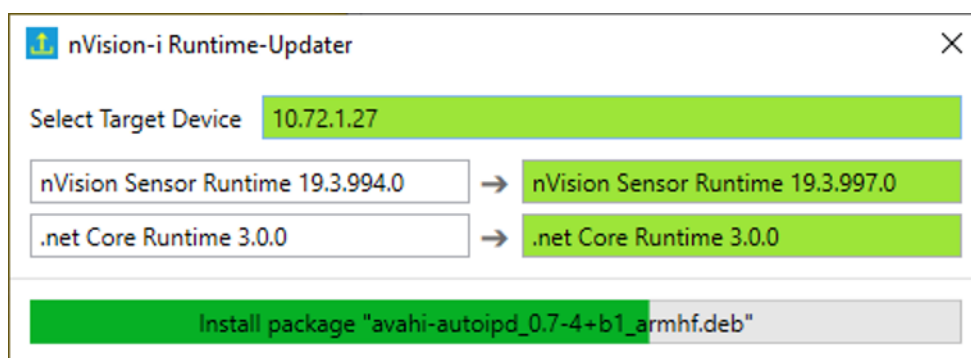
Fig. 12: Update firmware.

After updating the firmware the sensor is restarted. The sensor briefly disappears from the device list, starts new and reappears after the start.

1.7 Reset

If for any reason the **CS-60/ID-600** does not respond and is not found and listed within the nVision-i software any more, you can resort to the following steps:

- Reboot the **CS-60/ID-600** by powering it off, wait about 10 seconds and then power it on again.

If the **CS-60/ID-600** is not visible after a reboot via a power-off/on cycle, you can try to do a hard IP reset.

- Power the **CS-60/ID-600** off. The reset button is located within the lens cover, at 7 o'clock. Push and hold the reset button (with a thin wire or a bent paper clip) while powering the **CS-60/ID-600** on. Make sure to hold the button for at least 30 seconds, until you see a triple flash as confirmation.

Fig. 13: The location of the reset button.

A reset with the button will reset the IP and communication settings to the factory defaults. However, the jobs remain.

- IP address: 192.168.3.15

- Subnet mask: 255.255.255.0

- DHCP: off

- Profinet: off

- EtherNet/IP: off

- Command Server: off

- Web-Server: on

- FTP client: off

- Digital Inputs: Job Change

- The boot job setting is removed (but not the job itself)

USING NVISION-I

After **nVision-i** has been started it shows the main screen.

2.1 The nVision-i Main Screen

The main screen lists the vision sensors found on the network as well as the jobs that are available on the sensors.
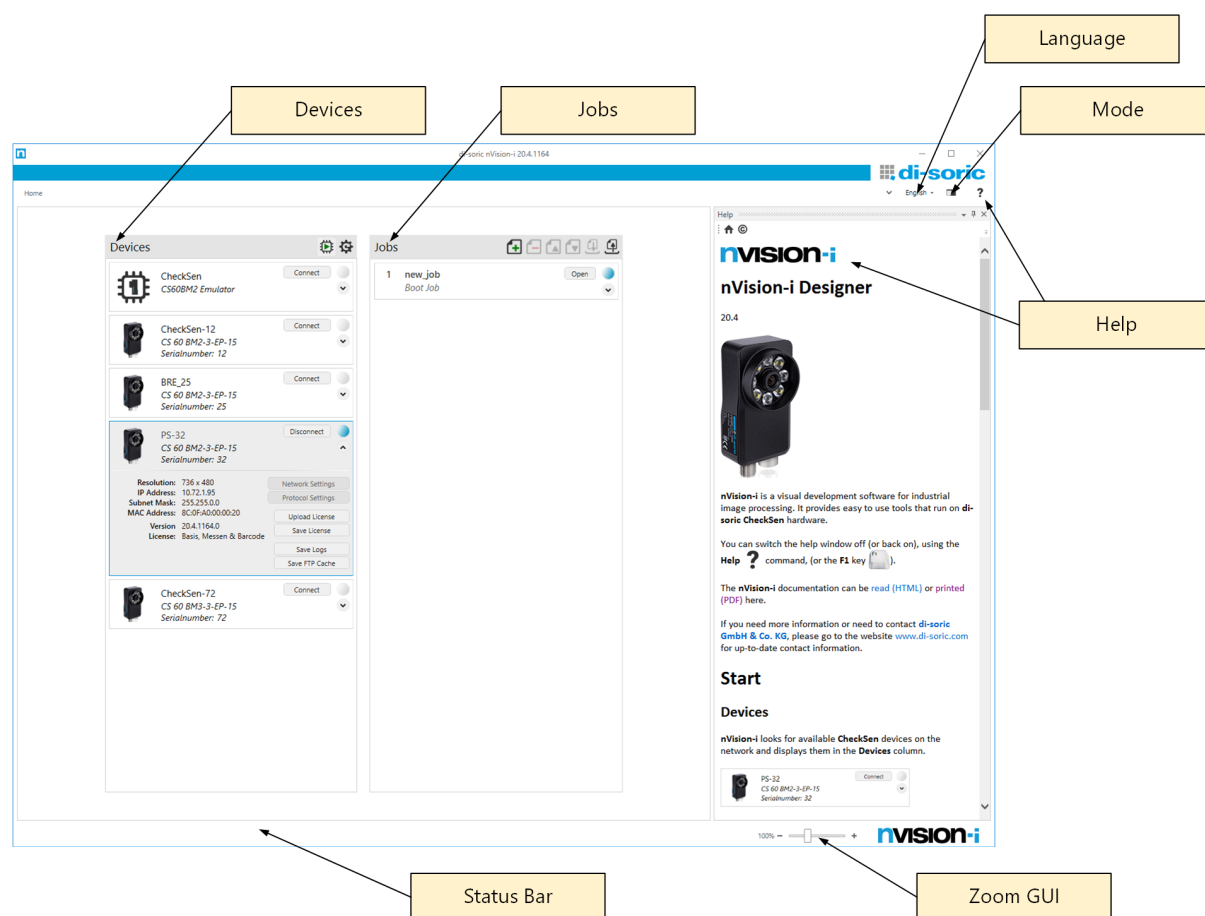
Fig. 1: The nVision-i main screen.

nVision-i enumerates the available **Devices** list by broadcasting *Welcome* messages into the

network. devices send responses back to nVision-i, and these responses are used to build the list of devices. The *Welcome* broadcasts are sent regularly in intervals of about 4 seconds, so that devices that become available will be listed in the **Devices** list.

$+$ adds a device by entering its IP address.

/ starts and stops the the emulator.

refreshes the device list.

When a connection to a vision sensor has been established, the list of available jobs will be displayed in the **Jobs** list. Jobs can be moved up or down in the list, which affects their Job Id, or they can be moved to other devices. In order to move jobs between sensors, nVision-i must have connections to both devices.

adds a job to the list.

removes a job from the list.

moves a job up to another slot.

moves a job down to another slot.

exports a job to disk.

imports a job from disk.

At the top right of the nVision-i windows there are further commands:

nVision-i is available in several languages, you can switch languages with the **Language Se-lection** menu.

With the **Mode Selection** button you can switch between three different appearances: white mode, light mode and dark mode.
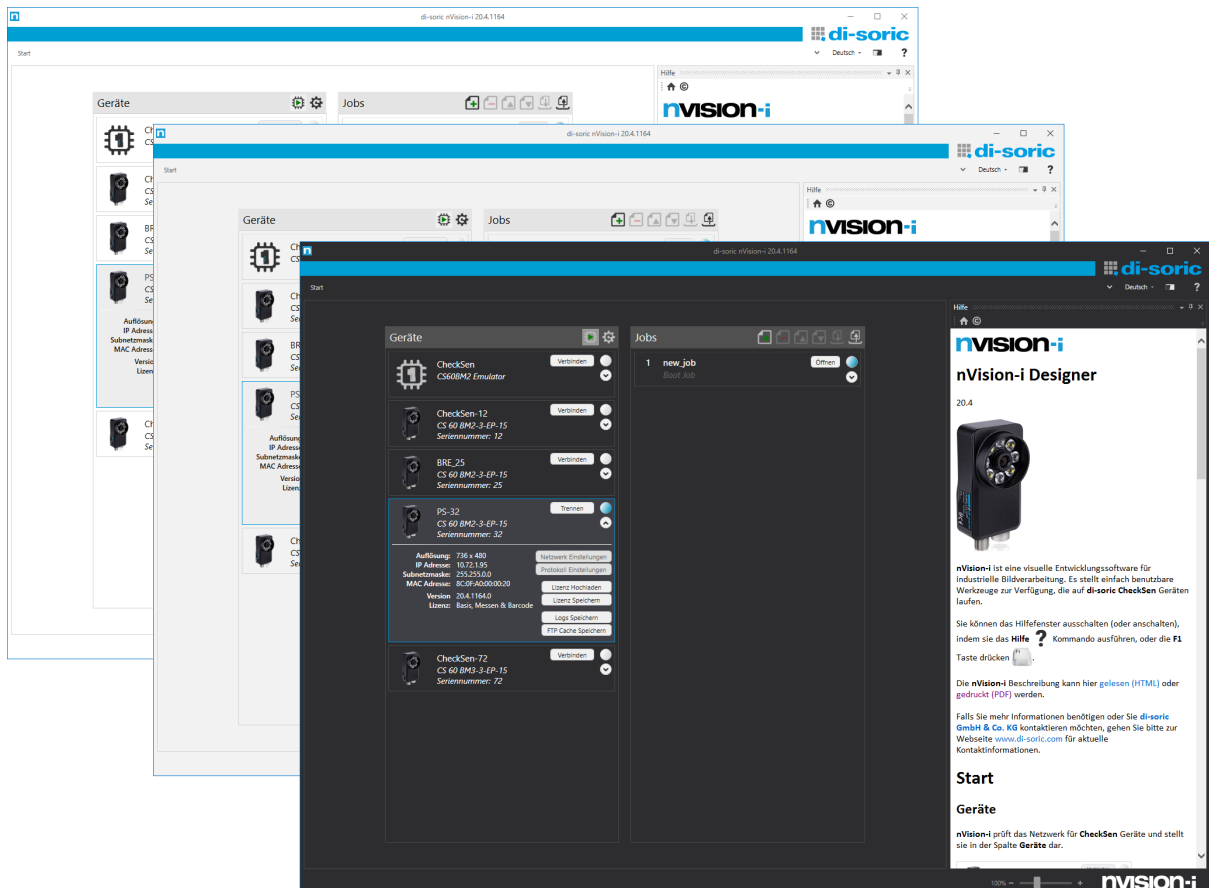
Fig. 2: White, light and dark mode appearance.

The **Help** button shows and hides the help screen.

On the **Status Bar** - in addition to displayed information - you can zoom the GUI, i.e. for use on touch devices.

2.2 Jobs and Tools

The **CS-60/ID-600** vision sensor is programmed by creating jobs. Jobs are built stepwise, by putting one tool after the other. Every job by default starts with the **Acquisition** tool and ends with the **Logic** tool. In between, other tools can be added, which specify the behaviour of the job.

The **Acquisition** tool specifies how the image and other inputs (digital input lines, fieldbus inputs and TCP/IP variables) are acquired.

The **Logic** tool specifies how outputs are determined and communicated to the outside world (digital output lines, fieldbus outputs, TCP/IP variables).

Besides the **Acquisition** and the **Logic** tools, the available processing tools of the **CS-60** sensor can be grouped into **Locate**, **Detect**, **Count** and **Measure**.

The tools in the **Locate** group are **Locate Area**, **Locate Edge**, **Locate Shape** and **Locate Barcode**. They are used to find a reference location in a scene - a so called pose consisting of a

position and a rotation - that can be used by any other tool in the processing pipeline.

The tools in the **Detect** group detect brightness and contrast, as well as areas, edges and barcodes.

The toools in the **Count** group count areas, edges and shapes.

The tools in the **Measure** group measure angles, circles and distances.

Not all tools are available on every sensor. The **Basic** model contains the tools from the **Locate**, **Detect** and **Count** groups - without **Locate Barcode** and **Detect Barcode**. The **Measure** extensions contain the measurement tools and the **ID** extension contains the **Locate Barcode** and **Detect Barcode** tools. In addition, there is an optional **ID-Pro** extension for improved reading of directly marked codes (Direct Part Marking) as well as verification of 2D codes.

The **ID-600** only contains the tool set **ID** and optionally the **ID-Pro** extension.

2.3 Input Data

Many tool parameters can be modified by selecting or typing values, and most of these parameters can also be manipulated by **Linking** - i.e. linking them to digital, fieldbus inputs or TCP/IP command variables.

Linking a value is a two step process: first the value is named on the **Digital Inputs**, **PROFINET Inputs**, **EtherNet/IP Inputs** or **TCP/IP Variables Inputs** tabs of the **Acquisition** tool. Then the named value can be used as a parameter with the ⟻ button.

Input data is acquired whenever an image is acquired. This happens at the beginning of an inspection cycle.

2.3.1 Linking Digital Inputs

The **CS-60/ID-600** vision sensor has two digital inputs that can be used for linking. The input nodes are displayed on the **Digital Inputs** page of the **Acquisition** tool. In order to use them for linking, they must be named. An input is named by adding a **Link** node (with a right click and selecting **Link**), connecting the input node with the **Link** node and typing the respective name in the **Link** node.
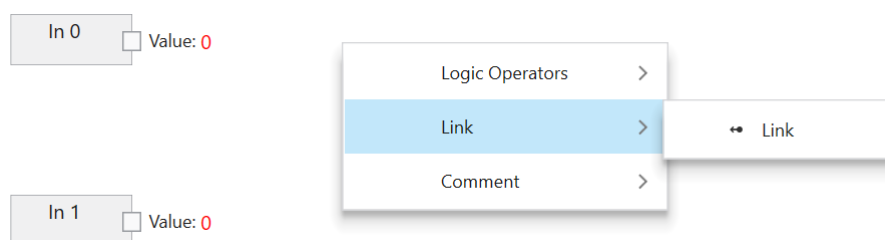
Fig. 3: Add a **Link** node with right click
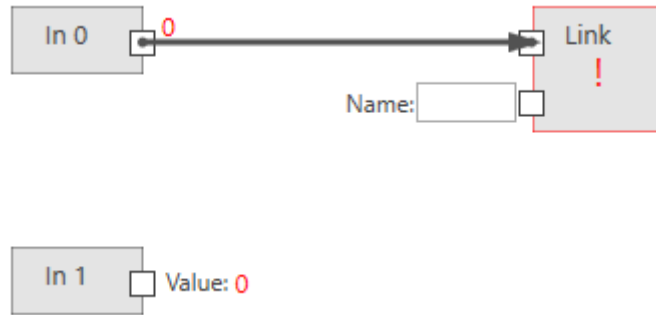
Fig. 4: Connect the input node with the **Link** node
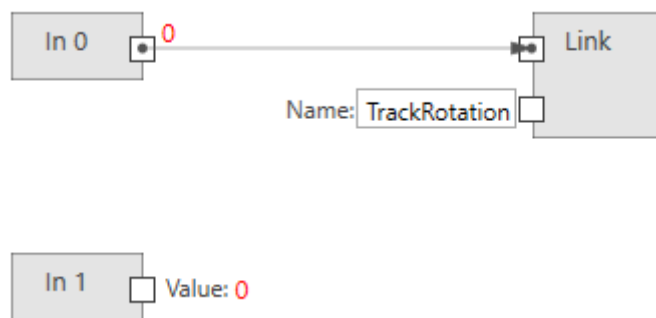
Fig. 5: Name **Link**

Once you have named links, you can use them as boolean values in any tool (besides the Acquisition tool itself). An example could be the **Track rotation** setting in the **Locate Area** tool.
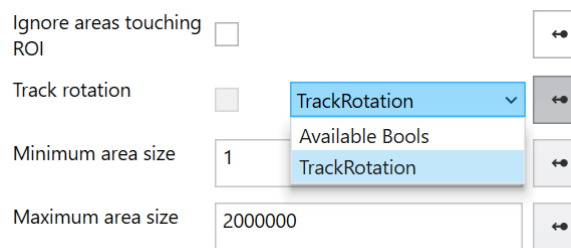
Fig. 6: Example of linking.

For technical reasons, digital inputs can not be used for linking values into the **Acquisition** tool itself.

2.3.2 Linking Fieldbus Inputs

Note: the screenshots show PROFINET, but the procedures for EtherNet/IP are exactly the same, unless where explicitly noted.
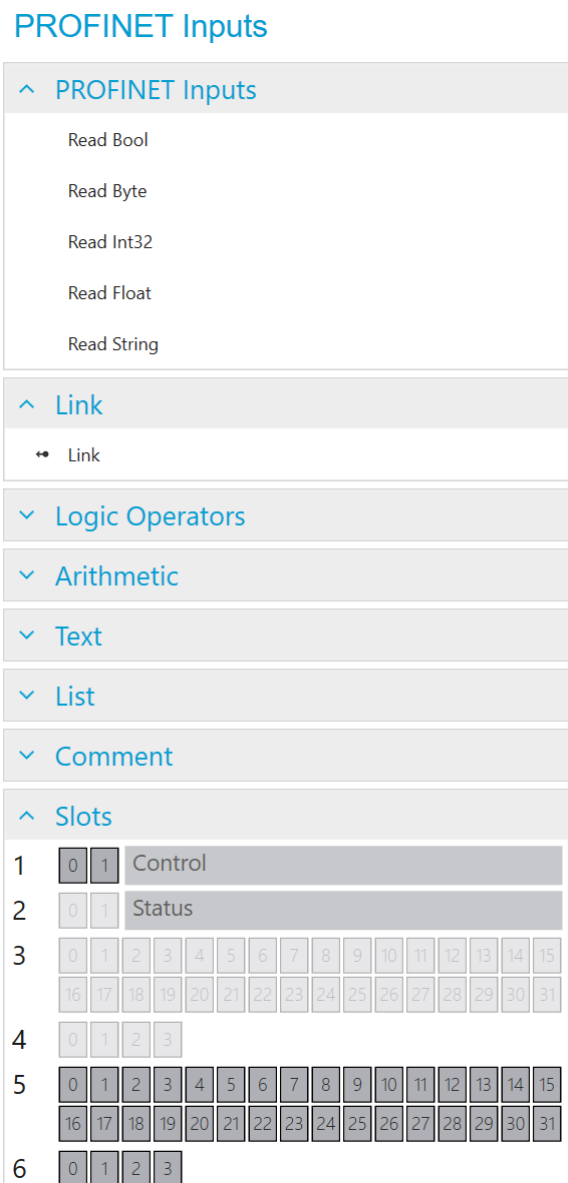
Fig. 7: The context menu on the left.

The **PLC** node is used to define modules in specific slots. This setup must exactly match the setup on the PLC, to enable communication between the PLC and the **CS-60/ID-600**. The PLC node is shown both on the Acquisition tool and the Logic tool. Inside the PLC node, the names Input and Output for the modules are used as seen from the PLC (A PLC Output is transferred from the PLC to the **CS-60/ID-600**, where it is seen as an input. A PLC Input is transferred from the **CS-60/ID-600** to the PLC, on the **CS-60/ID-600** this is seen as an output.)
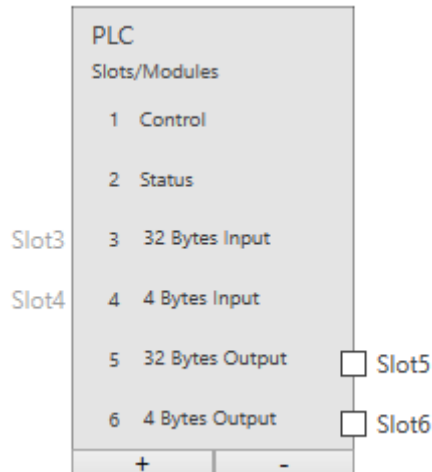
Fig. 8: The PLC node with six modules.

As an example, six modules are selected here (the control and status modules cannot be deselected). On the **PROFINET Inputs** page, only modules 5 and 6 can be connected (from the point of view of the PLC these are outputs, from the point of view of the **CS-60/ID-600** these are inputs.)

The modules are displayed graphically under the heading **Slots**. The used regions within the modules are also displayed.

Fig. 9: A string with 20 characters in module 5 at address 0.

PROFINET or EtherNet/IP can provide input values of different types: boolean (Bool), numeric (Byte, Int32, Float) and text (String). In order to used them for linking they must be defined and named. An input is defined by adding a read node of the desired type (by selecting them in the context menu on the left or with a right click and selecting **Read Bool**, **Read Byte**, **Read Int32**, **Read Float** or **Read String**) and connecting it to a specific module in a slot defined in the PLC node. An input is named by adding a **Link** node (with a right click and selecting **Link**), connecting a read node with the **Link** node and typing the respective name in the **Link** node.

Fig. 10: A boolean input.

The size of a boolean input is 1 byte (but you can allocate up to 8 in one byte).

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Bit indicates the bit to be addressed, where 0 is the rightmost bit and 7 is the leftmost bit in the byte.

Fallback specifies the boolean value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 11: A byte input.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Fallback specifies the byte value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 12: An integer input.

The size of an integer input is 4 bytes.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit integer. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback specifies the integer value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 13: A float input.

The size of a floating point input is 4 bytes.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback specifies the float value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 14: A string input.

The size of a string input is (2 + n) * s, where n is the maximum number of characters and s = 1 (for an Ansi STRING) or s = 2 (for a Unicode WSTRING). The first byte/word (Ansi/Unicode) contains the maximum number of characters the string can hold. The second byte/word (Ansi/Unicode) contains the actual number of characters the string holds.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Type specifies if the string is Ansi (STRING) or Unicode (WSTRING). This input is omitted with EtherNet/IP, as EtherNet/IP only supports Ansi texts.

Address specifies the address in the module.

Length specifies the number of characters in the string. A length of 1 specifies a single character.

Endianness specifies the byte order of the two bytes that comprise a 16 bit Unicode character. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first. With EtherNet/IP, this input is omitted as it is only relevant for Unicode texts.

Fallback specifies the string value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

2.3.3 Linking TCP/IP Command Variable Inputs

TCP/IP Variables Inputs

^ TCP/IP Variables Inputs	
Read Bool	
Read Byte	
Read Int32	
Read Float	
Read String	
^ Link	
↔ Link	
∨ Logic Operators	
∨ Arithmetic	
∨ Text	
∨ List	
∨ Comment	

Fig. 15: The context menu on the left.

The **Inputs** node is used to name variables in the TCP/IP namespace.

Variables

exp	☐ Variable 1:
+	-

Fig. 16: The inputs node with one variable.

As an example one variable is defined here. The name must begin with a letter and can optionally be followed by one or more letters, numbers or _ (underscore). This name is valid in the TCP/IP namespace, i.e. when used with the TCP/IP command `set name = value`. For example, the command `set exp=100` would assign the value `100` to the variable `exp`.

TCP/IP command variables can provide input values of different types: boolean (Bool), numeric (Byte, Int32, Float) and text (String). In order to use them for linking they must be defined and named. An input is defined by adding a read node of the desired type (by selecting them in the context menu on the left or with a right click and selecting **Read Bool**, **Read Byte**, **Read Int32**, **Read Float** or **Read String**) and connecting it to a specific variable defined in the Inputs node. An input is named by adding a **Link** node (with a right click and selecting **Link**), connecting a read node with the **Link** node and typing the respective name in the **Link** node.
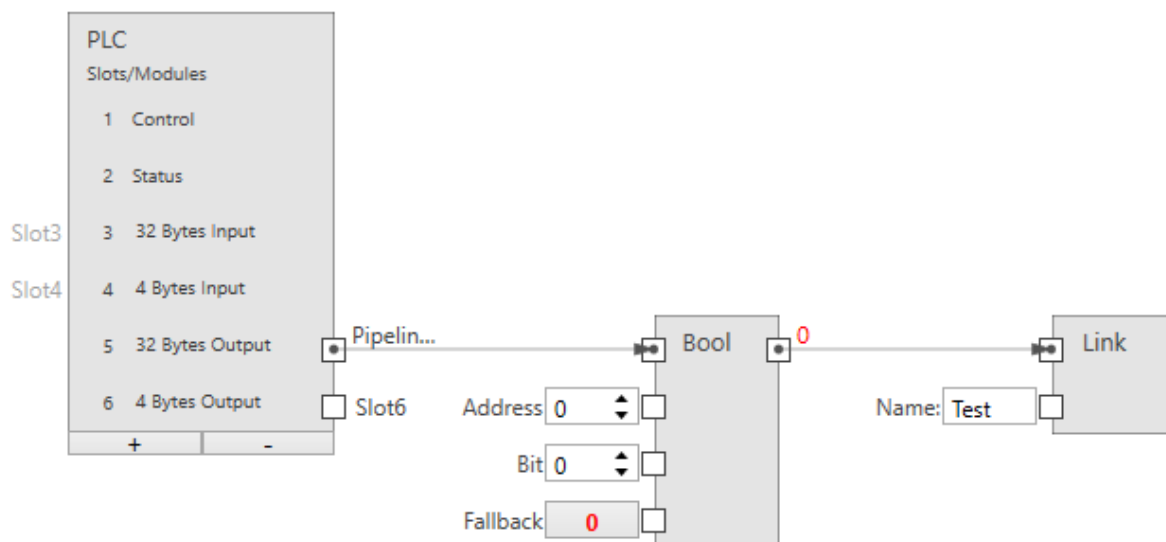
** Boolean Input**

Fig. 17: A boolean input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the boolean value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Byte Input

Fig. 18: A byte input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the byte value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.
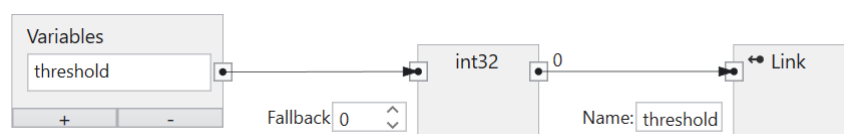
Integer Input

Fig. 19: An integer input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the integer value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Float Input

Fig. 20: A float input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the float value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

String Input

Fig. 21: A string input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the string value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

2.4 Output Data

A job can produce output data. Ouputs are produced at the end of an inspection cycle.

The **Logic** tool at the end of a job is used to specify the outputs of a job.

2.4.1 Producing Digital Outputs

The **CS-60/ID-600** has four digital outputs.

By default, the four digital outputs of the **CS-60/ID-600** are displayed as nodes on the canvas.
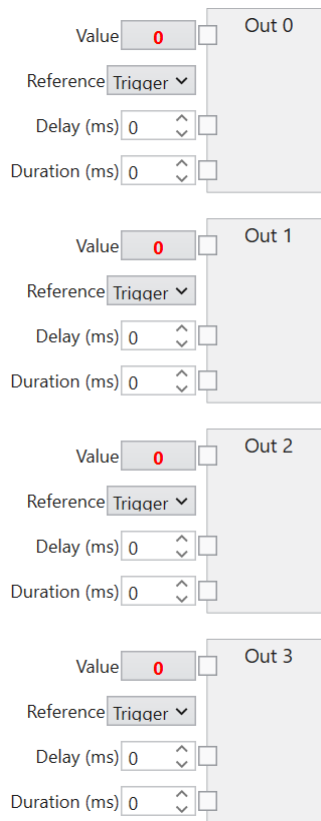
Fig. 22: The **Logic** tool digital output nodes.

The **Delay** and the **Length** of a digital output signal can be specified (in milliseconds). The delay is measured from the start of the image acquistion, a length of 0 means infinite duration.

Tools can be dragged in from the tool list, to create tool nodes that provide values that can be combined logically to create the desired outputs. Here is an example:
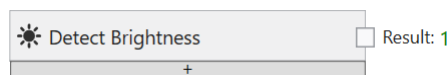
Fig. 23: A tool node.

The primary result of the tool is **True** or **False** and can be connected to any of the four outputs (by dragging from the little square output pin named **Result** of the tool to the respective input pin of the output named **Value**.

Fig. 24: Connecting a tool and an output.

2.4. Output Data 33

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added to the canvas by clicking with the right mouse button.
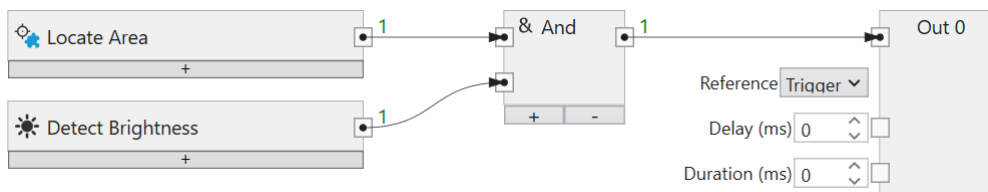
Fig. 25: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node or a connection between two nodes can be selected and deleted with the **Del** key.

A click on the little **+** button at the bottom of a tool node adds an additional row to the tool node, where you can create a logic expression of any secondary tool result.
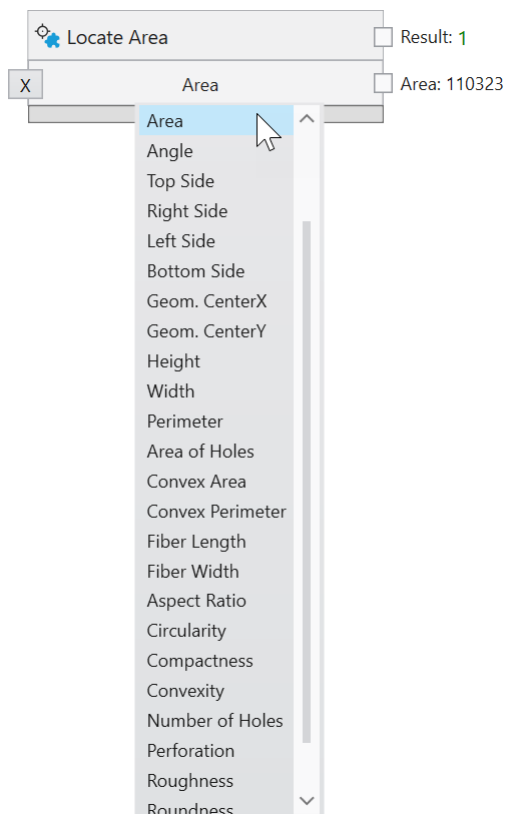
Fig. 26: Create additional boolean outputs with secondary tool results.

Enter minimum or maximum boundaries and create additional logic results from secondary numeric tool results.

The **X** button deletes a row, if it is no longer needed.

2.4.2 Producing Fieldbus Outputs

Note: the screenshots show PROFINET, but the procedures for EtherNet/IP are exactly the same, unless where explicitly noted.
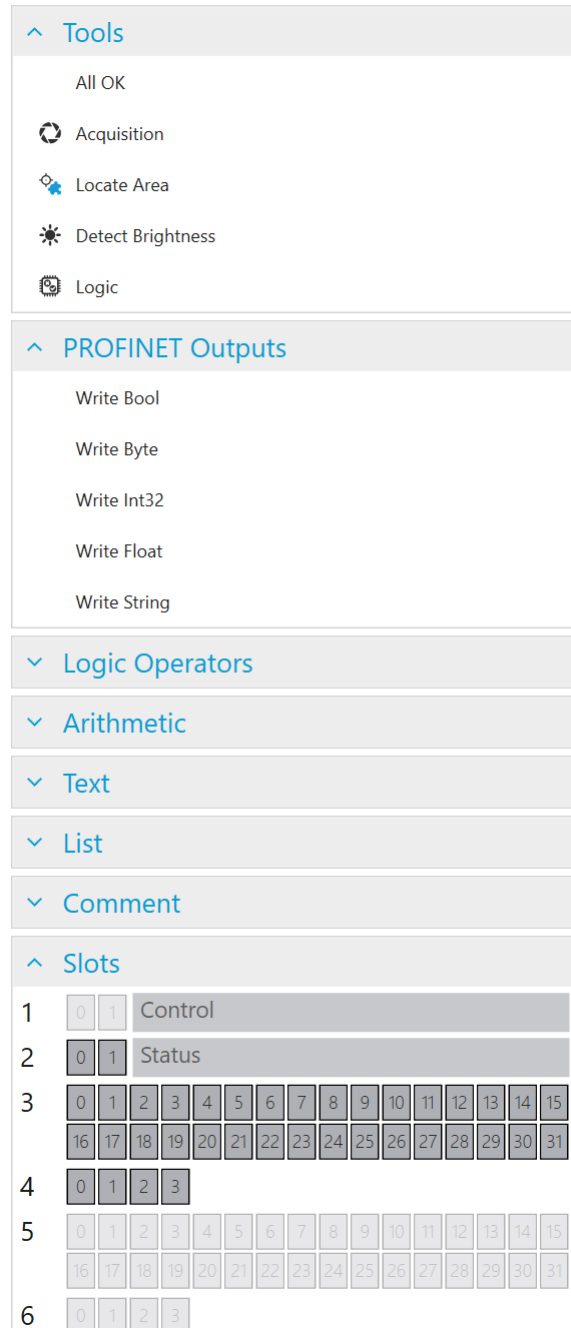
Fig. 27: The context menu of the PROFINET Outputs.

Fieldbus outputs must be defined by by selecting them in the context menu on the left or by clicking with the right mouse button into the canvas and chosing **Write Bool**, **Write Byte**, **Write Int32**, **Write Float** or **Write String**.

Tools can be selected in the context menu on the left, to create tool nodes that provide values that can be connected to the PROFINET output nodes. Here is an example:
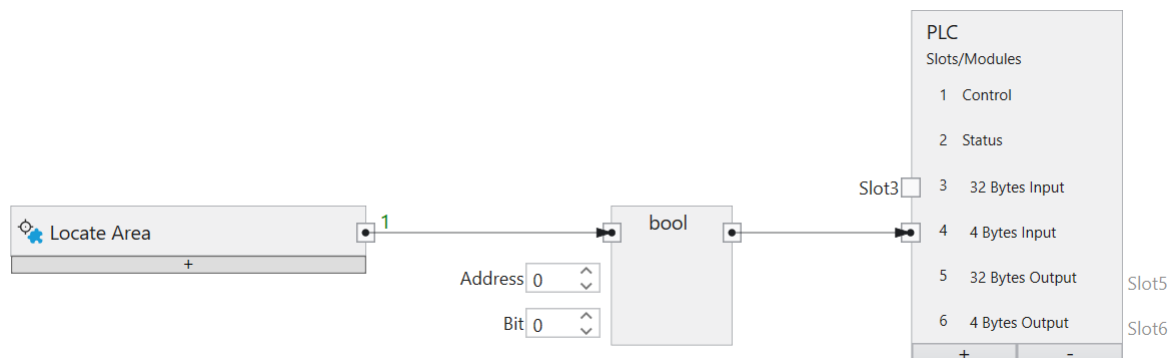
Fig. 28: A tool node for PROFINET.

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added to the canvas by clicking with the right mouse button.
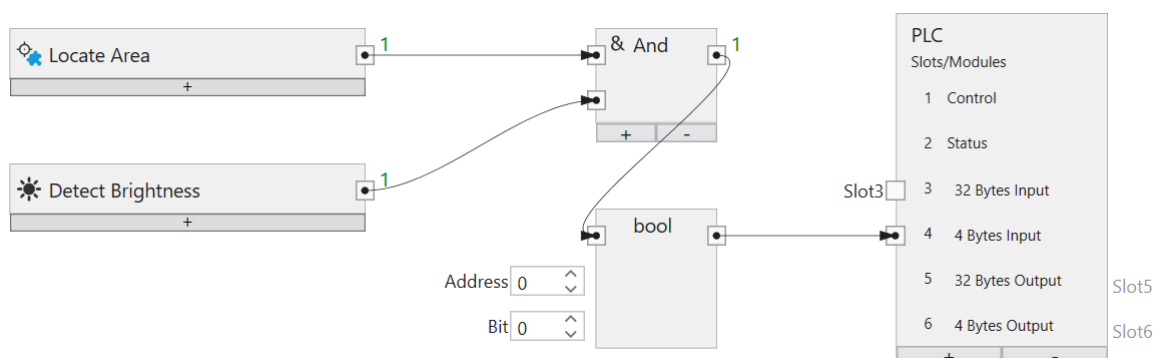
Fig. 29: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node or a connection between two nodes can be selected and deleted with the **Del** key.

In addition to logical values, PROFINET or EtherNet/IP can also transfer numerical values or texts.

The PROFINET or EtherNet/IP output nodes (**Write Bool**, **Write Byte**, **Write Int32**, **Write Float**, **Write String**) determine at which address within a module the values are written.
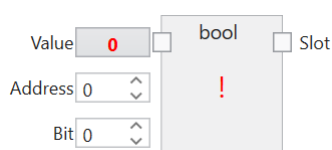
Fig. 30: A Boolean output.

The size of a boolean output is 1 byte (but you can allocate up to 8 bits in one byte).

Value is the value to be transferred.

Address specifies the address in the module.

Bit indicates the bit to be addressed, where 0 is the rightmost bit and 7 is the leftmost bit in the byte.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
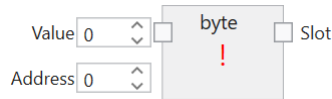
Fig. 31: A byte output.

Value is the value to be transferred.

Address specifies the address in the module.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
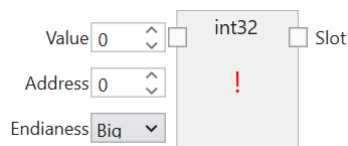
Fig. 32: An integer output.

The size of an integer output is 4 bytes.

Value is the value to be transferred.

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
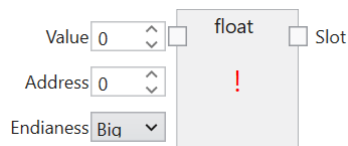
Fig. 33: A floating point output.

The size of a floating point output is 4 bytes.

Value is the value to be transferred.

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
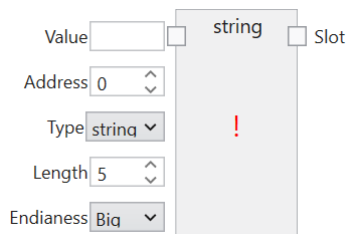
Fig. 34: A text output.

The size of a string output is (2 + n) * s, where n is the maximum number of characters and s = 1 (for an Ansi STRING) or s = 2 (for a Unicode WSTRING). The first byte/word (Ansi/Unicode) contains the maximum number of characters the string can hold. The second byte/word (Ansi/Unicode) contains the actual number of characters the string holds.

Value is the value to be transferred.

Type specifies if the string is Ansi (STRING) or Unicode (WSTRING). This input is omitted with EtherNet/IP, as EtherNet/IP only supports Ansi texts.

Address specifies the address in the module.

Length specifies the number of characters in the string. A length of 1 indicates a single character.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first. With EtherNet/IP, this input is omitted as it is only relevant for Unicode texts.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).

The **PLC** node is used to define modules in specific slots. This setup must exactly match the PROFINET or EtherNet/IP setup on the PLC, to enable communication between the PLC and the **CS-60/ID-600**. The PLC node is shown both on the Acquisition tool and the Logic tool. Inside the PLC node, the names Input and Output for the modules are used as seen from the PLC. A PLC Output is transferred from the PLC via PROFINET or EtherNet/IP to the **CS-60/ID-600**, where it is seen as an input. A PLC Input is transferred from the **CS-60/ID-600** via PROFINET or EtherNet/IP to the PLC, on the **CS-60/ID-600** this is seen as an output.

Fig. 35: The PLC node with six modules.

As an example, six modules are selected here.

The modules are displayed graphically under the heading **Slots**. The used regions within the modules are also displayed.

2.4.3 Producing TCP/IP Variables Outputs
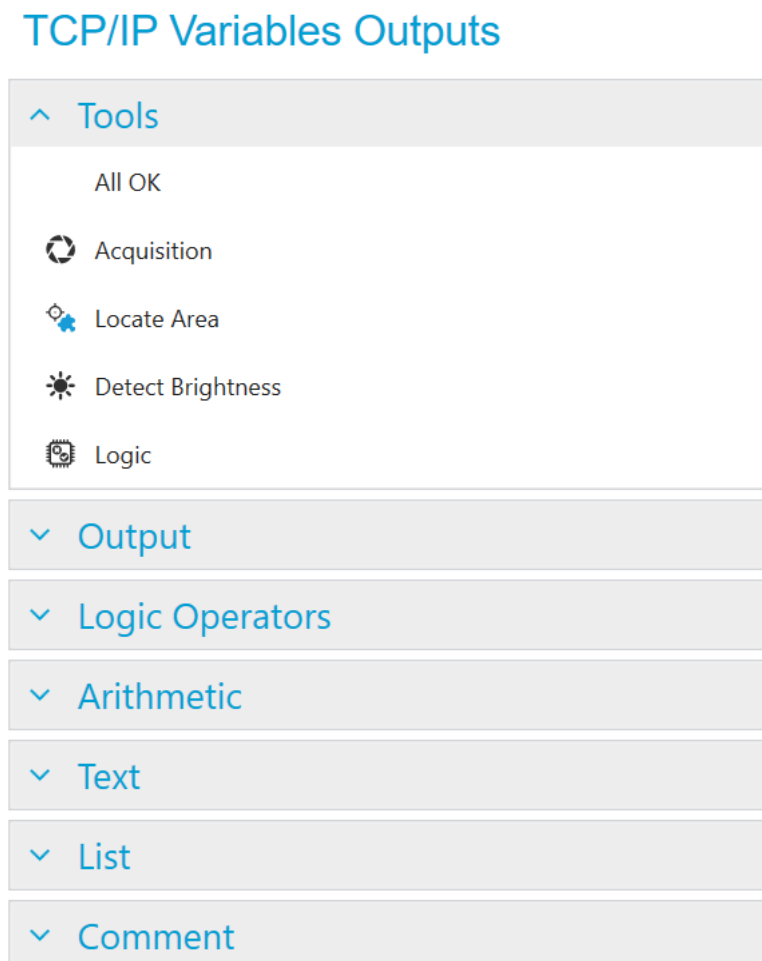
Fig. 36: The context menu of the TCP/IP Variables Outputs.

TCP/IP outputs must be defined by by selecting them in the context menu on the left or by clicking with the right mouse button and chosing **Write Bool**, **Write Byte**, **Write Int32**, **Write Float** or **Write String**.

Tools can be selected in the context menu on the left, to create tool nodes that provide values that can be connected to the TCP/IP Outputs node. Here is an example:
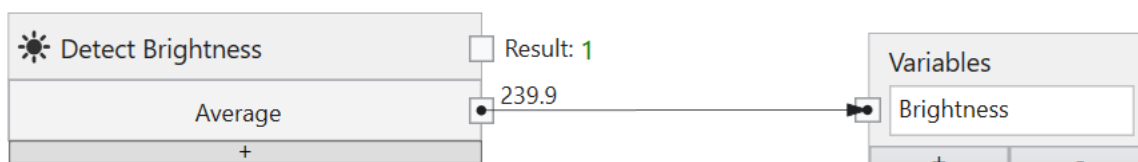
Fig. 37: A tool node for TCP/IP.

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added by clicking with the right mouse button.
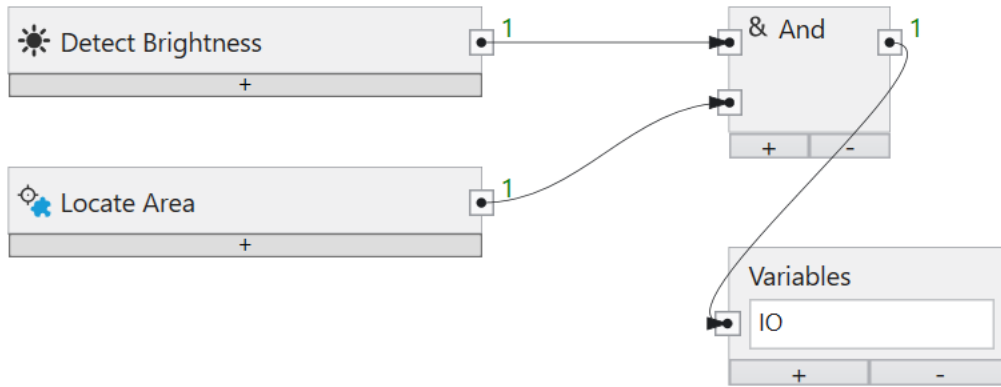
Fig. 38: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node or a connection between two nodes can be selected and deleted with the **Del** key.

In addition to logical values, TCP/IP commands variables can also transfer numerical values or texts.

MACHINE VISION METHODS

3.1 Pattern Matching

nVision-i provides three different methods for pattern recognition: correlation, geometric and keypoint. They are suitable for different scenarios of recognition of two-dimensional patterns under a vertical camera view. The first step is to train a template or pattern. It must be ensured that both the template and the image are exposed in such a way that the relevant pixel areas are not saturated (either deep black or bright white). In addition, the template should be selected in such a way that there is enough structure, such as gray value gradients, edges and corners.

The correlation method is based on gray values. The normalized cross-correlation between a template and an image area in the search image is calculated. The method can compensate linear variations in brightness to a certain degree. It is rotation-invariant, i.e. the original can be detected under different rotations in the search image. The correlation method is memory-intensive, and carries the risk of out-of-memory errors that lead to misbehavior, if the templates or the allowed rotation are chosen too large. Furthermore, the method is computationally intensive and should only be chosen if other methods do not lead to the desired success.

The geometric method is based on brightness gradients. The method can compensate for occlusions and copes relatively well with background clutter. It is rotation-invariant, i.e. the template can be recognized under different rotational positions in the search 341,17304,5 image. The geometric method is memory intensive, and carries the risk of out-of-memory errors leading to misbehaviour, if the templates or the permissible rotation are too large. In addition, the procedure computationally intensive and should only be selected if other methods do not lead to the desired success.

The keypoint method detects few key points in the template and the image area, and uses the correspondence of these points to determine the position. When selecting the template, it is important that the number of the found key points is sufficient, and that they are distributed across the whole template. In addition to the rotation invariance, the keypoint method is invariant to small changes in size.

The recognition of the patterns in the edge areas of the search image is made more difficult by two effects:

- the vignetting of the optics causes the brightness to drop in the corners.

- the distortion of the optics causes a geometric distortion that becomes stronger towards the edges.

Both effects make pattern recognition difficult in the peripheral areas of the image. The performance of pattern recognition can be improved if these effects are compensated.

3.2 Shading Correction

Shading correction compensates for the decrease in brightness in a captured image towards the edge (also called vignetting). The shading is an optical effect that varies with the lenses used and is also caused or intensified by the lighting. by the illumination. A prerequisite for optimal shading correction is the orthogonal viewing of a scene (the camera must look at the scene at a right angle).
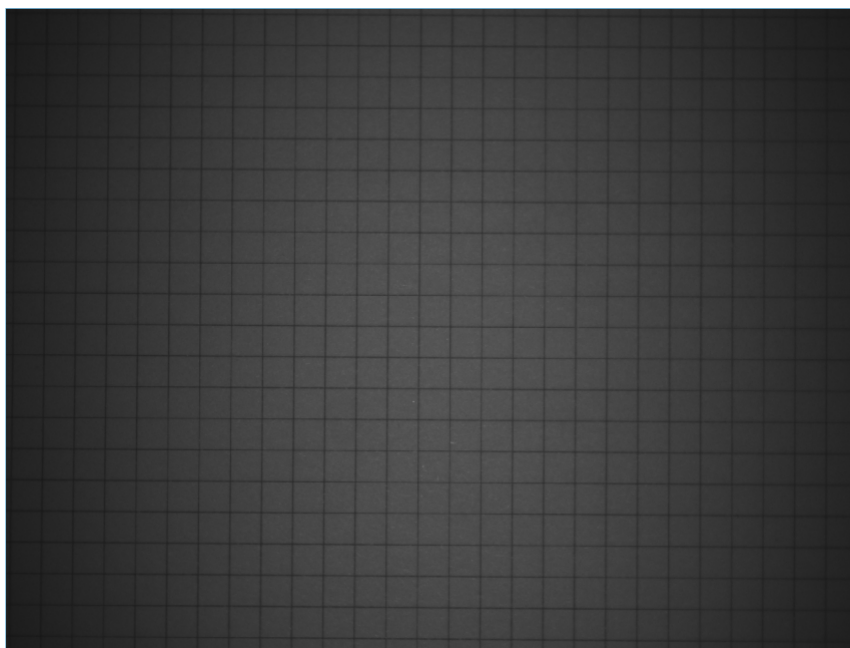
Fig. 1: Edge shading.

The calibration should be performed anew each time the lens is changed or the illumination is changed.

For calibration, an image should be taken that is as uniformly bright as possible, making sure that the intensities are neither too dark nor too bright. This can be done with the histogram display or with the functions **Highlight Underexposure** and **Highlight Overexposure** from the **Visualization** area.

If the correction image and the parameters for exposure and illumination are set correctly, the **Correct** key can be used to calculate the correction.

After the calculation has been completed, the correction can be saved with **Save correction** so that it can be applied in jobs, or deleted again by the **Reset** key.
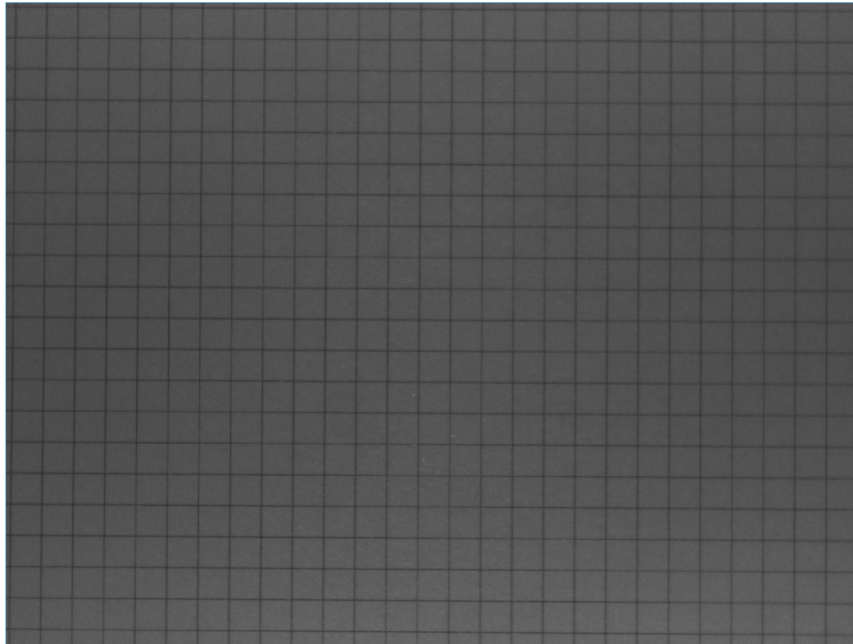
Fig. 2: Corrected shading.

If a sensor has been calibrated, the shading correction can be selected or deselected in the acquisition tool of a job. The correction of an image takes time, so the acquisition time will increase if the shading correction is selected. A Shading correction improves the performance of subsequent tools, because blob analysis, detection of edges and detection of patterns also at the uniform brightness conditions even at the edge of the image.

3.3 Distorion Correction

Distortion correction compensates for optical distortions that occur due to different lenses in varying degrees towards the edge. A prerequisite for optimal distortion correction is the orthogonal viewing of a scene (the camera must look at the scene at a right angle). A second effect of the distortion correction is the determination of the intrinsic and extrinsic camera parameters, as far as they are necessary for a calibrated measurement of an orthogonal measuring plane.

Calibration must be performed anew each time the lens is changed or the measurement plane is changed.

For the calibration, an image of a point-shaped calibration target must be taken, filling the image field as far as possible, and the point distances must be entered. Such targets are available for purchase, or they can be printed out by yourself with a printer for less precise requirements. In the directory **Docs** of the nVision-i Installation there is a PDF file for printing for different image field sizes.

50 x 33 2mm

.
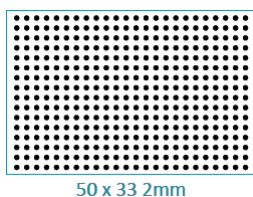
The calibration target must be placed in the measuring plane (only in this measuring plane the measured values are valid). With the endocentric lenses used, image parts that are further away are imaged smaller and image parts that are closer are imaged larger, i.e. the measured values are also correspondingly smaller or larger. Correctly measured values are only determined in the measuring plane.

The **Threshold** for the detection of the calibration target can be determined **Automatically** or set via a slider. The aim is to optimally determine the points of the calibration target.
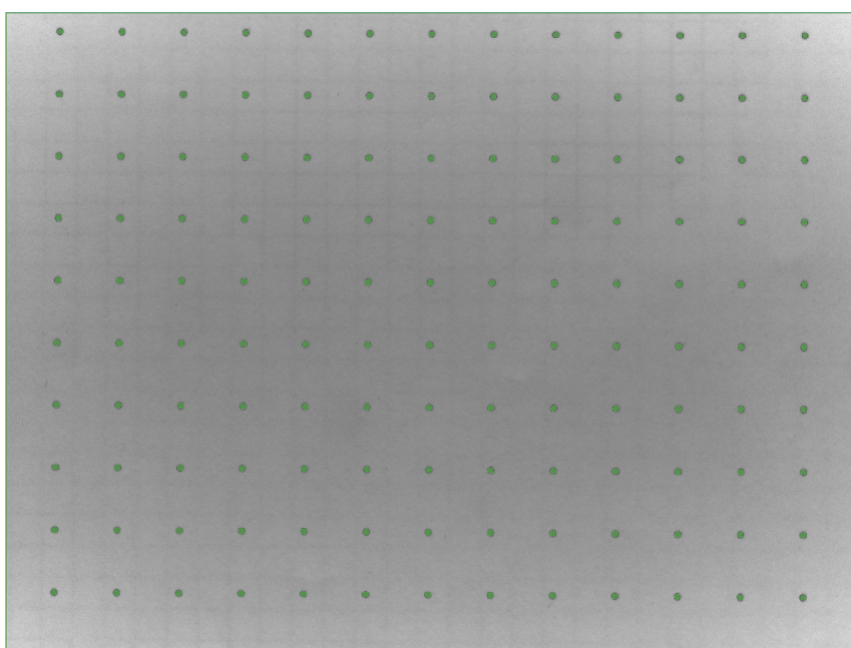
Fig. 3: Image of a target for geometric calibration.

The **Distance** is the distance between two points of the target (from center to center). This value must match the actual distance of the points on the target in order to calibrate correctly.

The region in the image should be used to narrow down the relevant points of the calibration target.

For correct setting the **Raster** can be selected to show the grid lines between the points of the target.
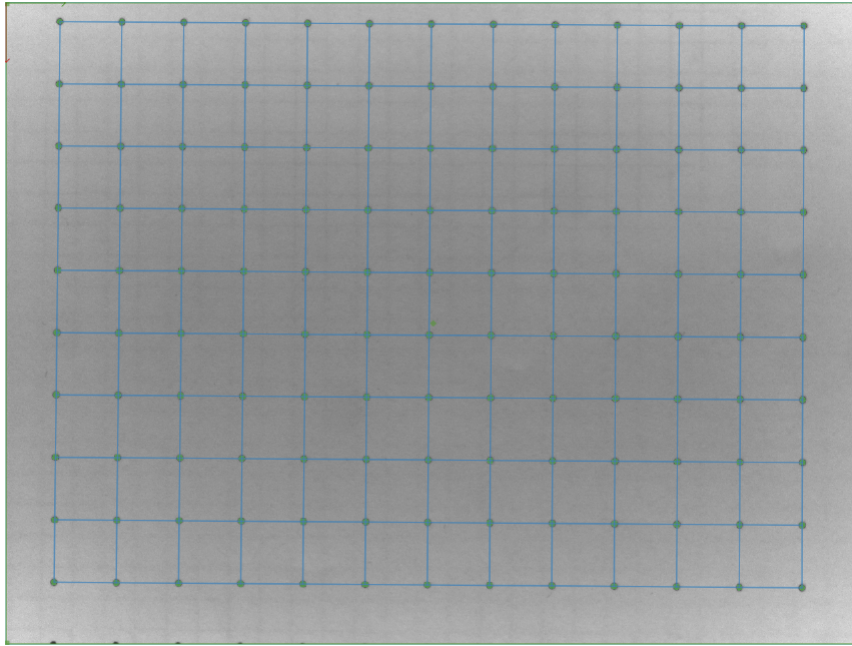
Fig. 4: Target with grid lines.

If a sensor has been calibrated, the equalization can be selected or deselected in the acquisition tool of a job. The correction of an image takes time, so the time for acquisition will increase if distortion correction is selected. Distortion correction improves the performance of subsequent tools, because blob analysis, detection of edges and detection of patterns also at the geometry even at the edge of the image.

3.4 Blob Analysis

A group of spatially contiguous pixels is commonly referred to as a blob.

Blob analysis uses a set of contiguous pixels, which are usually generated by imaging real objects, and from this set of pixels values or features are calculated such as their area, position, etc.

Often Blobs are generated by a segmentation process, and one of the simplest segmentation methods is the binary thresholding: each pixel with a value below the threshold is part of the background and every other pixel is part of a blob. Which blob the pixel belongs to is determined by a connectivity analysis, sometimes called labeling or analysis of connected components.

Here is a list of the features that can be calculated:

Feature	Description
Area	The area as a number of pixels.
Angle	The angle of the area.
Centroid	The center of gravity based on the region.
Top Side	The topmost row position (inclusive).
Right Side	The rightmost column position (exclusive).
Bottom Side	The bottommost row position (exclusive).
Left Side	The leftmost column position (inclusive).
Geom. Center	The geometric center.
Height	The height (axis parallel).
Width	The width (axis parallel).
Perimeter	The perimeter.
Area of Holes	The area of the holes as a number of pixels.
Convex Area	The area of the convex hull.
Convex Perimeter	The perimeter of the convex hull.
Fiber Length	A length measurement of fiber-like objects.
Fiber Width	A thickness measurement of fiber-like objects.
Aspect Ratio	The width to height ratio.
Circularity	The circularity.
Compactness	The compactness.
Convexity	The convexity.
Number of Holes	The number of holes of the region.
Perforation	The perforation.
Roughness	The roughness.
Roundness	The roundness.
Sphericity	The sphericity.

Blob analysis features can be applied in the **Locate Area**, **Detect Area Pixels** and **Count Areas** tools.

3.4.1 Area

The area of a region is an integer value and specifies the number of pixels in the object.
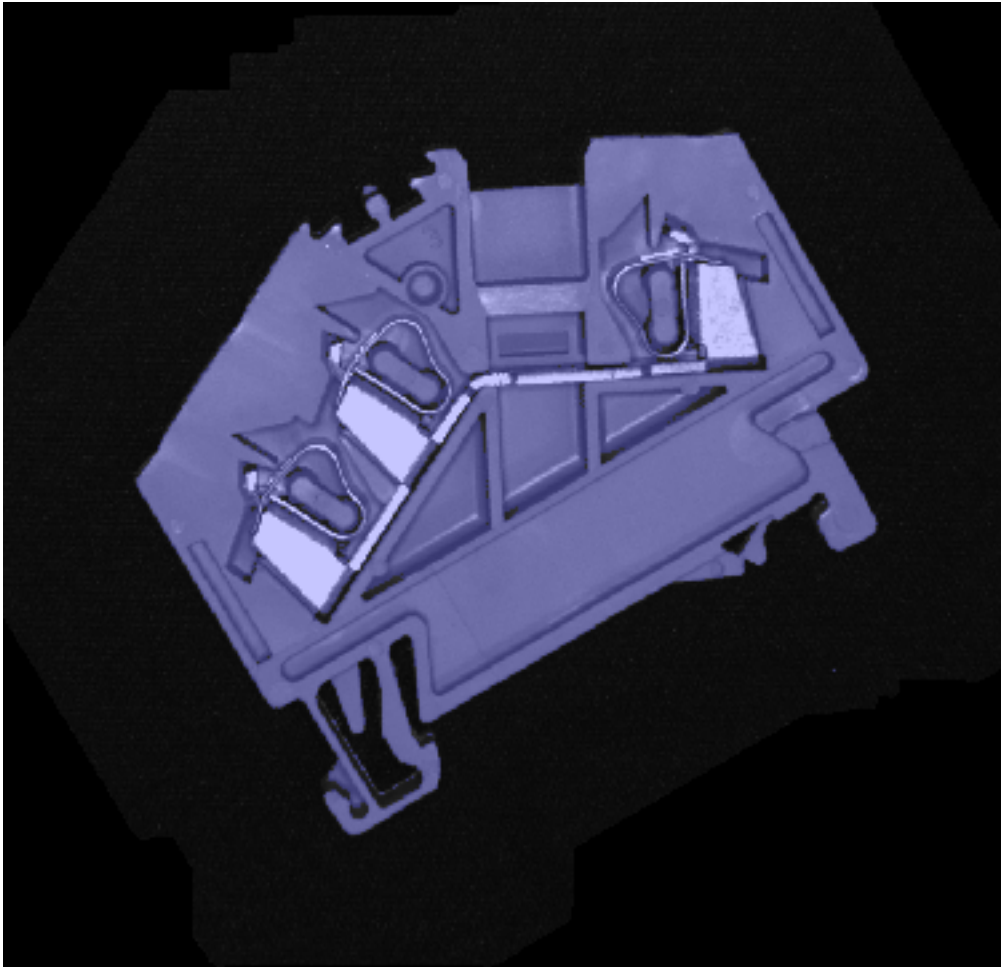
Fig. 5: The area of an object.

3.4.2 Angle

The angle of an area is the angle of the equivalent ellipse of the area.
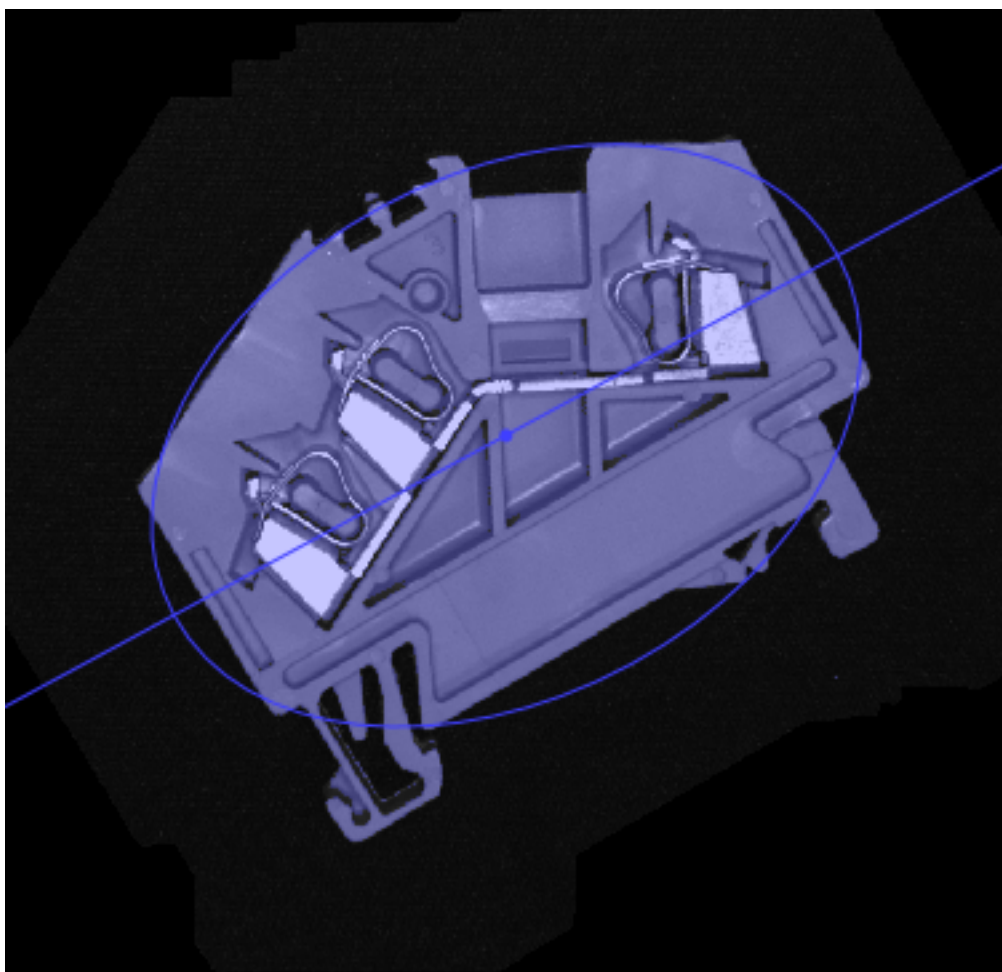
Fig. 6: The angle of an object.

3.4.3 Centroid

This calculates the object centroid.
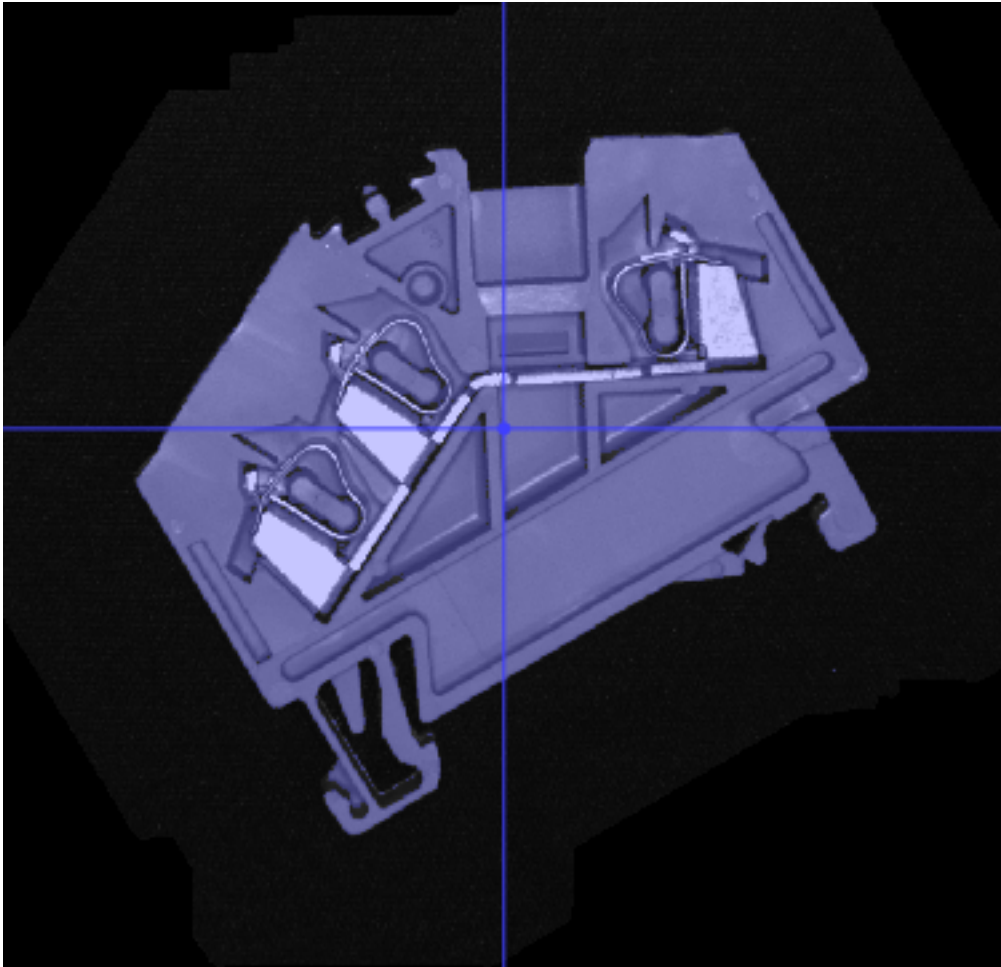
Fig. 7: The centroid of an object.

3.4.4 Top Side

This calculates the minimal y coordinate (inclusive) of the region.
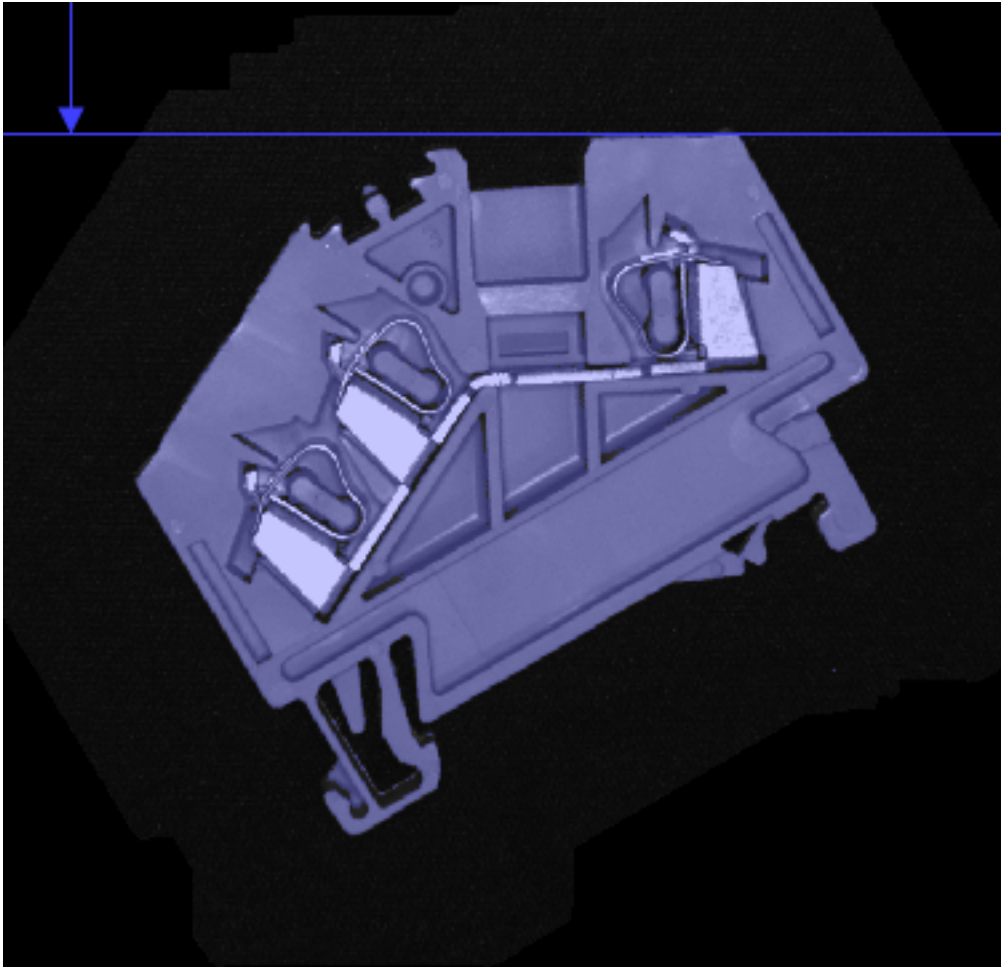
Fig. 8: The minimum vertical coordinate.

3.4.5 Right Side

This calculates the maximal x coordinate (exclusive) of the region.
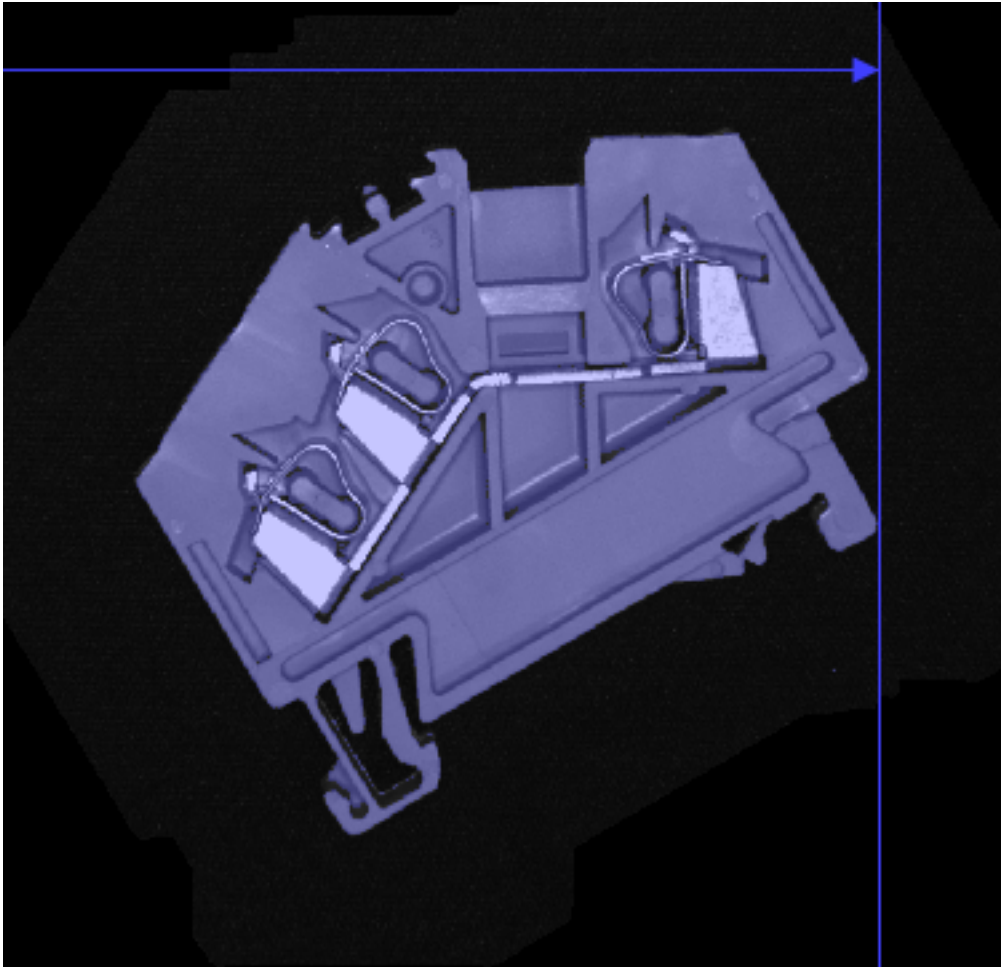
Fig. 9: The maximum horizontal coordinate.

3.4.6 Bottom Side

This calculates the maximal y coordinate (exclusive) of the region.
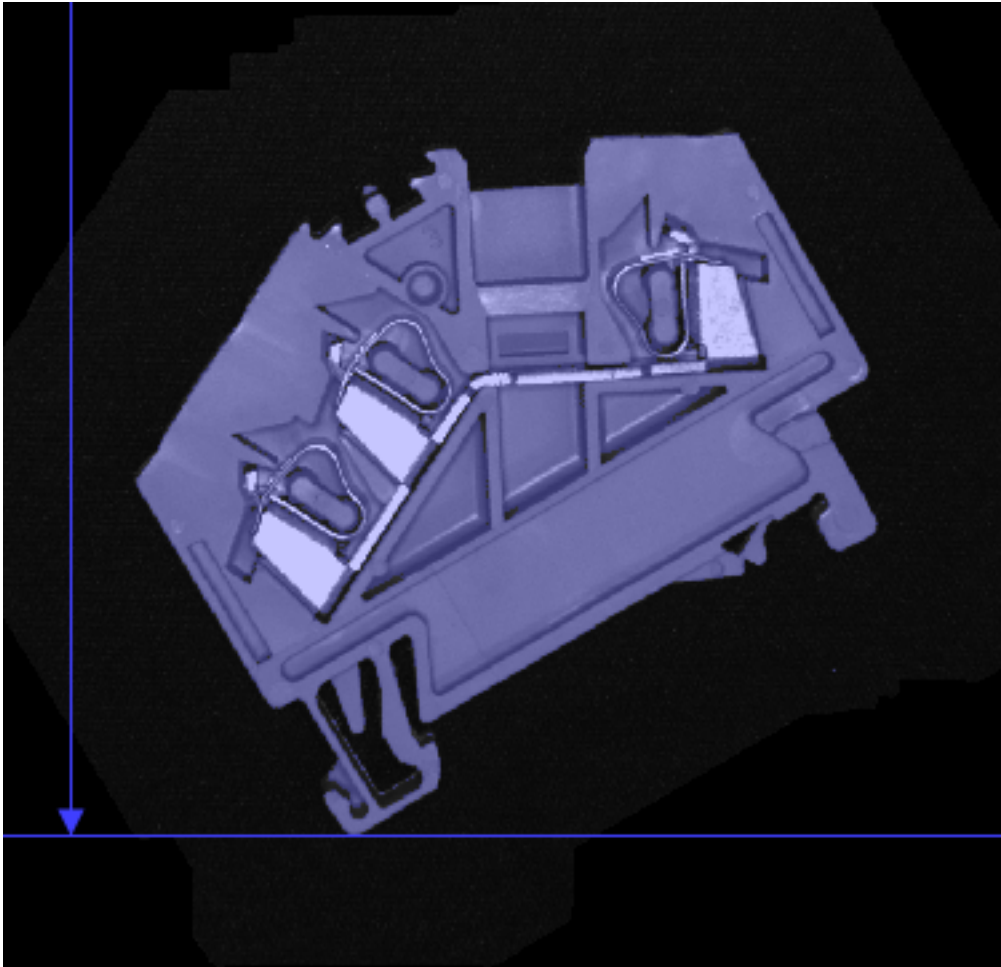
Fig. 10: The maximum vertical coordinate.

3.4.7 Left Side

This calculates the minimal x coordinate (inclusive) of the region.
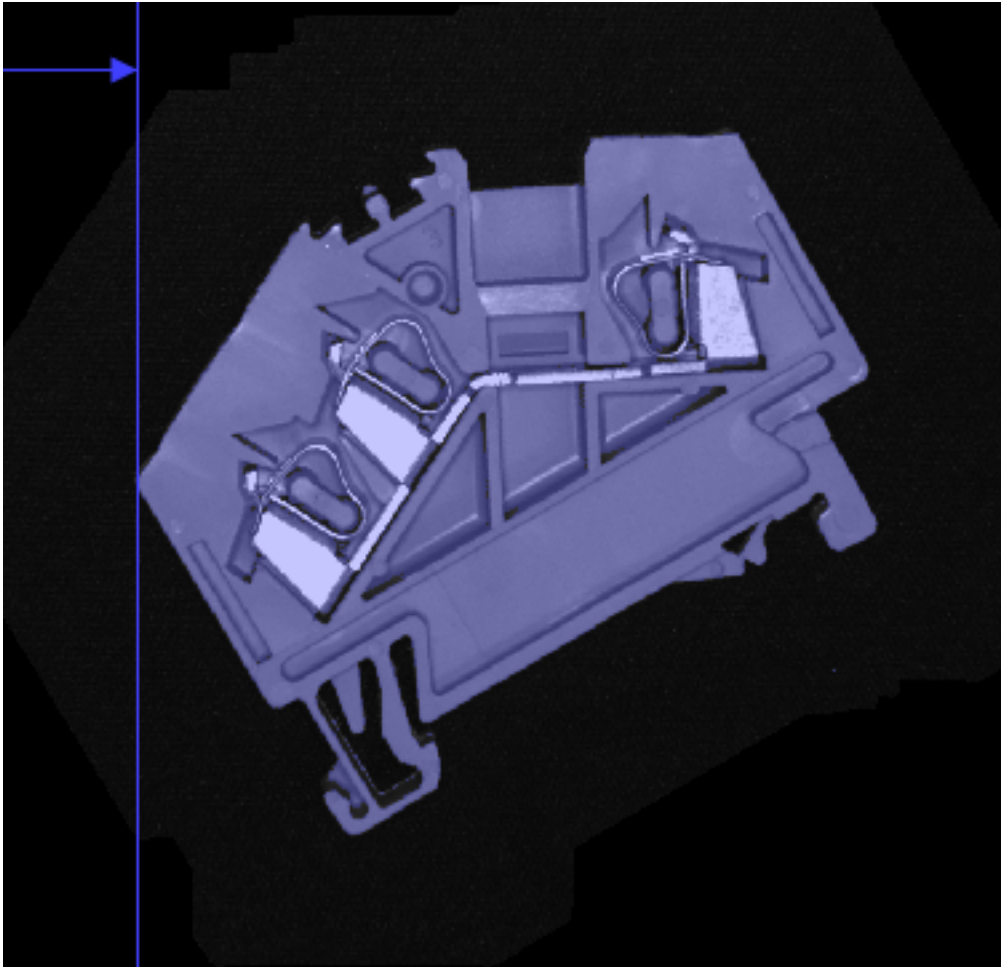
Fig. 11: The minimum horizontal coordinate.

3.4.8 Geometric Center

Calculates the center of the bounding rectangle.
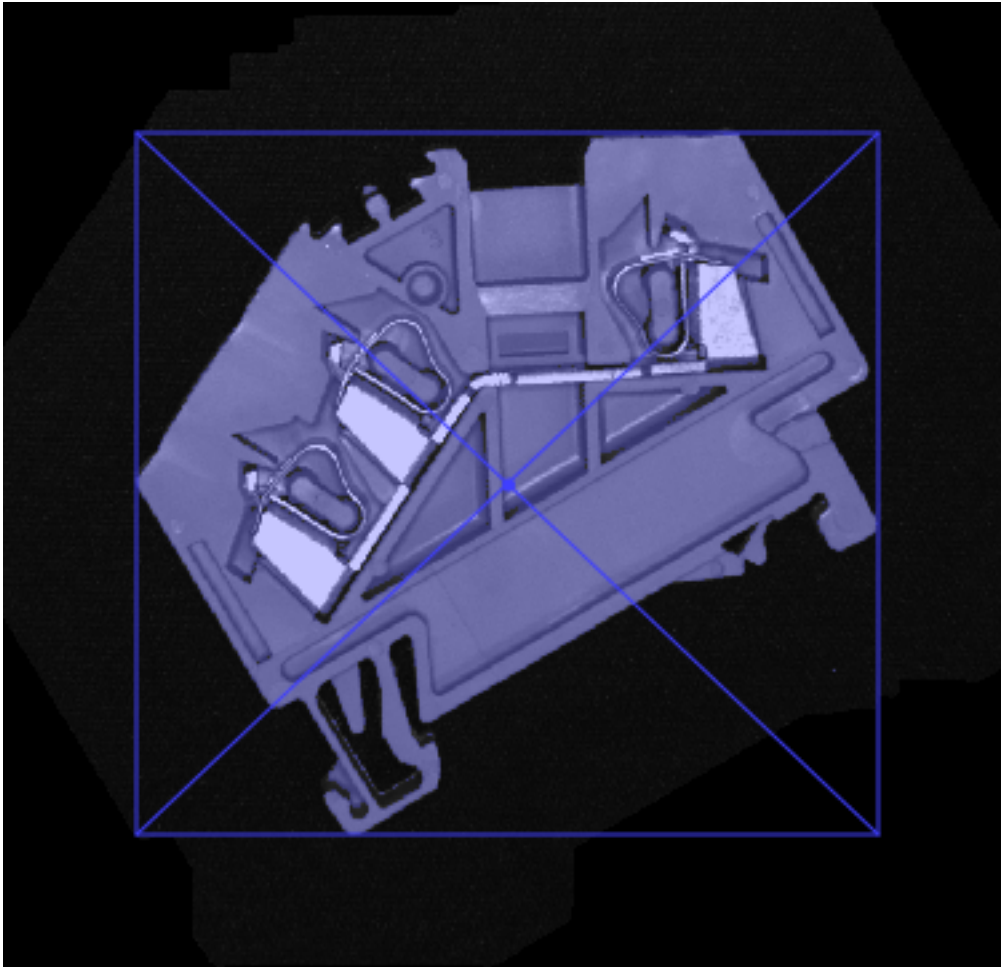
Fig. 12: The geometric center.

3.4.9 Height

This calculates the axis-parallel height of the region.
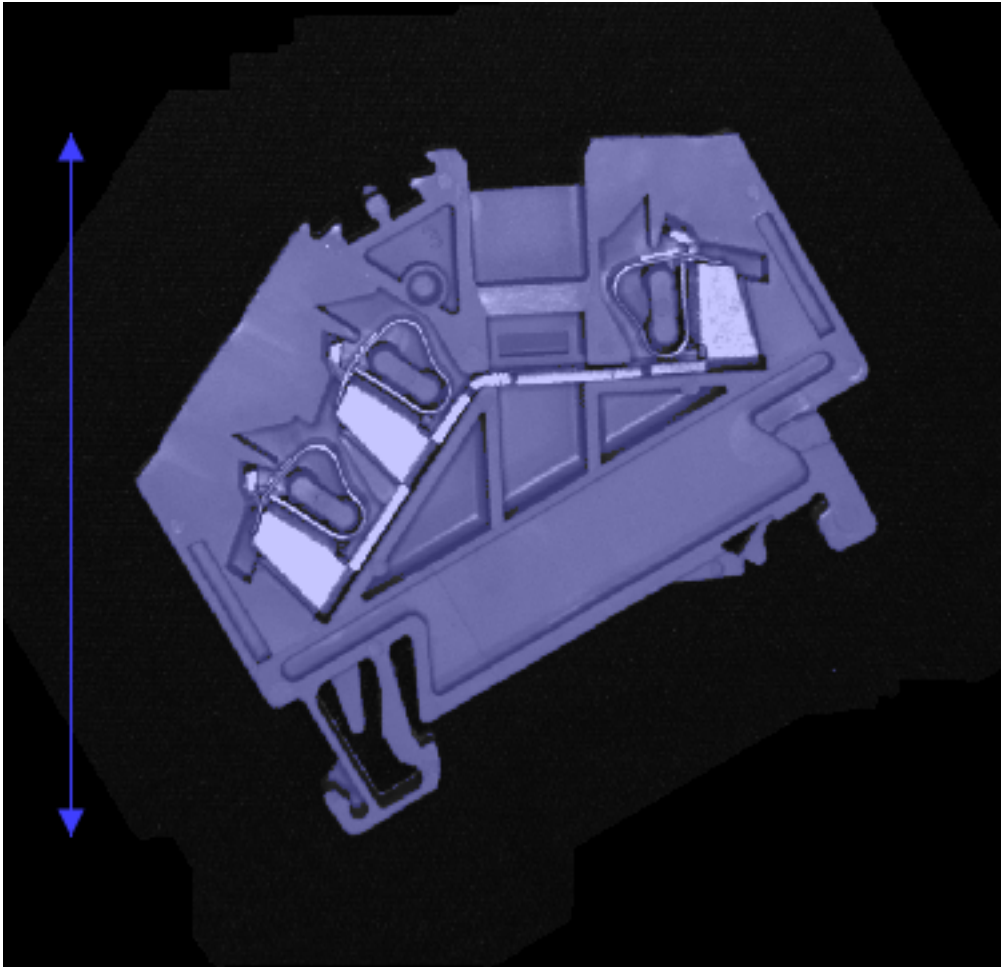
Fig. 13: The height of an object.

3.4.10 Width

This calculates the axis-parallel width of the region.
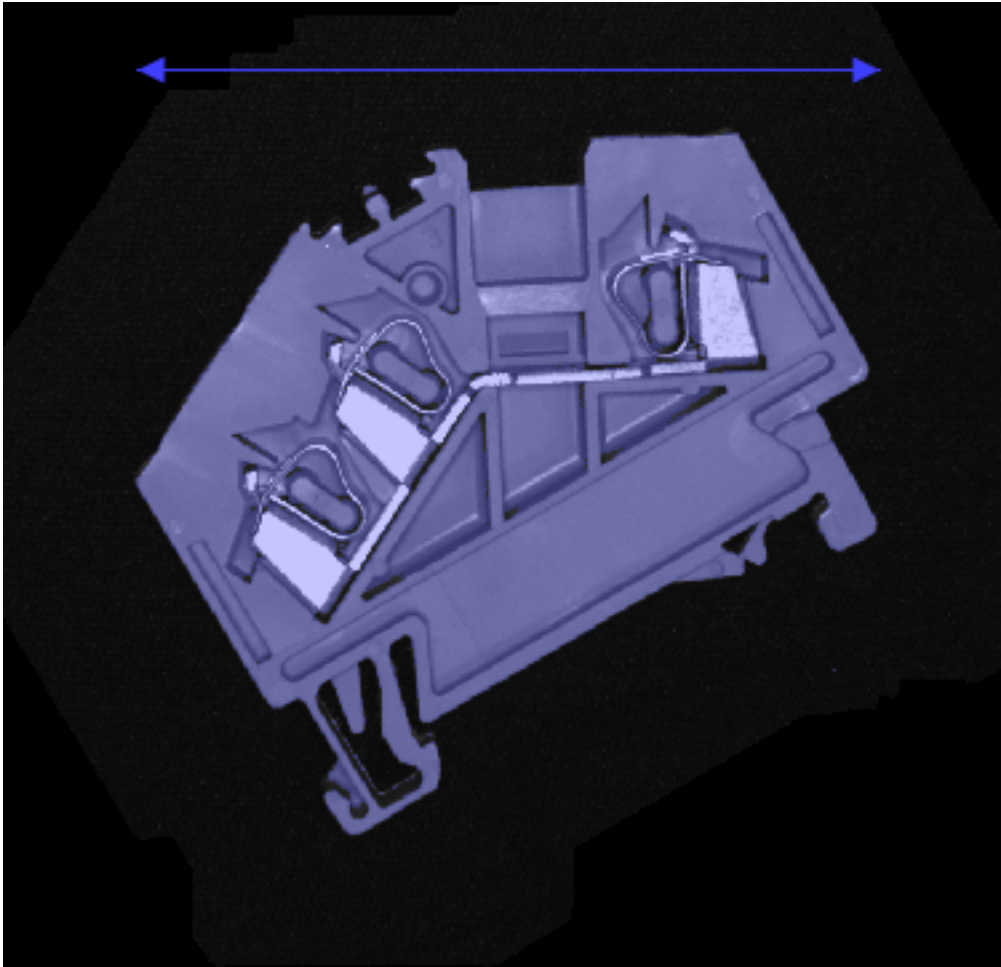
Fig. 14: The width of an object.

3.4.11 Perimeter

The perimeter of a region specifies the length of the object outline.
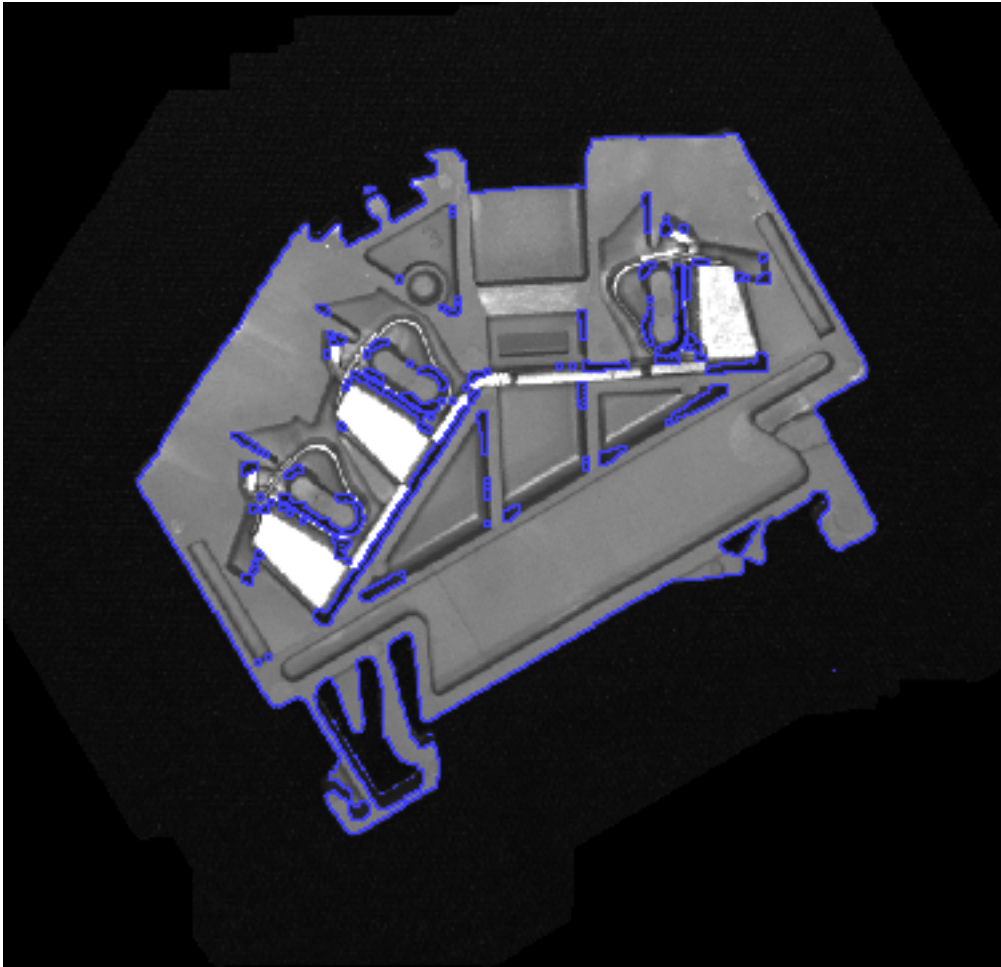
Fig. 15: The perimeter of an object.

With some objects, depending on the noise in the image acquisition, a high perimeter may result if the outline is jagged. The parameter **Tolerance** can be used to smooth this outline accordingly, resulting in a a more realistic perimeter.

3.4.12 Area of Holes

This calculates the total area of the holes of the region.
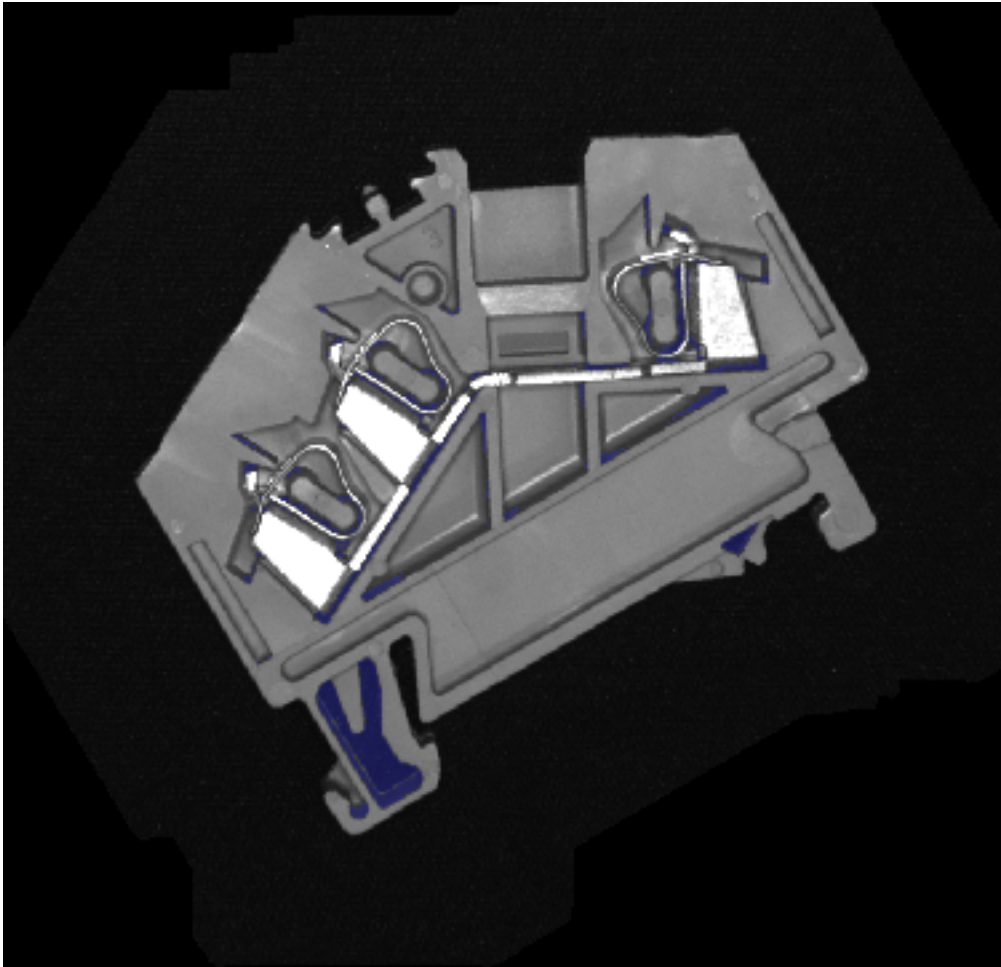
Fig. 16: The area of holes in an object.

3.4.13 Convex Area

This calculates the area enclosed by the convex hull.

The convex area is always the same or bigger than the area of the original region.
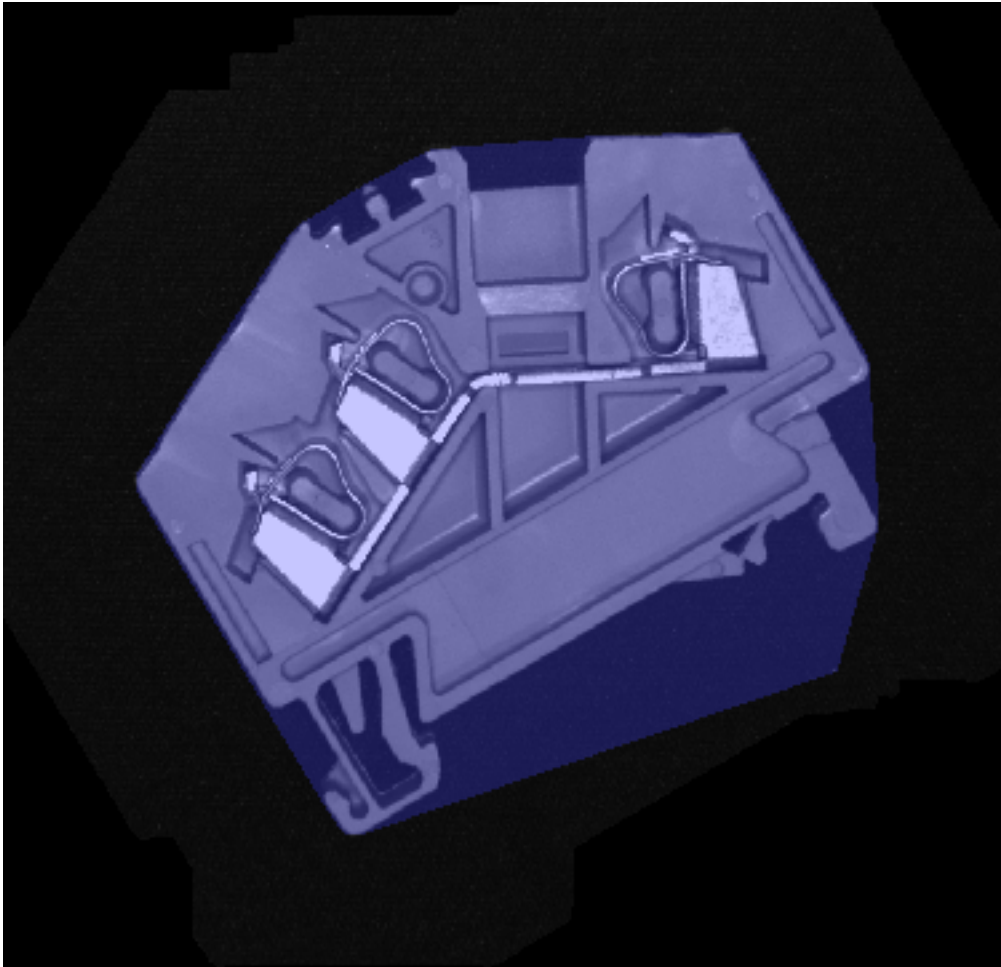
Fig. 17: The convex area of an object.

3.4.14 Convex Perimeter

This calculates the perimeter of the convex hull polygon.
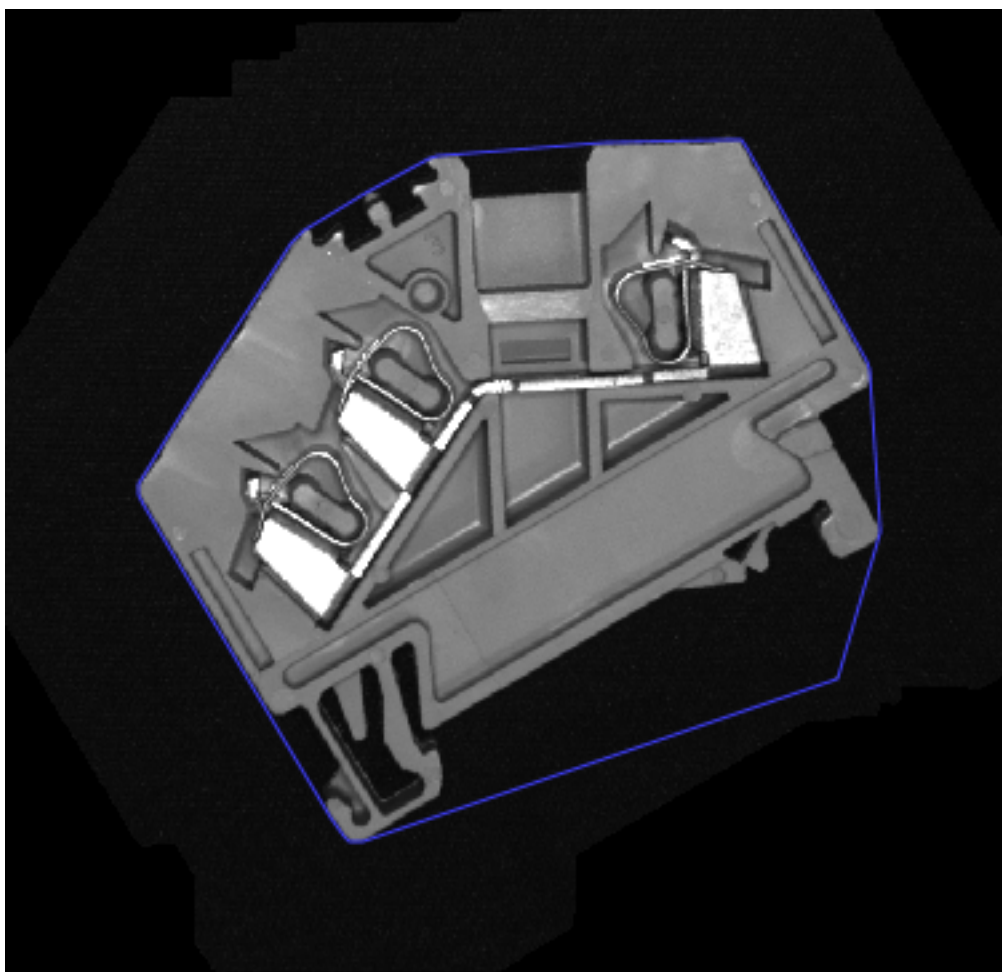
Fig. 18: The convex perimeter of an object.

3.4.15 Fiber Length

This calculates an approximation of a fiber length. This measurement is only valid for elongated objects. The fiber length is approximated as half the perimeter.

With some objects, depending on the noise in the image acquisition, a high perimeter may result if the outline is jagged. The parameter **Tolerance** can be used to smooth this outline accordingly, resulting in a a more realistic perimeter.

3.4.16 Fiber Width

This calculates an approximation of a fiber width. The fiber width is approximated as twice the maximum value of the distance transform.

3.4.17 Aspect Ratio

This calculates the width to height ratio using the following formula:

$$c = \frac{w}{h}$$

where w is the width and h is the height.

3.4.18 Circularity

This calculates the circularity of the region. The circularity is defined as the quotient of the perimeter of a circle having the same area as the object and the perimeter of the object. The circularity is calculated according to the following formula:

$$c = \frac{2\sqrt{a\pi}}{p}$$

where a is the area and p is the perimeter.

3.4.19 Compactness

This calculates the compactness of the region.

The compactness c is calculated according to this formula:

$$c = \frac{p^2}{4\pi a}$$

where a is the area and p is the perimeter.

3.4.20 Convexity

This calculates the convexity of the region.

The convexity c is calculated according to this formula:

$$c = \frac{a}{a_c}$$

where a is the area and a_c is the convex area.

The convexity is 1 for a region that already is convex. It is smaller than 1 for non-convex regions. The convexity is scale invariant.

3.4.21 Number of Holes

This calculates the number of holes of the region.

3.4.22 Perforation

This calculates the perforation of the region.

The perforation p is calculated using this formula:

$$p = \frac{a_h}{a}$$

where a is the area and a_h is the area of the holes.

The perforation is zero for an area that has no holes and increases the more holes the region has. The perforation is scale-invariant.

3.4.23 Roughness

This calculates the roughness of the region. The roughness is defined as the quotient of the perimeter of the object and the perimeter of the convex hull. The roughness is calculated according to the following formula:

$$r = \frac{p}{p_c}$$

where p is the perimeter and p_c is the perimeter of the convex hull.

3.4.24 Roundness

This calculates the roundness of the region. The roundness is defined as the quotient of the area of the object and the area of the minimum bounding circle. The roundness is calculated according to the following formula:

$$r = \frac{a}{a_c}$$

where a is the area and a_c is the area of the minimum bounding circle.

3.4.25 Sphericity

This calculates the sphericity of the region. The sphericity is defined as the quotient of the diameter of a circle having the same area as the object and the maximum feret diameter of the object. The sphericity is calculated according to the following formula:

$$s = \frac{2 * \sqrt{\frac{a}{\pi}}}{f_{max}}$$

where a is the area and f_{max} is the maximum feret diameter.

3.5 Barcode Decoding

The barcode decoder supports the important linear and 2D symbologies.

The following symbologies are supported:

3.5.1 Linear

Code 39 Code 93 Code 128 Codabar GS1 Databar RSS 14 GS1 Databar RSS 14 Stacked GS1 Databar RSS Limited GS1 Databar RSS Expanded GS1 Databar RSS Expanded Stacked Pharmacode UPC A UPC E EAN 8 EAN 13 Code 2of5 Interleaved Code 2of5 Matrix Code 2of5 Straight Code 2of5 HongKong Code 2of5 NEC Code 2of5 IATA

3.5.2 2D

Aztec Data Matrix Dotcode PDF 417 PDF 417 Micro QR 2005 QR 2005 Micro QR Model 1

3.5.3 Decoder Features

The decoder is used in three different tools: **Locate Barcode**, **Detect Barcode** and **Count Barcodes**. The three tools are largely similar, but differ in detail:

Barcode Locate can detect a code of a single symbology and provides its location for the purpose of tracking downstream tools. All parameters of the symbology can be set in detail.

Detect Barcode can detect a code of a single symbology. All parameters of the symbology can be set in detail.

Barcode Counting can detect multiple codes of different symbologies. Not all parameters can be set in detail, but standard values are used.

The decoder can calculate the following features:

Code

The content encoded in the barcode.

Symbology

The symbology used for each code.

Center (px)

The center of the decoded code in pixel coordinates.

In the logic tool, the coordinates are addressed individually as **CenterX (px)** or **CenterY (px)**.

Center (mm)

The center of the decoded code in mm. This value is only available if a calibration has been performed and applied.

In the logic tool, the coordinates are addressed individually as **CenterX (px)** or **CenterY (px)**.

Angle

The angle of the code in relation to the location of the ROI.

Size

The width and height of the code in pixels.

In the logic tool, the sizes are addressed individually as **width** and **height**.

CellSize

The number of cells in horizontal and vertical directions (only for 2D symbologies).

In the logic tool, the sizes are addressed individually as **CellWidth** and **CellHeight**.

Pixel per module

The number of pixels per module (only for 2D symbologies).

Quality (UEC)

A quality measure based on unused bits for error correction (only for 2D symbologies).

3.5.4 DPM

The decoder can decode simple direct-marked codes. For particularly difficult markings there is an optional ID-Pro module with additional decoding options.

The DPM settings only have an effect on Data Matrix and QR Codes.

3.5.5 Verification

The optional ID-Pro module also enables codes to be verified based on the ISO/IEC 15415 or ISO/IEC TR29158 standards.

Verification is only defined for Data Matrix and for QR Codes.

The decoder can detect the following features:

Grade	Description
Final	The overall grade is the lowest of all other grades.
Axial Non-Uniformity	Asseses the uneven scaling of the code.
Contrast	Asseses the difference between the darkes and brightest modules.
Fixed Pattern Damage	Asseses errors in L-shape, clock pattern and quiet zone.
Grid Non-Uniformity	Asseses the deviation from the grid.
Modulation	Asseses local deviations of contrast.
Reference Decode	The reference decoding algorithm: 4 for success, otherwise 0.
Reflectance Margin	Asseses how well each element is correctly distinguishable.
Unused Error Correction	Asseses bits unused for error correction.

The grades are integer numbers ranging from 0 (bad) to 4 (good). Sometimes letters are also used for designation: A = 4, B = 3, C = 2, D = 1 and F = 0, according to the American grading system.

Optical Character Recognition (OCR)

Optical character recognition is the technical term for identifying characters, i.e. reading text.

After segmentation of the symbols into regions representing the characters, a classifier is applied to get the interpretation of each character. A set of fonts is provided, which are based on a large amount of training data from various application areas. These fonts allow you to read text in documents, on pharmaceutical or industrial products, dot prints, and even handwritten numbers.

Our OCR reader assumes approximately horizontal text. If the text is not horizontally aligned, the ROI should be rotated so that the text within the ROI is horizontal with respect to the orientation of the ROI.

There are several groups of fonts. The members of the individual groups differ as they contain different. The content of a font is described by its name. For the names of the pre-trained fonts, the following nomenclature applies:

The name begins with the group name, e.g. **Document** or **DotPrint**, followed by indicators for the symbol set which are contained in the OCR font. The meaning of the indicators is as follows:

- **0-9**: The OCR font contains the digits 0 to 9.

- **A-Z**: The OCR font contains the capital letters A to Z.

- **+** : The OCR font contains special characters. The list of special characters varies slightly between the individual OCR fonts. It is listed individually for each OCR font below.

If the name of the OCR font does not contain any of the above characters, the OCR font normally contains the digits 0 to 9, the upper case letters A to Z, the lower case letters lower case letters a to z and special characters. Some OCR fonts do not contain lowercase letters (e.g., DotPrint). This is explicitly mentioned in the description of the respective font explicitly mentioned.

Document

The font `Document` can be used to read characters printed in fonts such as Arial, Courier or Times New Roman. These are typical fonts for printing documents or letters.

Note that the characters I and l of the Arial font cannot be distinguished. This means that an `l` can be mistaken for an `I` and vice versa.

Available special characters: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

The DotPrint OCR font can be used to read characters printed with dot printers. It does not contain lower case letters.

Available special characters: - / . * :
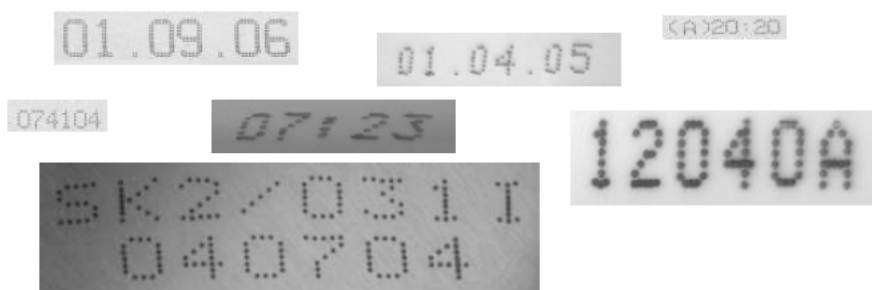
Fig. 19: Examples of dotted font.

HandWritten

The HandWritten_0-9 font can be used to read handwritten numbers.

It contains the digits 0-9.

Available special characters: none
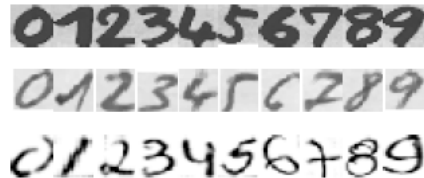
Fig. 20: Examples of handwritten digits.

Industrial

The Industrial font can be used to read characters written in fonts such as Arial, OCR-B or other sans serif fonts. other sans serif fonts. These fonts are normally used for printing z. labels, for example.

Available special characters: - / + . $ % * e £ ¥
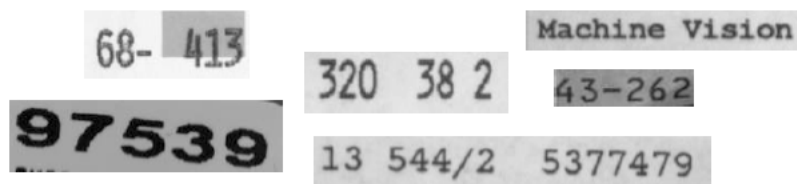
Fig. 21: Examples of industrial prints.

OCR-A

The OCR-A font can be used to read characters that are printed in the OCR-A font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

```
0123456789
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrtsuvwxyz
-?!/\=+<>.#$%&()@*
```

Fig. 22: Selected characters of the OCR-A font.

OCR-B

The OCR-B font can be used to read characters that are printed in the OCR-B font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

```
0123456789
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrtsuvwxyz
-?!/\=+<>.#$%&()@*
```

Fig. 23: Selected characters of the OCR-B font.

Pharma

The Pharma font can be used to read characters printed in fonts such as Arial, OCR-B and and other fonts typically used in the pharmaceutical industry.

This OCR font does not contain lowercase letters.

Available special characters: - / . ( ) :

Fig. 24: Examples of pharmaceutical prints.

SEMI

The SEMI font can be used to read characters that are printed in the SEMI font, which consists of characters that are designed in such a way that they can be easily distinguished from each other. It has a limited set of characters.

This OCR font does not contain lowercase letters.

Available special characters: - .

Fig. 25: Character set of the SEMI font.

Universal

The Universal OCR font can be used to read a variety of different characters. The training of this CNN-based font is based on the character sets 'Document', 'DotPrint', 'SEMI' and 'Industrial'.

Available special characters: - / = + : < > . # $ % & ( ) @ * e £ ¥

Fig. 26: Example for the Universal character set.

REFERENCE

4.1 GUI

4.1.1 Ribbon

The ribbon shows commands used to control **nVision-i**.
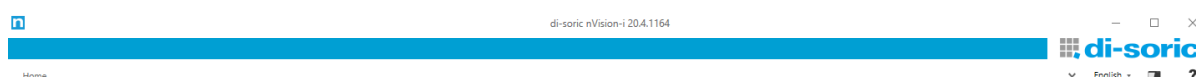
Fig. 1: The nVision-i ribbon

At the right of the ribbon, below the **di-soric** logo, there are three buttons and the language selector.

Language

Select the language.

Toggle Theme

Toggle the theme between white, light and dark modes.

Help : F1

Show/Hide the help window.

Home

If **nVision-i** is connected to a device and a job has been opened for editing, the **Home** tab shows the available commands:
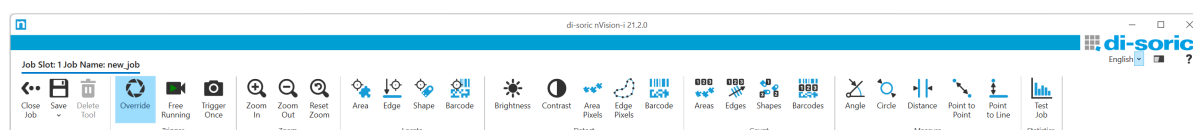
Fig. 2: The ribbon for a **CS-60**

The available commands are different for different product lines. If a job is edited on a **CS-60**, all commands are available. If a job is edited on an **ID-600**, only the commands for barcodes are available.

 Close Job

Goes back to the main page.

 Save Job

Save a job permanently.

 Save Image : Ctrl-S

Save an image.

 Delete Tool : Del

Delete the selected tool.

Trigger

 Override

Take control over triggering.

 Free Running

Trigger the image acquisition continuously.

 Trigger Once

Trigger the image acquisition once.

Zoom

Zoom In : Num-Plus

Zoom into an image.

Zoom Out : Num-Minus

Zoom out of an image.

Reset Zoom

Reset the zoom.

Locate

On a **CS-60**, all commands in the group **Locate** are available. On an **ID-600**, only the **Locate Barcode** tool is available.

Area

Insert the locate area tool.

Edge

Insert the locate edge tool.

Shape

Insert the locate shape tool.

Barcode

Insert the locate barcode tool.

Detect

On a **CS-60**, all commands in the group **Detect** are available. On an **ID-600**, only the **Detect Barcode** tool is available.

Brightness

Insert the detect brightness tool.

Contrast

Insert the detect contrast tool.

Area Pixels

Insert the detect/count area pixels tool.

Edge Pixels

Insert the detect/count edge pixels tool.

Barcode

Insert the detect barcode tool.

Count

On a **CS-60**, all commands in the group **Count** are available. On an **ID-600**, only the **Count Barcodes** tool is available.

Areas

Insert the count areas tool.

Edges

Insert the count edges tool.

Shapes

Insert the count shapes tool.

Barcodes

Insert the count barcodes tool.

Measure

The commands in the **Measure** group are only available on a **CS-60**.

Angle

Insert the measure angle tool.

Circle

Insert the measure circle tool.

Distance

Insert the measure distance tool.

Point to Point

Insert the measure point to point tool.

Point to Line

Insert the measure point to line tool.

Statistics

📊 **Test Job**

Provides statistics about a job.

4.1.2 Statusbar

The statusbar shows information about the sensor and a job.
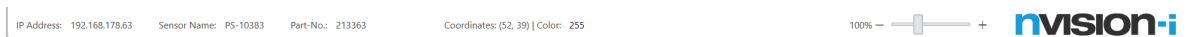
Fig. 3: The statusbar

At the left of the statusbar, the IP-Address, the name of the sensor as well the name and the slot of the job are shown.

In the middle of the statusbar, the position of the mouse cursor in pixels and the pixel value as well as the color under the mouse cursor is shown.

The slider at the right of the statusbar zooms the controls of the application. To see more, zoom down to 50 %. To use nVision-i on touch devices, zoom up to 200 %.

4.1.3 Test Job

Fig. 4: The Icon of the **Test Job** command.

The **Test Job** command measures the execution time on the sensor. The execution time with a connected nVision-i is always slightly higher than with a free-running sensor due to communication overhead.

Settings
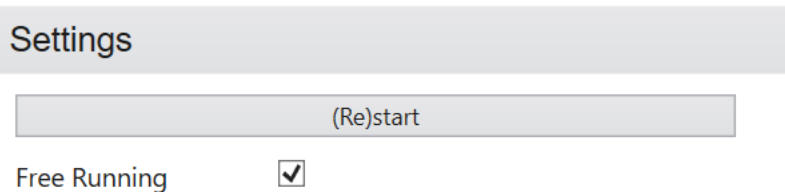
Fig. 5: The Settings of the **Test Job** command.

The **(Re)start** button starts the data collection for the statistics displayed on the right. As long as this is active, every inspected image is included and displayed in the statistics.

The **Free running** option is activated as by default. In this state, the set trigger is ignored and a new image acquisition is started automatically after an image has been inspected. If this option is deactivated, the job is tested with the given trigger setting. Depending on the setting, a new evaluation is started after a digital or PROFINET trigger is received.

Statistics
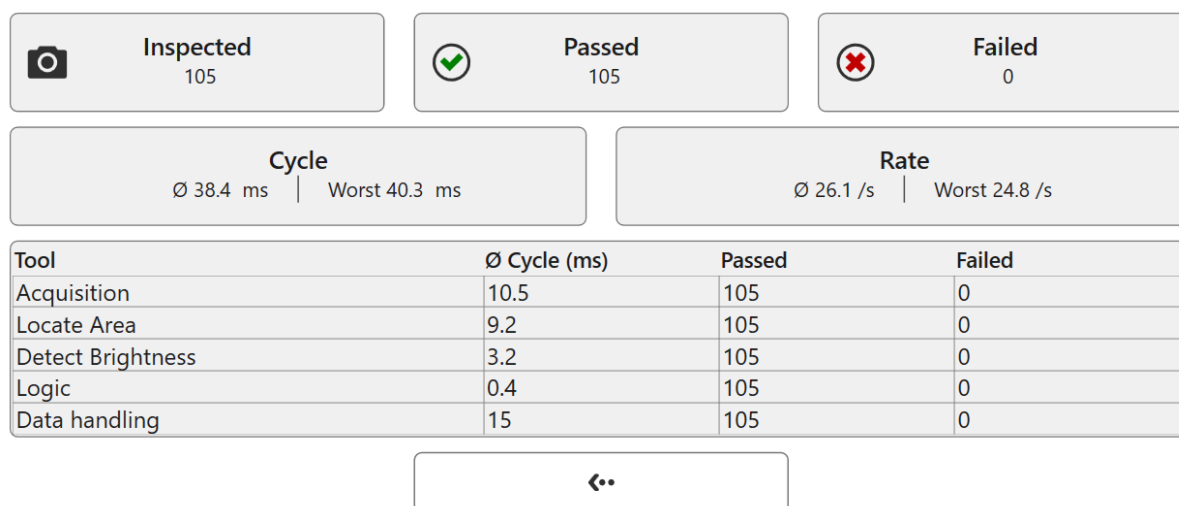
Fig. 6: The statistics of the **Test Job** command.

The current statistics are displayed on the right side.

In the first line the number of images evaluated so far is displayed, as well as the information how many of them passed or failed the evaluation.

The second line shows the average as well as the longest time spent on the inspection of the images inspected so far. The **Cycle** time indicates the absolute duration in milliseconds. The **Rate**, on the other hand, indicates how many inspections can be performed per second.

In the third column, the currently active tools are listed separately. Here, an average cycle time per tool as well as the absolute number of passed or failed inspections is listed.

With the button in the fourth column the **Test Job** display can be closed. The collected statistics are reset after closing and can not be displayed again.

4.2 Correction Tools

4.2.1 Shading

The shading correction compensates for the decrease in brightness in a captured image towards the edge (also called vignetting). The shading is an optical effect that varies with the lenses used and is also caused or intensified by the lighting. by the illumination. A prerequisite for optimal shading correction is the orthogonal viewing of a scene (the camera must look at the scene at a right angle).
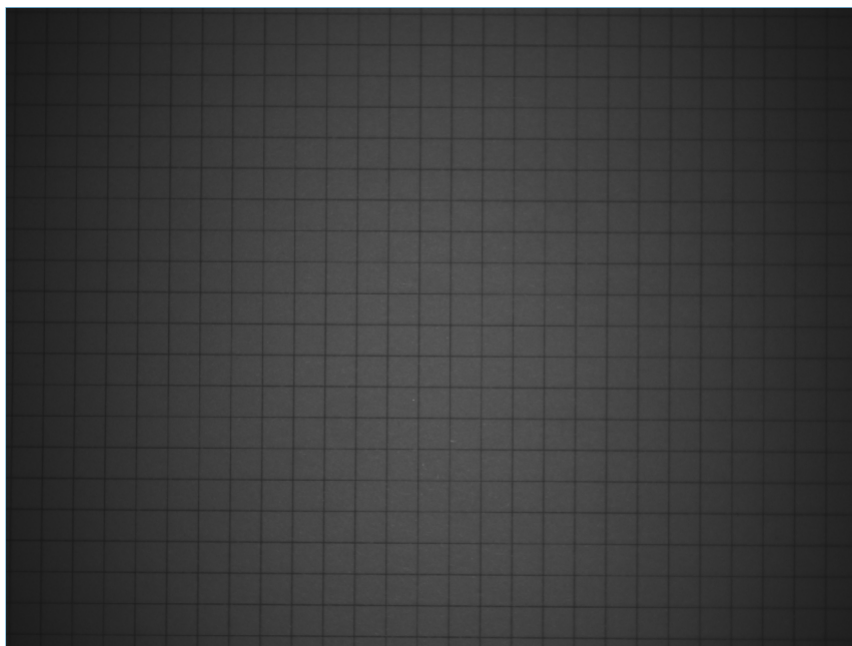
Fig. 7: Edge shading.

The calibration should be performed anew each time the lens is changed or the illumination is changed.

For calibration, an image should be taken that is as uniformly bright as possible, making sure that the intensities are neither too dark nor too bright. This can be done with the histogram display or with the functions **Highlight Underexposure** and **Highlight Overexposure** from the **Visualization** area.

If the correction image and the parameters for exposure and illumination are set correctly, the **Correct** button can be used to calculate the correction. correction can be calculated.
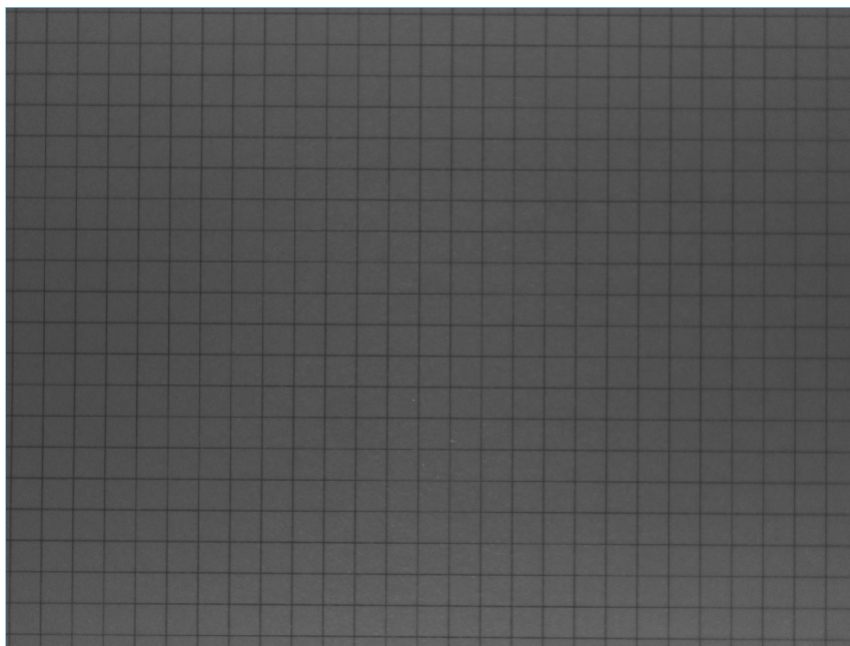
Fig. 8: Corrected shading.

After the calculation has been completed, the correction can be saved with **Save correction** so that it can be applied in jobs, or deleted again by the **Reset** button.

Parameters

The histogram shows the gray value distribution of the captured images.

Exposure Time controls the exposure time of a single image, starting at 25 µs and going up to 2000 µs. When the internal flash is disabled (under **Flash Settings**, **Internal**), the **Exposure Time** can be selected up to a maximum of 500000 µs. The exposure time affects the brightness of the captured image.

Gain controls the gain between 1 (no gain) and 16. The gain has a linear effect on the brightness, i.e. an image taken with gain 2 has twice as high intensity values as an image taken without gain.

Flash Control

The **CS-60/{ID600_NAME}}** vision sensor has built-in illumination. When **Internal** is set to **On**, four high-power LEDs illuminate the scene. The illumination is synchronized with the image capture. When **Internal** is set to **Off**, the built-in LEDs are off and do not illuminate the scene, for example when external lighting is used. The **Color** can be set to **White** or **Red** to use the respective LED color.

In addition to the internal flash, a digital output can also be selected under **External** to control external lighting. Available for selection are all Digital Outputs which are not used in the **Logic**

tool under **Digital Outputs** (whose input **Value** are not connected). Conversely, Digital Outputs that have been selected as External Flash can no longer be used in the **Logic** tool.

Settings

With **Smoothing** an image smoothing between **Sharp** and **Blurred** can be selected, as a pre-processing step for the calculation of the shading correction.

Visualization

Corrected image displays an image corrected in brightness.

Original Image displays the original, uncorrected image.

Highlight Overexposure shows overexposed parts of the image in yellow color.

Highlight Underexposure shows underexposed parts of the image in blue color.

4.2.2 Distortion

Distortion correction compensates for optical distortions that occur due to different lenses in varying degrees towards the edge. A prerequisite for optimal distortion correction is the orthogonal view on a scene (the camera must look at the scene at a right angle). A second effect of the distortion correction is the determination of the intrinsic and extrinsic camera parameters, as far as they are necessary for a calibrated measurement of an orthogonal measuring plane.

Calibration must be performed anew each time the lens is changed or the measurement plane is changed.

For the calibration, an image of a point-shaped calibration target must be taken, filling the image field as far as possible, and the point distance must be entered. Such targets are available for purchase, or they can be printed out by yourself with a printer for less precise requirements. In the **Docs** directory of the nVision-i installation there is a PDF file to print out for different image field sizes.

50 x 33 2mm

.
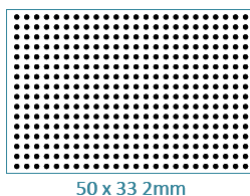
The calibration target must be placed in the measurement plane (only in this measurement plane the measured values are valid). With the endocentric lenses used, image parts that are further away are imaged smaller and image parts that are closer are imaged larger, i.e. the measured values are also correspondingly smaller or larger. Correct measured values are only determined in the measuring plane.

Parameters

The histogram shows the gray value distribution of the captured images.

Exposure Time controls the exposure time of a single image, starting at 25 µs and going up to 2000 µs. When the internal flash is disabled (under **Flash Settings**, **Internal**), the **Exposure Time** can be selected up to a maximum of 500000 µs. The exposure time affects the brightness of the captured image.

Gain controls the gain between 1 (no gain) and 16. The gain has a linear effect on the brightness, i.e. an image taken with gain 2 has twice as high intensity values as an image taken without gain.

Flash Control

The **CS-60/ID-600** vision sensor has built-in illumination. When **Internal** is set to **On**, four high-power LEDs illuminate the scene. The illumination is synchronized with the image capture. When **Internal** is set to **Off**, the built-in LEDs are off and do not illuminate the scene, for example when external lighting is used. The **Color** can be set to **White** or **Red** to use the respective LED color.

In addition to the internal flash, a digital output can also be selected under **External** to control external lighting. Available for selection are all Digital Outputs which are not used in the **Logic** tool under **Digital Outputs** (whose input **Value** are not connected). Conversely, Digital Outputs that have been selected as External Flash can no longer be used in the **Logic** tool.

Distortion

The **Threshold** for the detection of the calibration target can be determined **Automatically** or set via a slider. The aim is to optimally determine the points of the calibration target.
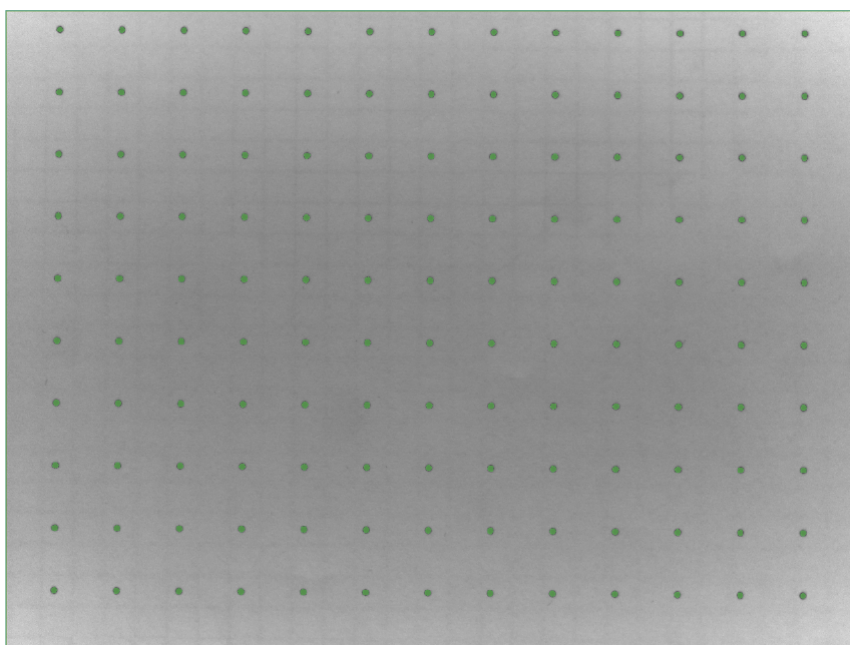
Fig. 9: Image of a target for geometric calibration.

The **Distance** is the distance between two points of the target (from center to center). This value must match the actual distance of the points on the target in order to calibrate correctly.

The region in the image should be used to narrow down the relevant points of the calibration target.

For correct setting the **Raster** can be selected to show the grid lines between the points of the target.
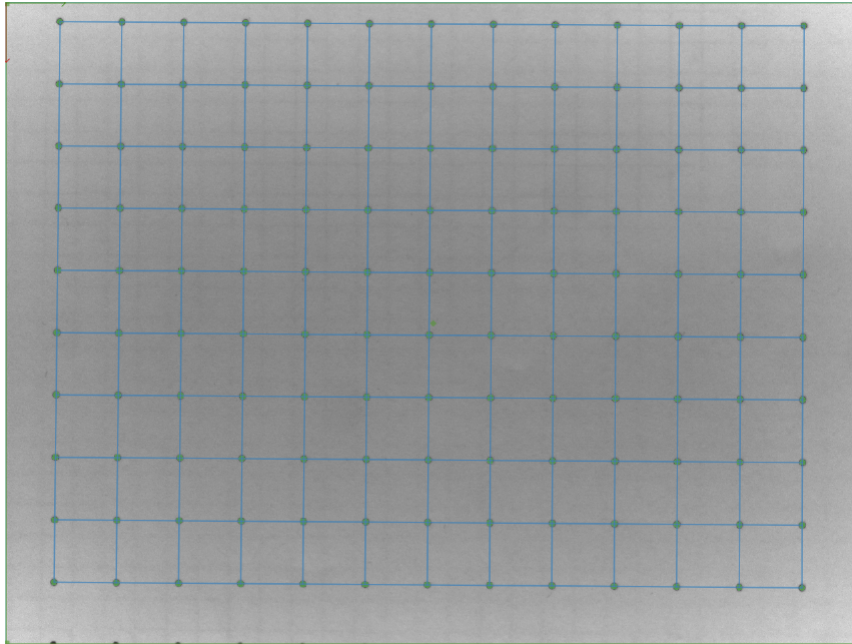
Fig. 10: Target with grid lines.

If the correction image and the parameters for exposure and illumination are set correctly, the **Correct** key can be used to calculate the correction.

The **calibration errors** can be displayed to judge the quality of the calibration.
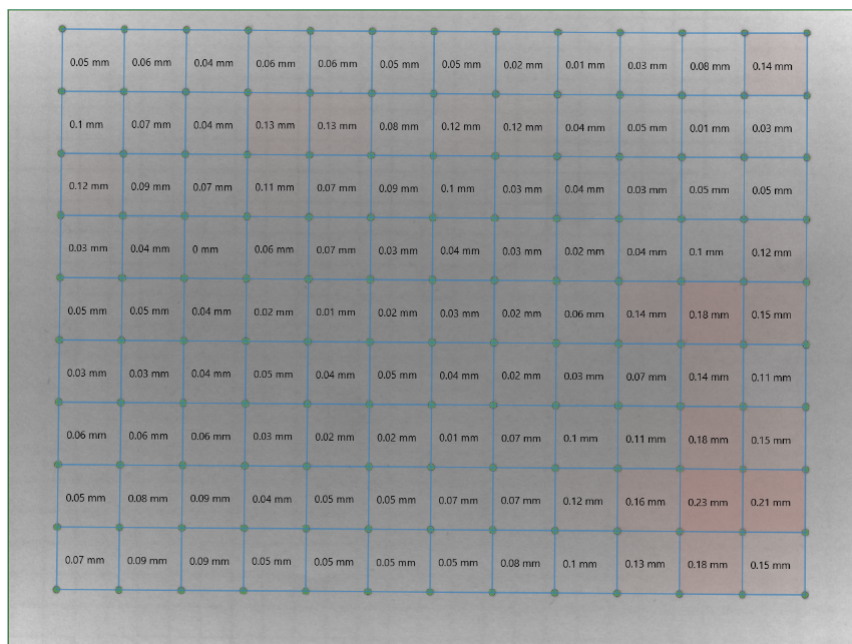
Fig. 11: Target with calibration error display

After the calculation has been completed, the correction can be saved with **Save Correction** so that it can be applied in jobs, or deleted again by pressing the **Reset** button.

Calibration Target

The **Polarity** can be set **Dark to Light** or **Light to Dark**, according to the conditions of the target used.

Areas outside **Minimum Area** and **Maximum Area** can be filtered out to avoid interference with the pattern.

Visualization

Corrected Image displays a rectified image.

Original image displays the original, non-rectified image.

Raster displays raster lines for a detected target.

Calibration error displays calibration error values.

4.3 Tools

4.3.1 Acquisition - Image

Fig. 12: The **Acquisition** tool icon.

The **Acquisition** tool is used to set the parameters of the image acquisition.

The **Acquisition** tool is the first tool in any pipeline, is always there and cannot be deleted. In this respect, the **Acquisition** tool is special. In any inspection, acquisition is the first step in a processing pipeline.

Fig. 13: The **Acquisition** tool tile.

The **Acquisition** tool shows the flash setting as well as the acquisition Id. If the **Acquisition** tool is selected in the pipeline, it will display an extended GUI at the left. At the right, the tool will display the acquired image.

Parameters

With **Image Directory** you can select a you can choose a directory containing images. You must make sure that the directory contains only monochrome images with a bit-depth of 8 Bit per Pixel, how they are exported by nVision-i. Otherwise, images may not load correctly. This control is only shown, when the emulator is used.

The histogram displays the grey-value distribution of the acquired images.

Exposure Time controls the exposure period of a single frame, starting at 25 µs and ranging up to 2000 µs. With internal flash disabled (under **Flash Control**, **Internal**), the **exposure time** can be selected up to a maximum of 500000 µs. The exposure time affects the brightness of the acquired image.

Gain controls the gain between 1 (no gain) and 16. The gain acts linearly on the brightness, i.e. an image captured with gain 2 has twice as high intensity values as an image taken without gain.

Flash Control

The **CS-60/ID-600** vision sensor has built in illumination. When **Internal** is set to **On**, four high power LEDs illuminate the scene. The illumination is synchronized with the image acquisition. When **Internal** is set to **Off** the built-in LEDs are off and do not illuminate the scene, e.g. if you want to use external illumination. The **Color** can be set to **White** or **Red** to use the respective LED color.

In addition to the internal flash, a digital output can also be selected under **External** to control an external lighting. All digital outputs which are not used in the **Logic** tool under **Digital Outputs** (whose input **Value** is not connected) are available for selection. Conversely, digital outputs selected as external flash can no longer be used in the **Logic** Tool.

Trigger

The **Inspection Mode** can be set to **Standard** or **Gatee**. In **Standard** mode, an inspection cycle consists of capturing an image and inspecting the captured image. In **Gated** mode, images are continuously captured and inspected as long as the gate is open. The gate is opened or closed by digital triggers or commands on the command interface. The individual cycles are executed in free-running sequence as long as the gate is open.

The trigger **Type** can be set to **Digital**, **Free Running** or **Virtual** in **Standard** mode; **Digital** or **Virtual** can be selected in **Gated** mode. If the trigger is set to **Digital**, no linking of the parameters of the acquisition tool is possible; corresponding settings are ignored.

Activation in **Gated** mode can be set with **Pattern**. **Start Stop Level** opens the gate with a signal on the trigger input and keeps it open as long as the signal is present. When the trigger signal is no longer present, the gate is closed. With **Start Stop Pulse**, the gate is opened with the first pulse on the trigger input and closed with the next pulse. With **Start Pulse**, the gate is opened with a pulse on the trigger input and remains open until another condition closes it: a successful inspection with the result OK, a timeout (the time can be set with **Close Timeout (ms)** or a number of inspections (is set with **Max # of Acquisitions**)).

If the trigger is set to **Digital**, the image acquisition or opening of the gate is triggered by an external trigger signal (on the connector). The **Polarity** of the trigger signal can be set to **Rising edge** or **Falling edge** to select the appropriate signal edge. The **Delay** can be used to specify a delay between the trigger signal and the actual start of detection.

If the trigger is set to **Free running**, the **CS-60/ID-600** will trigger istelf after any inspection. If the trigger type is set to **Virtual**, triggering by Profinet, EtherNet/IP or TCP/IP commands is activated.

Image correction

Image correction includes both **Shading** correction to compensate for the drop in brightness and **Distortion** correction to compensate for geometric distortion. Both effects are non-linear and are more pronounced towards the edges of the image. The strength of the effects depends on the lens used and, in the case of shading, also on the lighting.

The corrections are complex and can therefore be optionally selected or deselected. By default, they are switched off or deselected.

The corrections are only applicable if a previous calibration step has been performed.

Visualization

Highlight Overexposure displays overexposed parts of the image in yellow color.
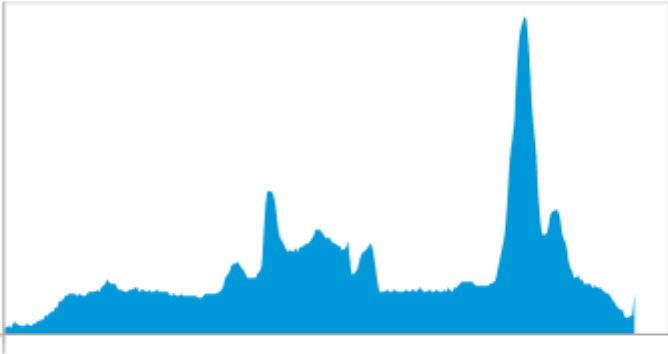
Highlight Underexposure displays underexposed parts of the image in blue color.

Fig. 14: The **Acquisition** tool parameters.

4.3.2 Acquisition - Digital Inputs

Many tool parameters can be changed by selecting or entering values. Most parameters can also be **Linked** - e.g connected with digital inputs.

Linking a value is done in two steps: first, the value on the tab **Digital Inputs** of the tool **Acquisition** is named. Then the named value can be used as a paramter using the ⟷ button.

Input data is captured whenever an image is captured. This happens at the beginning of an inspection cycle.

Linking Digital Inputs

Digital inputs are available for linking only if they are not used for job-change. This can be set for each sensor within nVision-i.

Fig. 15: **CS-60/ID-600**, settings for digital inputs

With **Job Change** the digital inputs are used for a job change, with **Linking** the values of the digital inputs can be read.
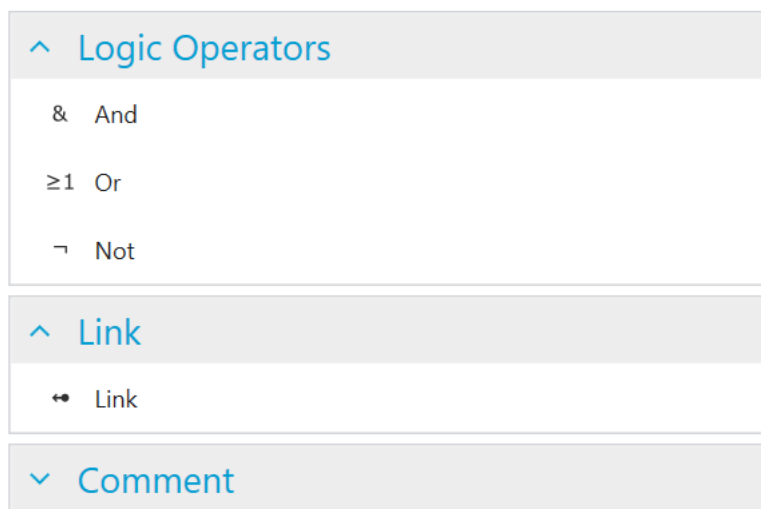
Fig. 16: The context menu on the left.

The **CS-60/ID-600** vision sensor has two digital inputs that can be used for linking. The input nodes are displayed on the **Digital Inputs** page of the **Acquisition** tool. In order to use them for linking, they must be named. An input is named by adding a **Link** node (by selecting them in the context menu on the left or with a right click on the canvas and selecting **Link**), connecting the input node with the **Link** node and typing the respective name in the **Link** node.
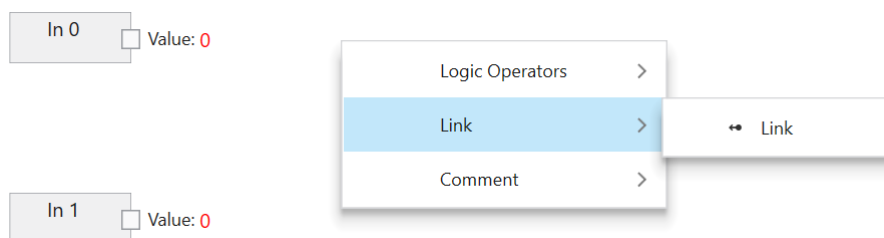
Fig. 17: Add a **Link** node with right click
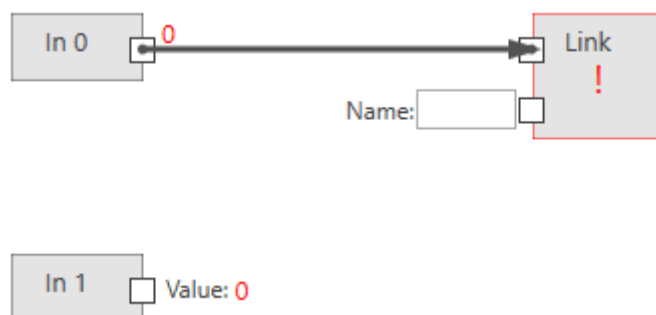
Fig. 18: Connect the input node with the **Link** node
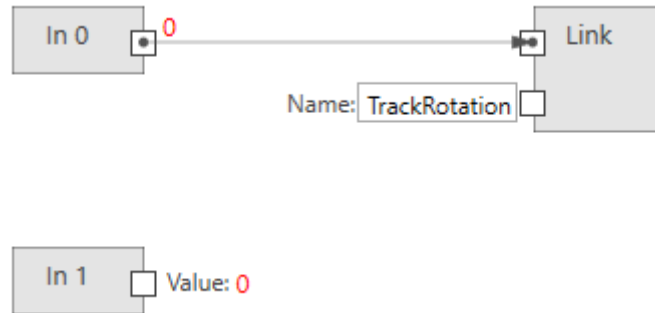
Fig. 19: Name **Link**

Once you have named links, you can use them as boolean values in any tool (besides the Acquisition tool itself). An example could be the **Track rotation** setting in the **Locate Area** tool.
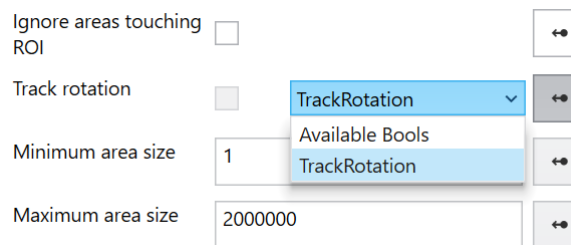
Fig. 20: Example of linking.

For technical reasons, digital inputs can not be used for linking values into the **Acquisition** tool itself.

4.3.3 Acquisition - Fieldbus Inputs

Many tool parameters can be changed by selecting or entering values. Most parameters can also be **Linked** - e.g connected with fieldbus inputs.

Linking a value is done in multiple steps on the tabs **PROFINET Inputs** or **EtherNet/IP Inputs** of the tool **Acquisition**:

1. A PROFINET or EtherNet/IP module in a slot is connected to a Read node.

2. The output of the read node is connected to a Link node.

3. The Link node defines a name that can be used as a paramter using the ⟷ button.

Input data is captured whenever an image is captured. This happens at the beginning of an inspection cycle.

Linking Fieldbus Inputs

Note: the screenshots show PROFINET, but the procedures for EtherNet/IP are exactly the same, unless where explicitly noted.

PROFINET Inputs
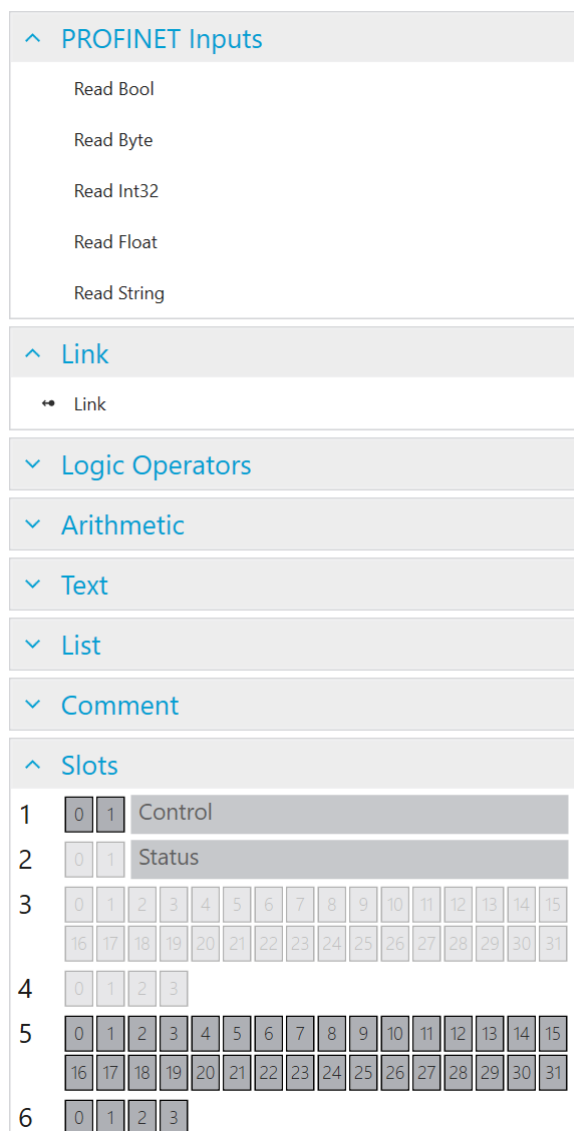
Fig. 21: The context menu on the left.

The **PLC** node is used to define modules in specific slots. This setup must exactly match the PROFINET or EtherNet/IP setup on the PLC, to enable communication between the PLC and the **CS-60/ID-600**. The PLC node is shown both on the Acquisition tool and the Logic tool. Inside the PLC node, the names Input and Output for the modules are used as seen from the PLC (A PLC Output is transferred from the PLC via PROFINET or EtherNet/IP to the **CS-60/ID-600**, where it is seen as an input. A PLC Input is transferred from the **CS-60/ID-600** via PROFINET or EtherNet/IP to the PLC, on the **CS-60/ID-600** this is seen as an output.)
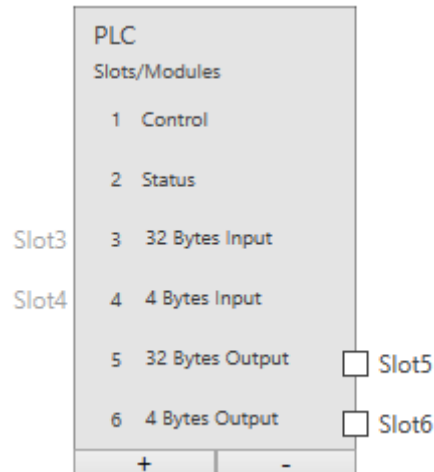
Fig. 22: The PLC node with six modules.

As an example, six modules are selected here (the control and status modules cannot be deselected). On the **PROFINET Inputs** page, only modules 5 and 6 can be connected (from the point of view of the PLC these are outputs, from the point of view of the **CS-60/ID-600** these are inputs.)

The modules are displayed graphically under the heading **Slots**. The used regions within the modules are also displayed.

Fig. 23: A string with 20 characters in module 5 at address 0.

PROFINET or EtherNet/IP can provide input values of different types: boolean (Bool), numeric (Byte, Int32, Float) and text (String). In order to used them for linking they must be defined and named. An input is defined by adding a read node of the desired type (by selecting them in the context menu on the left or with a right click on the canvas and selecting **Read Bool**, **Read Byte**, **Read Int32**, **Read Float** or **Read String**) and connecting it to a specific module in a slot defined in the PLC node. An input is named by adding a **Link** node (with a right click on the canvas and selecting **Link**), connecting a read node with the **Link** node and typing the respective name in the **Link** node.

Fig. 24: A boolean input.

The size of a boolean input is 1 byte (but you can allocate up to 8 in one byte).

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Bit indicates the bit to be addressed, where 0 is the rightmost bit and 7 is the leftmost bit in the byte.

Fallback specifies the boolean value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 25: A byte input.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Fallback specifies the byte value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 26: An integer input.

The size of an integer input is 4 bytes.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit integer. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback specifies the integer value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 27: A float input.

The size of a floating point input is 4 bytes.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback specifies the float value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Fig. 28: A string input.

The size of a string input is (2 + n) * s, where n is the maximum number of characters and s = 1 (for an Ansi STRING) or s = 2 (for a Unicode WSTRING). The first byte/word (Ansi/Unicode) contains the maximum number of characters the string can hold. The second byte/word (Ansi/Unicode) contains the actual number of characters the string holds.

Slot connects the node with an output module (SPS -> **CS-60/ID-600**).

Type specifies if the string is Ansi (STRING) or Unicode (WSTRING). This input is omitted with EtherNet/IP, as EtherNet/IP only supports Ansi texts.

Address specifies the address in the module.

Length specifies the number of characters in the string. A length of 1 specifies a single character.

Endianness specifies the byte order of the two bytes that comprise a 16 bit Unicode character. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first. With EtherNet/IP, this input is omitted as it is only relevant for Unicode texts.

Fallback specifies the string value that is used, when the fieldbus data is not available. fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

C4G Mode

For the C4G mode, the predefined modules **Control** and **Status** must be removed and replaced manually by a C4G compatible setup (see chapter C4G Mode Status and Control Signal Setup). Any further modules can be attached to these modules.

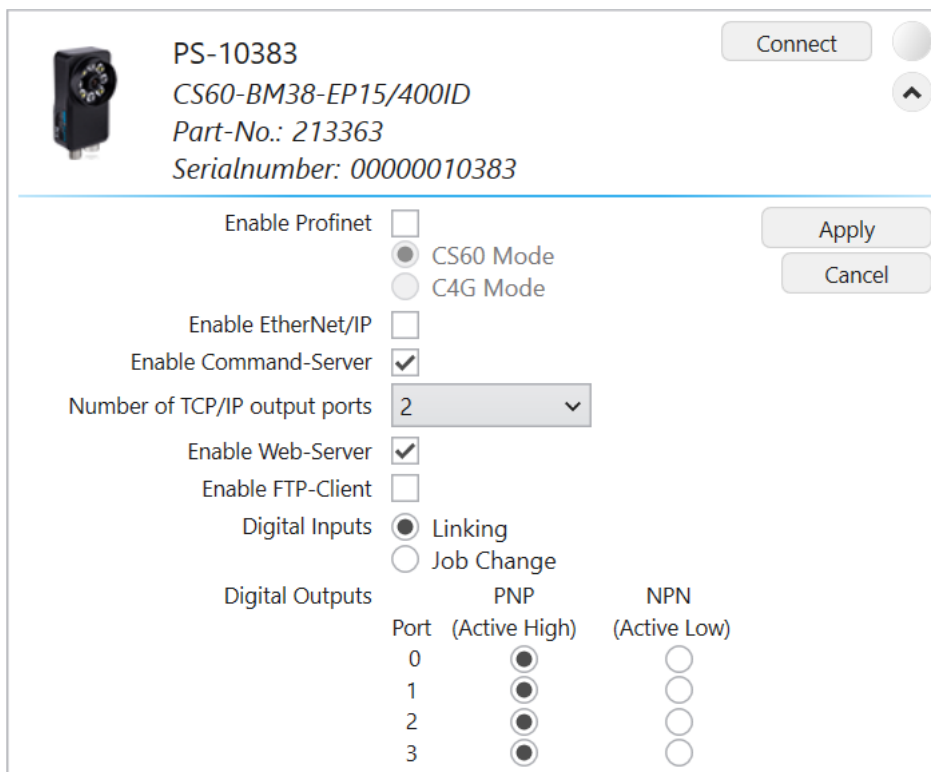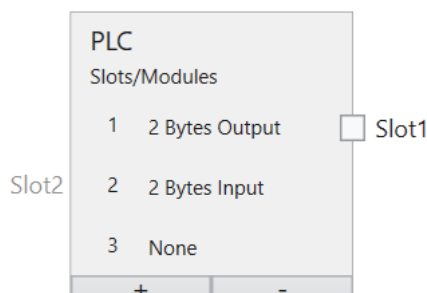Fig. 29: C4G Mode

4.3.4 Acquisition - TCP/IP Command Inputs

Many tool parameters can be changed by selecting or entering values. Most parameters can also be **Linked** - e.g connected with PROFINET inputs.

Linking a value is done in multiple steps the tab **TCP/IP Variables Inputs** of the tool **Acquisition**:

1. A TCP/IP variable is named and connected to a Read node.

2. The output of the read node is connected to a Link node.

3. The Link node defines a name that can be used as a paramter using the ⬛ button.

The names under points 1 and 3 can be different, but ultimately refer to the same object. Under point 1 the name is defined with which the value of the TCP/IP command `set name = value` is identified. Under point 3 the name is defined under which the value is identified with the links in the job.

Input data is captured whenever an image is captured. This happens at the beginning of an inspection cycle.

Linking TCP/IP Command Variable Inputs

TCP/IP Variables Inputs

Fig. 30: The context menu on the left.

The **Inputs** node is used to name variables in the TCP/IP namespace.

Fig. 31: The inputs node with one variable.

As an example one variable is defined here. The name must begin with a letter and can optionally be followed by one or more letters, numbers or _ (underscore). This name is valid in the TCP/IP namespace, i.e. when used with the TCP/IP command `set name = value`. For example, the command `set exp=100` would assign the value `100` to the variable `exp`.

TCP/IP command variables can provide input values of different types: boolean (Bool), numeric (Byte, Int32, Float) and text (String). In order to use them for linking they must be defined and named. An input is defined by adding a read node of the desired type (by selecting them in the context menu on the left or with a right click on the canvas and selecting **Read Bool**, **Read Byte**, **Read Int32**, **Read Float** or **Read String**) and connecting it to a specific variable defined in the Inputs node. An input is named by adding a **Link** node (with a right click on the canvas and selecting **Link**), connecting a read node with the **Link** node and typing the respective name in the **Link** node.

** Boolean Input**

Fig. 32: A boolean input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the boolean value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

Byte Input

Fig. 33: A byte input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the byte value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.
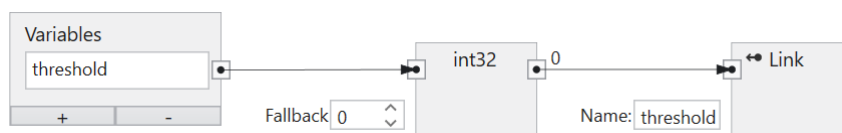
Integer Input

Fig. 34: An integer input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the integer value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.
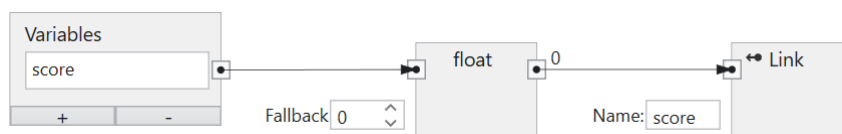
Float Input

Fig. 35: A float input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the float value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

String Input

Fig. 36: A string input.

Variable connects the node with a variable in the Inputs node.

Fallback specifies the string value that is used, when the TCP/IP command data is not available.

Name provides the symbolic name that is used, when this value will be linked to any tool input.

4.3.5 Detect Area Pixels

Fig. 37: The **Detect Area Pixels** tool icon.

The **Detect Area Pixels** tool is used to count and verify the number of pixels of a part in a scene using blob analysis. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Detect Area Pixels** tool is the number of pixels.
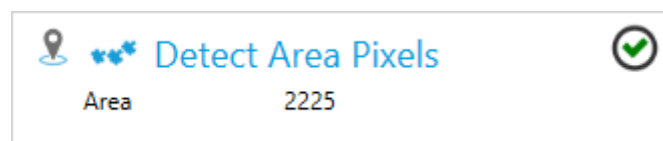
Fig. 38: The **Detect Area Pixels** tool tile.

The **Detect Area Pixels** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **shape of the ROI** can be selected as **rectangle**, **circle** or **ring**. Depending on the selection, the shape of the ROI set interactively on the right side will be adjusted.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or, in the case of the shape selection **rectangle**, also rotated (by dragging the rotation point in the middle of the top border).

The **Polarity** can be used to select **Dark areas** or **Light areas**. Dark or light is determined via a **Threshold** that can be calculated **Automatic** or can be set in a manual way with a slider using the logarmithmically scaled histogram display of the image.

Ignore areas touching the ROI ignore any objects that touch the region of interest that is specified interactively at the right.

Minimum area size and **Maximum area size** can be used to specify a range in square pixels for parts to be used for counting.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

Count

The primary tool result is **True/Ok** if the count is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, all areas are determined and their accumulated size in pixels is determined and their **Count** is displayed.
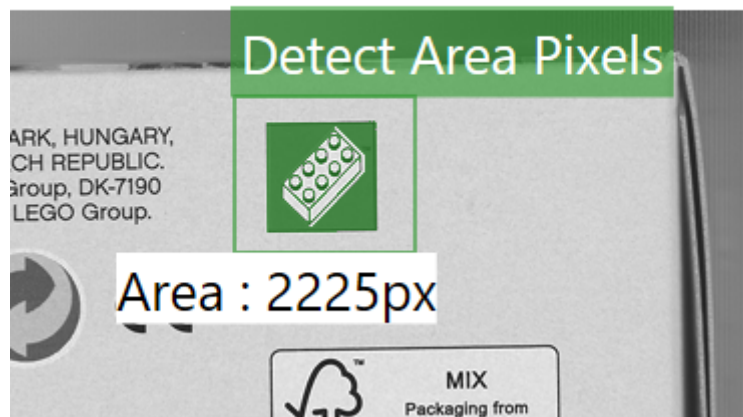
Fig. 39: The **Detect Area Pixels** tool gui.

4.3.6 Count Areas

Fig. 40: The **Count Areas** tool icon.

The **Count Areas** tool is used to count and verify the number of parts in a scene using blob analysis. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Count Areas** tool is a **Count** of objects.
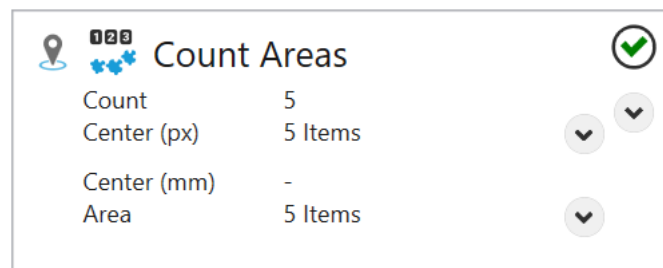
Fig. 41: The **Count Areas** tool tile.

The **Count Areas** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **shape of the ROI** can be selected as **rectangle**, **circle** or **ring**. Depending on the selection, the shape of the ROI set interactively on the right side will be adjusted.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or, in the case of the shape selection **rectangle**, also rotated (by dragging the rotation point in the middle of the top border).

The **Polarity** can be used to select **Light areas** or **Dark areas**. What is dark or light is determined via a **Threshold** that can be calculated **Automatic** or can be set in a manual way with a slider using the logarithmically scaled histogram display of the image.

Ignore areas touching the ROI ignore any objects that touch the region of interest that is specified interactively at the right.

Minimum area and **Maximum area** size can be used to specify a range in square pixels for areas to be used for counting.

Sorting specifies the order of the results with respect to their center.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Count

The primary tool result is **True/Ok** if the **Count** is between **Minimum** and **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the number of distinct areas is counted and displayed with colored graphical overlays. In addition, the **Count** is displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border lines or corners) or rotated (by dragging the pivot handle in the middle of the top edge).

Fig. 42: The **Count Areas** tool image-GUI.

4.3.7 Count Characters

Fig. 43: The icon of the **Count Characters** tool

The **Count Characters** tool is used to count and check the number of test characters in a scene. This tool is available for **CS-60** and **ID-600**.

In addition to the primary **Ok/Not Ok** result, a secondary result of the **Count Characters** tool is the **Number** of characters.
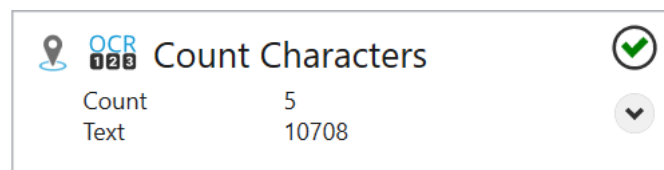
Fig. 44: The tile of the **Count Characters** tool

The **Count Characters** tool can use the pose of any localisation tool in the pipeline. To do this, select a specific localisation tool using the pin button.

Font

The font can be selected from a list.

There are several groups of fonts. The members of the individual groups differ as they contain different. The content of a font is described by its name. For the names of the pre-trained fonts, the following nomenclature applies:

The name begins with the group name, e.g. **Document** or **DotPrint**, followed by indicators for the symbol set which are contained in the OCR font. The meaning of the indicators is as follows:

- **0-9**: The OCR font contains the digits 0 to 9.

- **A-Z**: The OCR font contains the capital letters A to Z.

- **+** : The OCR font contains special characters. The list of special characters varies slightly between the individual OCR fonts. It is listed individually for each OCR font below.

If the name of the OCR font does not contain any of the above characters, the OCR font normally contains the digits 0 to 9, the upper case letters A to Z, the lower case letters lower case letters a to z and special characters. Some OCR fonts do not contain lowercase letters (e.g., DotPrint). This is explicitly mentioned in the description of the respective font explicitly mentioned.

To set the character set by linking, it must be transferred as text. The text must correspond exactly to the character set, as it can be found under **Font**.

Document

The font `Document` can be used to read characters printed in fonts such as Arial, Courier or Times New Roman. These are typical fonts for printing documents or letters.

Note that the characters I and l of the Arial font cannot be distinguished. This means that an `l` can be mistaken for an `I` and vice versa.

Available special characters: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

The DotPrint OCR font can be used to read characters printed with dot printers. It does not contain lower case letters.

Available special characters: - / . * :

Fig. 45: Examples of dotted font.

HandWritten

The HandWritten_0-9 font can be used to read handwritten numbers.

It contains the digits 0-9.

Available special characters: none
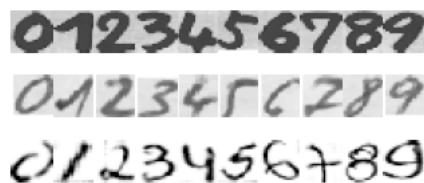
Fig. 46: Examples of handwritten digits.

Industrial

The Industrial font can be used to read characters written in fonts such as Arial, OCR-B or other sans serif fonts. other sans serif fonts. These fonts are normally used for printing z. labels, for example.

Available special characters: - / + . $ % * e £ ¥

Fig. 47: Examples of industrial prints.

OCR-A

The OCR-A font can be used to read characters that are printed in the OCR-A font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

Fig. 48: Selected characters of the OCR-A font.

OCR-B

The OCR-B font can be used to read characters that are printed in the OCR-B font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

Fig. 49: Selected characters of the OCR-B font.

Pharma

The Pharma font can be used to read characters printed in fonts such as Arial, OCR-B and and other fonts typically used in the pharmaceutical industry.

This OCR font does not contain lowercase letters.

Available special characters: - / . ( ) :

Fig. 50: Examples of pharmaceutical prints.

SEMI

The SEMI font can be used to read characters that are printed in the SEMI font, which consists of characters that are designed in such a way that they can be easily distinguished from each other. It has a limited set of characters.

This OCR font does not contain lowercase letters.

Available special characters: - .

Fig. 51: Character set of the SEMI font.

Universal

The Universal OCR font can be used to read a variety of different characters. The training of this CNN-based font is based on the character sets 'Document', 'DotPrint', 'SEMI' and 'Industrial'.

Available special characters: - / = + : < > . # $ % & ( ) @ * e £ ¥

Fig. 52: Example for the Universal character set.

Segmentation

Min. contrast determines the minimum contrast between letters and their background. The value can be set between 1 and 255 the default value is 15.

The **Polarity** can be set to **Dark on Light** for dark characters on a light background, **Light on Dark** for light characters on a dark background or **Both** if both types of text are to be segmented.

If **Eliminate border** is selected, regions that touch the border of the ROI will be discarded.

If **Add fragments** is selected, fragments such as the dot on an "i" are added to the segmented letters. However, this can also lead to errors being added to the segmented letters.

Touching characters determines the separation of pairs or small groups of adjacent characters that are segmented as a single region. segmented as a single region. If **Standard** is selected, such regions are recognised and split into two or more individual characters. If the value is set to **false**, no separation of touching characters is performed.

Character size

Minimum character height determines the minimum height of a character in pixels. In order to segment letters of any height, the value value 0 can be passed. **Minimum character height** restricts the height of letters, but not the height of punctuation or separators. separators.

Maximum character height determines the maximum height of a letter in pixels. In order to segment letters of any height, the value value 0 can be passed. **Maximum character height** restricts the height of letters, but not the height of punctuation or separators. separators.

Minimum character width determines the minimum width of a letter in pixels. To segment letters of any width, the value 0 can be passed. **Minimum character width** restricts the width of letters, but not the width of punctuation or separator characters. punctuation marks or separators.

Maximum character width determines the maximum width of a letter in pixels. To segment letters of any width, the value 0 can be passed. **Maximum character width** restricts the width of letters, but not the width of punctuation or separators. punctuation or separator characters.

Minimum stroke width determines the minimum stroke width of a letter in pixels. If the minimum stroke width is to be estimated automatically during text segmentation, the value 0 can be transferred. **Minimum stroke width** restricts the stroke width of letters, but not the stroke width of punctuation marks or separators.

Maximum stroke width determines the maximum stroke width of a letter in pixels. If the maximum stroke width is to be estimated automatically during text segmentation, the value 0 can be transferred. **Maximum stroke width** restricts the stroke width of letters, but not the stroke width of punctuation marks or separators.

Special characters

If **Punctuation** is selected, small punctuation marks that are close to the baseline of the text line (such as full stops or commas) are added to the segmented characters. comma) are added to the segmented letters.

If **Separator** is selected, separators such as a minus or the equal sign are added to the segmented letters. are added to the segmented letters.

Dot print

Dot print must be selected if the text to be segmented contains dot print letters.

Tight characters must be selected if the gaps between neighbouring letters are smaller than the gap between two dots within a letter. dots within a letter. If **Dot print** is not selected, this parameter has no effect. If the exact size of the smallest gap between neighbouring letters is known, this can be set using **Minimum character gap** instead. instead. In this case, the setting of **Tight characters** is ignored.

Minimum character gap is the size of the smallest gap between two letters in pixels. This parameter can be used in cases cases where the gaps between letters are smaller than those between dots within the letters, this parameter can lead to a more accurate segmentation. segmentation. If the minimum size is not known or is larger than the maximum gap between the dots within a letter, the value 0 can be passed. If **Dot print** is not selected, this parameter has no effect. If letters are printed very close to each other but the size of the minimum gap is unknown, **Tight characters** can be set instead.

Fig. 53: Here the smallest gap between two letters is 8 pixels

Maximum dot gap is the size of the largest gap between two dots within a letter in pixels. To segment any dot print letters, the value 0 can be passed. If **Dot print** is not selected, this parameter has no effect. parameter has no effect. If the largest gap between two dots is larger than the smallest gap between two letters, **Tight characters** should also be selected or **Minimum character gap** should be set accordingly. If **Maximum dot gap** is set, the performance of text recognition can be significantly increased.

Fig. 54: Here the largest gap between two dots is 3 pixels

Structure

Line structure defines the structure of the text being searched for. The distance between consecutive letters is calculated for each line. Based on these distances, the line is divided into individual letter blocks. Very small characters such as ., _ and - are ignored and interpreted as spaces. Additional separators can also be specified, which should also be ignored, see **Line separators**. The resulting letter blocks are then checked to see whether they correspond to one of the of the transferred structures.

A structure consists of a string that contains the number of letters/digits per letter block, which are separated by spaces. For example, if a date with two digits each for the day, month and year is to be found, the structure would be 2 2 2. If the year can also consist of four digits, the structure would be 2 2 2-4, so that the last block can consist of two to four letters/digits can consist of two to four letters/digits. If the structure corresponds to an empty text, the text to be found can consist of any structure.

Each structure found is saved as a single line. If the complete line containing a structure is to be returned instead is to be returned instead, **Whole line** can be selected.

Line separator is a string of separators that should be ignored when searching for structures, see **Line structure**. Please note that such a separator must always be included in the classifier. If, for example, the characters : and \ are to be treated as separators, the string :\\ can be transferred. Please note that the preceding \ prevents the use of the symbol \ as a literal.

Whole line if only the structures found are to be returned as lines, 'false' if every line containing a matching structure is to be returned in full.

Parameters

Sorting influences the order of the results in relation to their centre point.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect Objects (Close)** performs a morphological closing, **Reduce Objects (Erode)** performs an erosion, **Enlarge Objects (Erode)** performs an erosion. Erosion, **Enlarge objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations are to the brightness of the objects: **Divide objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Divide objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. are complementary. The **Gradient filter (Sobel)** enhances the edges and the **Noise filter (Lowpass)** reduces the noise.

If calibration data has been saved for the sensor, this data can be used for a conversion of image coordinates (px) to world coordinates (mm). With **Apply calibration** the conversion from image coordinates (px) to world coordinates (mm) is switched on.

These control elements are only displayed if calibration data has been saved.

Quantity

The primary tool result is **True/Ok** if the **Count** is between **Minimum** and **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

The image is displayed on the right-hand side, overlaid with a rectangular ROI. Within the ROI, the number of characters is counted and displayed with coloured graphic overlays. The **number** is also displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border lines or corners) or rotated (by dragging the pivot point in the centre of the upper edge).

4.3.8 Detect Edge Pixels

Fig. 55: The **Detect Edge Pixels** tool icon.

The **Detect Edge Pixels** tool is used to count and verify the number of edge pixels of a part in a scene using blob analysis. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Detect Edge Pixels** tool is the number of pixels.

Fig. 56: The **Detect Edge Pixels** tool tile.

The **Detect Edge Pixels** toolcan use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **shape of the ROI** can be selected as **rectangle**, **circle** or **ring**. Depending on the selection, the shape of the ROI set interactively on the right side will be adjusted.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or, in the case of the shape selection **rectangle**, also rotated (by dragging the rotation point in the middle of the top border).

Edge pixels are determined and selected via a **Threshold** that can be calculated **Automatic** or can be set in a manual way with a slider using the logarithmically scaled histogram display of the edge map of the image.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

Count

The primary tool result is **True/Ok** if the count is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Edges are calculated within the ROI and the number of edge pixels above the threshold is determined and displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border lines or corners) or rotated (by dragging the pivot handle in the middle of the top edge).

Fig. 57: The **Detect Edge Pixels** tool gui.

4.3.9 Count Edges

Fig. 58: The **Count Edges** tool icon.

The **Count Edges** tool is used to count and verify the number of edges along a line. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Count Edges** tool is a **Count** of edges.

Fig. 59: The **Count Edges** tool tile.

The **Count Edges** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **Polarity** can be used to select **Dark -> Light**, **Light -> Dark** or **Both**.

Smoothing affects the amount of smoothing applied to the **Edge strength**, gradually between **Sharp** and **Blurred**.

The edges to count are determined with **Edge strength** that can be determined visually by inspecting the gradient along the line.

Count

The primary tool result is **True/Ok** if the count is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a linear LOI (line of interest) on top. Along the LOI, the edge strength is calculated and used to determine the edges that are counted. In addition the **Count** is displayed.

The LOI (line of interest) can be moved (by dragging the line or the endpoints or by using the keyboard arrow keys). If the mouse cursor is on the endpoints, a four-way cursor is shown, and the endpoint can be moved in all directions. If the mouse cursor is moved a bit outwards from the endpoints, a two-way cursor is shown, and the endpoint can be moved in the direction of the line only.

The LOI must be as perpendicular as possible to the edges to be measured, i.e. optimally under 90°. If the search line becomes close to 0° the edges are more smoothed by averaging and the results become less precise.
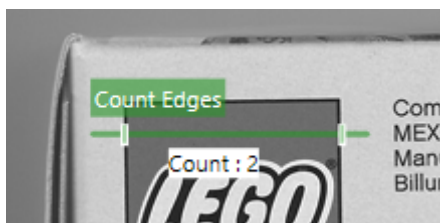
Fig. 60: The **Count Edges** tool image-GUI.

4.3.10 Count Shapes

Fig. 61: The **Count Shapes** tool icon.

The **Count Shapes** tool is used to count and verify the number of parts in a scene using pattern matching. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Count Shapes** tool is a **Count** of objects.

Fig. 62: The **Count Shapes** tool tile.

The **Count Shapes** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

Under **Method** there are three different methods for localizing image areas to choose from.

With the **Correlation** method, areas are in the foreground. The method can be used with a uniform background.

The **Geometric** method focuses on edges, and is more reliable on uneven backgrounds and on partial occlusions.

Both **Geometric** and **Correlation** should be constrained in rotation as memory requirements are significant.

The **Keypoint** method looks for prominent pixels in the template and compares them with those in the search image. As a result, the method recognizes different scaling to a certain extent and is not susceptible to partial occlusions.

The **Method** is only selectable in Training mode.

Allowed rotation allows you to constrain the rotation of the parts. This parameter is only visible in teach mode.

The **Template** used for counting can be created with a click on the **Teach pattern** button. Once you have teached a pattern, you are in search mode, where the ROI (region of interest) for the search region is selected. A click on the **New Pattern** button brings you back to teach mode.

Match quality can be used to select gradual settings between **Precise** and **Fast**. Faster matching is less precise and more precise matching is slower. This allows you to select the proper tradeoff for your application.

Accept threshold sets the minimum threshold for match scores in percent. Any matches below this treshold are ignored. A threshold of 50 % is in most cases too low for good matches.

Filter background excludes uniform background regions from processing, which can be used to speed up processing. If the background is cluttered wich many details, this option should not be used, since it can increase processing time in such cases.

Sorting specifies the order of the results with respect to their center.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Count

The primary tool result is **True/Ok** if the count is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. In teach mode, the ROI is used to select the pattern.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

Fig. 63: The **Count Shapes** tool teach gui.

In search mode, the ROI is used to select the search area. Inside the search area the pattern is searched, counted and displayed.
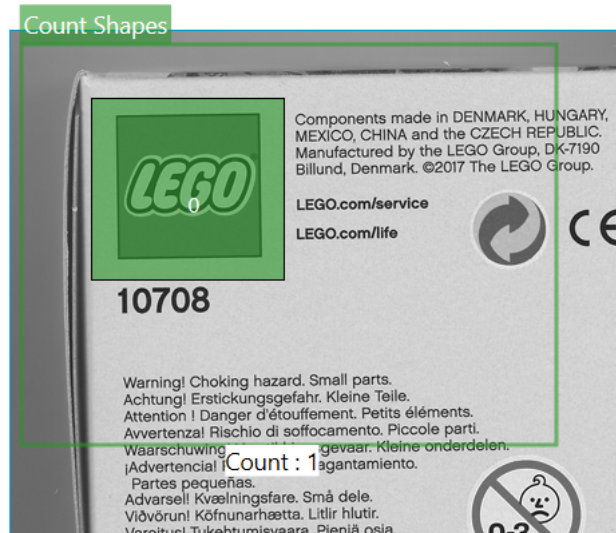
Fig. 64: The **Count Shapes** tool search gui.

4.3.11 Count Barcodes

Fig. 65: The **Count Barcode** tool icon.

The **Count Barcodes** tool is used to count barcodes or matrixcodes inside a region of interest. This tool is available for the **CS-60** and the **ID-600**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Count Barcode** tool is the **Count** of decoded codes.
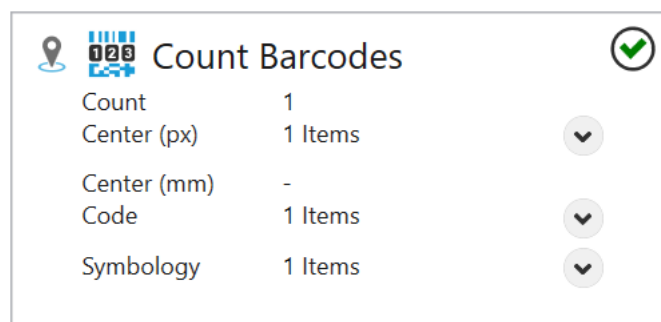
Fig. 66: The **Count Barcode** tool tile.

The **Count Barcode** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Code

The symbology can be detected **Auto**matically or **Manual**ly. Alternatively, the code can be set by **Linking** through the inputs.

In **Auto**matic mode, click **Identify Code** to detect the symbology automatically.

In **Manual** mode, filter by **1D** or **2D** codes and select the desired symbologies from the **Code** list.

To set the code by **Linking** it must be passed as text. The text must correspond exactly to the code as it can be found under **Available codes**. To recognize multiple codes at the same time, the text can also contain more than one of the available codes.

Parameters

The necessary image **Contrast** needed by the decoder may be affected by checking **Low** or **Enhance**, which will increase the performance at the cost of runtime.

The **Accuracy** can be set in four steps between **Precise** and **Fast**.

DPM marking can be used to influence the decoding of directly marked codes.

No DPM (Direct Part Marking): no special decoding of directly marked codes. **Laser/Etched**: reads laser or chemically etched directly marked codes. **Inkjet/Dotpeen**: reads directly marked codes printed with inkjet printers.

If these simple methods are not sufficient, there are advanced options with the ID pro license:

Pro: DotPeen Dark on Light: robust method for dark on light dotpeen directly marked codes. **Pro: DotPeen Light on Dark**: robust method for light on dark dotpeen directly marked codes. **Pro: Laser/Chemical Etched**: robust method for lasered and etched directly marked codes. **Pro: DotPeen and Etched**: robust method for dotpeen, lasered and etched directly marked codes (but maybe a bit slower than the specialized methods).

The verification method for 2D codes can be selected with **Verification**:

None: no verification is performed. **ISO/IEC 15415**: verification is carried out based on the ISO/IEC 15415 standard. **ISO/IEC TR29158**: a verification based on the ISO/IEC TR29158 standard is carried out.

The two verification modes are only available if a **Pro** method is selected for **DPM marking** at the same time.

Sorting specifies the order of the results with respect to their center.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are

also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Count

The primary tool result is **True/Ok** if the **Count** is between **Minimum** and **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

Count Barcodes

Code

Expected codes

| Auto ▼ |

| Identify code |

☑ 1D
  ☐ Code128
  ☐ Code39
  ☐ Codabar
  ☐ GS1
  ☑ UPC/EAN
  ☐ Code 2of5
  ☐ Pharmacode
☑ 2D
  ☐ Aztec
  ☐ DataMatrix
  ☐ Pdf417
  ☐ QR

Parameters

Contrast	☐ Low	
	☐ Enhance	
Min. barcode height	10	
Accuracy	Precise ——————◯—— Fast	
Image preprocessing	None ▼

Count

| Minimum | 1 | |
| Maximum | 8 | |

Fig. 67: The **Count Barcodes** tool parameters.

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the number of distinct barcodes is counted and displayed with colored graphical overlays. In addition, the **Count** is displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border

lines or corners) or rotated (by dragging the pivot handle in the middle of the top edge).

Fig. 68: The **Count Barcodes** tool image-GUI.

4.3.12 Detect Barcode

Fig. 69: The **Detect Barcode** tool icon.

The **Detect Barcode** tool is used to decode a barcode or matrixcode inside a region of interest. This tool is available for the **CS-60** and the **ID-600**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Detect Barcode** tool is the decoded **Code**.
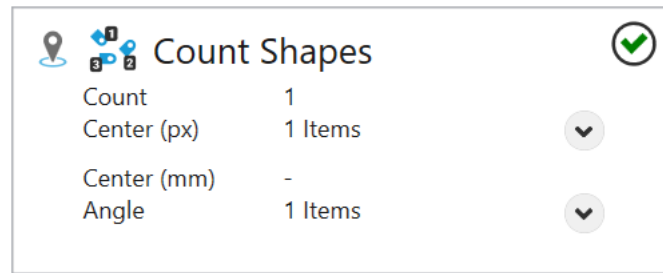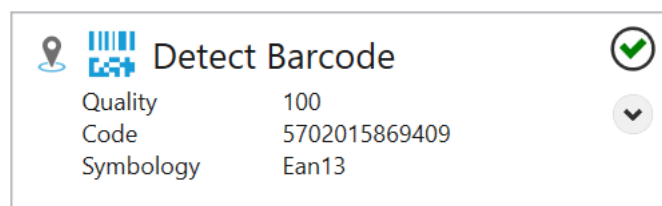
Fig. 70: The **Detect Barcode** tool tile.

The **Detect Barcode** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Code

The symbology can be detected **Auto**matically or **Manual**ly. Alternatively, the code can be set by **linking** through the inputs.

In **Auto**matic mode, click **Identify Code** to detect the symbology automatically.

In **Manual** mode, filter by **1D** or **2D** codes and select the desired symbology from the **Code** list.

To set the code by **Linking** it must be passed as text. The text must correspond exactly to the code as it can be found under **Available codes**.

Symbology Settings

Each of the symbologies may have specific settings.

Code 128

Codes with content shorter than **Min. length** will not be accepted.

Code 39

Codes with content shorter than **Min. length** will not be accepted.

Checksum checking can be **Disable**d, or **Enable**d, **Enable and strip** will check but strip the checksum characters from the result.

Full Ascii will enable full ascii decoding.

Stip start/stop will strip the start and stop characters from the result.

Codabar

Codes with content shorter than **Min. length** will not be accepted.

Checksum checking can be **Disable**d, or **Enable**d, **Enable and strip** will check but strip the checksum characters from the result.

Stip start/stop will strip the start and stop characters from the result.

Ratio will enable decoding with out of spec wide to narrow ratio over 5.5.

Pharmacode

Pharmacode is a code without error checking. The settings help to reduce reading errors. **Min. bars** and **Max. bars** define the range of the allowed number of bars. **Min. value** and **Max. value** set the range of numbers allowed. **Reverse decoding** sets the direction of decoding.

UPC/EAN

Expansion enables or disables UPC E expansion.

Supplement 2 enables or disables decoding of 2-digit supplemental code for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies.

Supplement 5 enables or disables decoding of 5-digit supplemental code for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies.

Add space to supplement enables the space character between the UPC/EAN code and the 2 or 5 digit supplemental code for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies. Applicable only when the decoding of the corresponding supplemental code is enabled.

Supplement required requires that the supplemental code can be decoded for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies.

Strip number enables or disables stripping the number system digit in decoded data for UPC-A and UPC-E symbologies.

Strip character enables or disables stripping the check digit for UPC-A, UPC-E, EAN13 and EAN8 symbologies.

Code 2of5

Codes with content shorter than **Min. length** will not be accepted.

Checksum checking can be **Disable**d, or **Enable**d, **Enable and strip** will check but strip the checksum characters from the result.

Reject partial enables or disables rejection of partial 2of5 decoding. Enabling this feature makes it less likely to partially decode a code when it is damaged or truncated because of its placement in the image (e.g. part of the barcode is missing).

Aztec

Polarity can be set to **Dark on Light**, **Light on Dark** or **Both**.

If **Mirror** is checked, codes seen from behind are also decoded.

Data Matrix

Polarity can be set to **Dark on Light**, **Light on Dark** or **Both**.

If **Mirror** is checked, codes seen from behind are also decoded.

Rectangular enables or disables rectangular codes.

QR Code

Send ECC Codewords specifies, whether error correction codewords will be available in results.

Parameters

The necessary image **Contrast** needed by the decoder may be affected by checking **Low** or **Enhance**, which will increase the performance at the cost of runtime.

The **Accuracy** can be set in four steps between **Precise** and **Fast**.

DPM marking can be used to influence the decoding of directly marked codes.

No DPM (Direct Part Marking): no special decoding of directly marked codes. **Laser/Etched**: reads laser or chemically etched directly marked codes. **Inkjet/Dotpeen**: reads directly marked codes printed with inkjet printers.

If these simple methods are not sufficient, there are advanced options with the ID pro license:

Pro: DotPeen Dark on Light: robust method for dark on light dotpeen directly marked codes. **Pro: DotPeen Light on Dark**: robust method for light on dark dotpeen directly marked codes. **Pro: Laser/Chemical Etched**: robust method for lasered and etched directly marked codes. **Pro: DotPeen and Etched**: robust method for dotpeen, lasered and etched directly marked codes (but maybe a bit slower than the specialized methods).

The verification method for 2D codes can be selected with **Verification**:

None: no verification is performed. **ISO/IEC 15415**: verification is carried out based on the ISO/IEC 15415 standard. **ISO/IEC TR29158**: a verification based on the ISO/IEC TR29158 standard is carried out.

The two verification modes are only available if a **Pro** method is selected for **DPM marking** at the same time.

NoReadText: this is the text that is output if no code could be read.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Expected Code

The decoded content is compared to the text in **Expected code**, according to the **Method**, which can be **None**, **Equals**, **Contains**, **Starts with** or **Ends with**.

The primary tool result is **True/Ok** if the content matches, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the barcode is decoded and displayed.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

Fig. 71: The **Detect Barcode** tool gui.

4.3.13 Detect Text
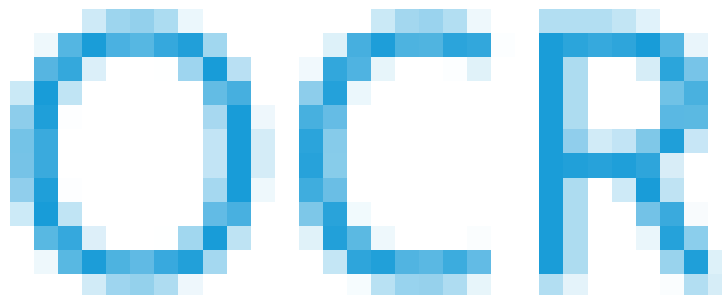
Fig. 72: The **Detect Text** tool icon.

The **Detect Text** tool is used to recognize a text inside a region of interest. This tool is available for the **CS-60** and the **ID-600**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Text Barcode** tool is the decoded **Text**.
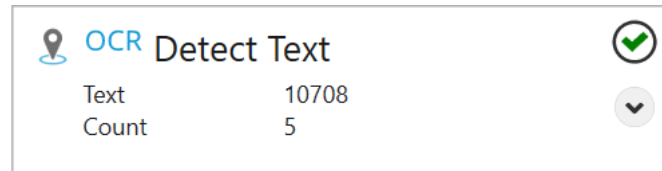
Fig. 73: The **Detect Text** tool tile.

The **Detect Text** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Font

The font can be selected from a list.

There are several groups of fonts. The members of the individual groups differ as they contain different. The content of a font is described by its name. For the names of the pre-trained fonts, the following nomenclature applies:

The name begins with the group name, e.g. **Document** or **DotPrint**, followed by indicators for the symbol set which are contained in the OCR font. The meaning of the indicators is as follows:

- **0-9**: The OCR font contains the digits 0 to 9.

- **A-Z**: The OCR font contains the capital letters A to Z.

- **+** : The OCR font contains special characters. The list of special characters varies slightly between the individual OCR fonts. It is listed individually for each OCR font below.

If the name of the OCR font does not contain any of the above characters, the OCR font normally contains the digits 0 to 9, the upper case letters A to Z, the lower case letters lower case letters a to z and special characters. Some OCR fonts do not contain lowercase letters (e.g., DotPrint). This is explicitly mentioned in the description of the respective font explicitly mentioned.

To set the character set by linking, it must be transferred as text. The text must correspond exactly to the character set, as it can be found under **Font**.

Document

The font `Document` can be used to read characters printed in fonts such as Arial, Courier or Times New Roman. These are typical fonts for printing documents or letters.

Note that the characters I and l of the Arial font cannot be distinguished. This means that an `l` can be mistaken for an `I` and vice versa.

Available special characters: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

The DotPrint OCR font can be used to read characters printed with dot printers. It does not contain lower case letters.

Available special characters: - / . * :

Fig. 74: Examples of dotted font.

HandWritten

The HandWritten_0-9 font can be used to read handwritten numbers.

It contains the digits 0-9.

Available special characters: none
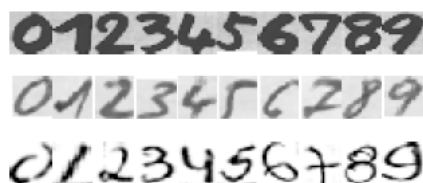
Fig. 75: Examples of handwritten digits.

Industrial

The Industrial font can be used to read characters written in fonts such as Arial, OCR-B or other sans serif fonts. other sans serif fonts. These fonts are normally used for printing z. labels, for example.

Available special characters: - / + . $ % * e £ ¥

Fig. 76: Examples of industrial prints.

OCR-A

The OCR-A font can be used to read characters that are printed in the OCR-A font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

Fig. 77: Selected characters of the OCR-A font.

OCR-B

The OCR-B font can be used to read characters that are printed in the OCR-B font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

Fig. 78: Selected characters of the OCR-B font.

Pharma

The Pharma font can be used to read characters printed in fonts such as Arial, OCR-B and and other fonts typically used in the pharmaceutical industry.

This OCR font does not contain lowercase letters.

Available special characters: - / . ( ) :

Fig. 79: Examples of pharmaceutical prints.

SEMI

The SEMI font can be used to read characters that are printed in the SEMI font, which consists of characters that are designed in such a way that they can be easily distinguished from each other. It has a limited set of characters.

This OCR font does not contain lowercase letters.

Available special characters: - .

Fig. 80: Character set of the SEMI font.

Universal

The Universal OCR font can be used to read a variety of different characters. The training of this CNN-based font is based on the character sets 'Document', 'DotPrint', 'SEMI' and 'Industrial'.

Available special characters: - / = + : < > . # $ % & ( ) @ * e £ ¥

Fig. 81: Example for the Universal character set.

Segmentation

Min. contrast determines the minimum contrast between letters and their background. The value can be set between 1 and 255 the default value is 15.

The **Polarity** can be set to **Dark on Light** for dark characters on a light background, **Light on Dark** for light characters on a dark background or **Both** if both types of text are to be segmented.

If **Eliminate border** is selected, regions that touch the border of the ROI will be discarded.

If **Add fragments** is selected, fragments such as the dot on an "i" are added to the segmented letters. However, this can also lead to errors being added to the segmented letters.

Touching characters determines the separation of pairs or small groups of adjacent characters that are segmented as a single region. segmented as a single region. If **Standard** is selected, such regions are recognised and split into two or more individual characters. If the value is set to **false**, no separation of touching characters is performed.

Character size

Minimum character height determines the minimum height of a character in pixels. In order to segment letters of any height, the value value 0 can be passed. **Minimum character height** restricts the height of letters, but not the height of punctuation or separators. separators.

Maximum character height determines the maximum height of a letter in pixels. In order to segment letters of any height, the value value 0 can be passed. **Maximum character height** restricts the height of letters, but not the height of punctuation or separators. separators.

Minimum character width determines the minimum width of a letter in pixels. To segment letters of any width, the value 0 can be passed. **Minimum character width** restricts the width of letters, but not the width of punctuation or separator characters. punctuation marks or separators.

Maximum character width determines the maximum width of a letter in pixels. To segment letters of any width, the value 0 can be passed. **Maximum character width** restricts the width of letters, but not the width of punctuation or separators. punctuation or separator characters.

Minimum stroke width determines the minimum stroke width of a letter in pixels. If the minimum stroke width is to be estimated automatically during text segmentation, the value 0 can be transferred. **Minimum stroke width** restricts the stroke width of letters, but not the stroke width of punctuation marks or separators.

Maximum stroke width determines the maximum stroke width of a letter in pixels. If the maximum stroke width is to be estimated automatically during text segmentation, the value 0 can be transferred. **Maximum stroke width** restricts the stroke width of letters, but not the stroke width of punctuation marks or separators.

Special characters

If **Punctuation** is selected, small punctuation marks that are close to the baseline of the text line (such as full stops or commas) are added to the segmented characters. comma) are added to the segmented letters.

If **Separator** is selected, separators such as a minus or the equal sign are added to the segmented letters. are added to the segmented letters.

Dot print

Dot print must be selected if the text to be segmented contains dot print letters.

Tight characters must be selected if the gaps between neighbouring letters are smaller than the gap between two dots within a letter. dots within a letter. If **Dot print** is not selected, this parameter has no effect. If the exact size of the smallest gap between neighbouring letters is known, this can be set using **Minimum character gap** instead. instead. In this case, the setting of **Tight characters** is ignored.

Minimum character gap is the size of the smallest gap between two letters in pixels. This parameter can be used in cases cases where the gaps between letters are smaller than those between dots within the letters, this parameter can lead to a more accurate segmentation. segmentation. If the minimum size is not known or is larger than the maximum gap between the dots within

a letter, the value 0 can be passed. If **Dot print** is not selected, this parameter has no effect. If letters are printed very close to each other but the size of the minimum gap is unknown, **Tight characters** can be set instead.

Fig. 82: Here the smallest gap between two letters is 8 pixels

Maximum dot gap is the size of the largest gap between two dots within a letter in pixels. To segment any dot print letters, the value 0 can be passed. If **Dot print** is not selected, this parameter has no effect. parameter has no effect. If the largest gap between two dots is larger than the smallest gap between two letters, **Tight characters** should also be selected or **Minimum character gap** should be set accordingly. If **Maximum dot gap** is set, the performance of text recognition can be significantly increased.

Fig. 83: Here the largest gap between two dots is 3 pixels

Structure

Line structure defines the structure of the text being searched for. The distance between consecutive letters is calculated for each line. Based on these distances, the line is divided into individual letter blocks. Very small characters such as ., _ and - are ignored and interpreted as spaces. Additional separators can also be specified, which should also be ignored, see **Line separators**. The resulting letter blocks are then checked to see whether they correspond to one of the of the transferred structures.

A structure consists of a string that contains the number of letters/digits per letter block, which are separated by spaces. For example, if a date with two digits each for the day, month and year is to be found, the structure would be 2 2 2. If the year can also consist of four digits, the

structure would be 2 2 2-4, so that the last block can consist of two to four letters/digits can consist of two to four letters/digits. If the structure corresponds to an empty text, the text to be found can consist of any structure.

Each structure found is saved as a single line. If the complete line containing a structure is to be returned instead is to be returned instead, **Whole line** can be selected.

Line separator is a string of separators that should be ignored when searching for structures, see **Line structure**. Please note that such a separator must always be included in the classifier. If, for example, the characters : and \ are to be treated as separators, the string :\\ can be transferred. Please note that the preceding \ prevents the use of the symbol \ as a literal.

Whole line if only the structures found are to be returned as lines, 'false' if every line containing a matching structure is to be returned in full.

Parameters

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect Objects (Close)** performs a morphological closing, **Reduce Objects (Erode)** performs an erosion, **Enlarge Objects (Erode)** performs an erosion. Erosion, **Enlarge objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations are to the brightness of the objects: **Divide objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Divide objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. are complementary. The **Gradient filter (Sobel)** enhances the edges and the **Noise filter (Lowpass)** reduces the noise.

If calibration data has been saved for the sensor, this data can be used for a conversion of image coordinates (px) to world coordinates (mm). With **Apply calibration** the conversion from image coordinates (px) to world coordinates (mm) is switched on.

These control elements are only displayed if calibration data has been saved.

Image display with overlay

The image is displayed on the right-hand side, overlaid with a rectangular ROI. Within the ROI, the number of characters is counted and displayed with coloured graphic overlays. The **number** is also displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border lines or corners) or rotated (by dragging the pivot point in the centre of the upper edge).

4.3.14 Locate Area

Fig. 84: The **Locate Area** tool icon.

The **Locate Area** tool is used to locate a part in a scene using blob analysis. This tool is only available for the **CS-60**.

When the **Locate Area** tool can locate a part, it determines the position and optionally rotation of the part and forwards this to downstream tools. The main result of the **Locate Area** tool is a **Pose** that combines position and rotation.

Fig. 85: The **Locate Area** tool tile.

The **Locate Area** tool shows the **Pose** as well as the **Area** of the part in pixels.

Tools in the pipeline can use this pose. You can have multiple location tools in one pipeline and in any tool you can choose the pose by selecting it from the context menu shown when you click the pin.

Parameters

The **shape of the ROI** can be selected as **rectangle**, **circle** or **ring**. Depending on the selection, the shape of the ROI set interactively on the right side will be adjusted.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or, in the case of the shape selection **rectangle**, also rotated (by dragging the rotation point in the middle of the top border).

The **Polarity** can be used to select **Dark areas** or **Light areas**. What is dark or light is determined via a **Threshold** that can be calculated **Automatic** or can be set in a manual way with a slider using the logarithmically scaled histogram display of the image.

Ignore areas touching the ROI ignore any objects that touch the region of interest that is specified interactively at the right. The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging rotation point in the middle of the top line).

Track rotation can be used to track the rotation of a part in addition to its position. Tracking the rotation is only possible with parts that have an inherent preferable direction (E.g. a circular part

has an infinite number of directions, which means that its direction is meaningless. A square has four possible directions, which also means that a preferable direction cannot be calculated.)

Minimum area size and **Maximum area size** can be used to specify a range in square pixels for parts to be used for location.

Connect areas combines separate areas when checked; when unchecked the biggest area is used.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Optional features

The tool can calculate additional parameters of the detected objects. Since this calculation takes time, the additional features can be selected as required.

The optional features are described in more detail in the chapter **Blob Analysis** in the manual.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the biggest area is determined and used for location. In addition, its size in pixels is displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border lines or corners) or rotated (by dragging the pivot handle in the middle of the top edge).
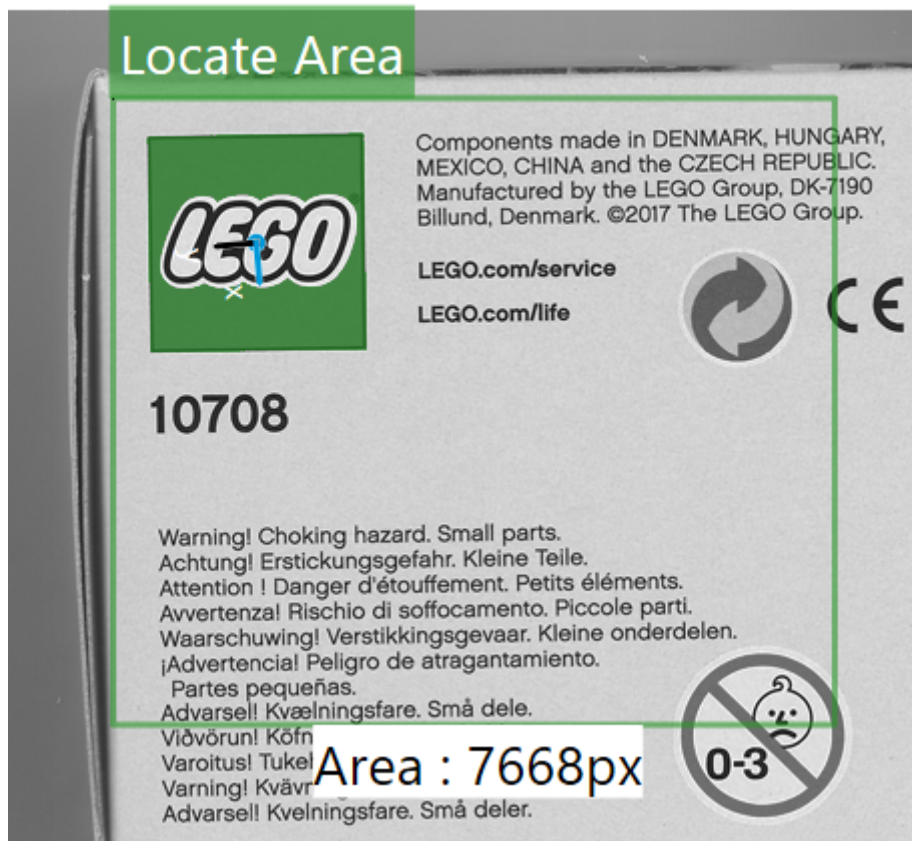
Fig. 86: The **Locate Area** tool gui.

4.3.15 Locate Barcode

Fig. 87: The **Locate Barcode** tool icon.

The **Locate Barcode** tool is used to locate a barcode in a scene. This tool is available for the **CS-60** and the **ID-600**.

When the **Locate Barcode** tool can locate a barcode, it determines the position and optionally rotation of the barcode and forwards this to downstream tools. The main result of the **Locate Barcode** tool is a **Pose** that combines position and rotation.
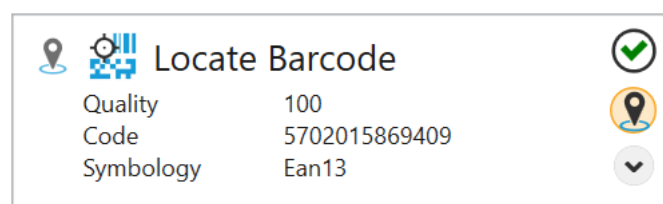
Fig. 88: The **Locate Barcode** tool tile.

The **Locate Barcode** tool shows the **Pose** of the barcode.

Tools in the pipeline can use this pose. You can have multiple location tools in one pipeline and in any tool you can choose the pose by selecting it from the context menu shown when you click the pin.

Code

The symbology can be detected **Auto**matically or **Manual**ly. Alternatively, the code can be set by **Linking** through the inputs.

In **Auto**matic mode, click **Identify Code** to detect the symbology automatically. **Identified code** will show the detected symbology.

In **Manual** mode, filter by **1D** or **2D** codes and select the desired symbology from the **Code** list.

To set the code by **Linking** it must be passed as text. The text must correspond exactly to the code as it can be found under **Available codes**.

Symbology Settings

Each of the symbologies may have specific settings.

Code 128

Codes with content shorter than **Min. length** will not be accepted.

Code 39

Codes with content shorter than **Min. length** will not be accepted.

Checksum checking can be **Disable**d, or **Enable**d, **Enable and strip** will check but strip the checksum characters from the result.

Full Ascii will enable full ascii decoding.

Stip start/stop will strip the start and stop characters from the result.

Codabar

Codes with content shorter than **Min. length** will not be accepted.

Checksum checking can be **Disable**d, or **Enable**d, **Enable and strip** will check but strip the checksum characters from the result.

Stip start/stop will strip the start and stop characters from the result.

Ratio will enable decoding with out of spec wide to narrow ratio over 5.5.

Pharmacode

Pharmacode is a code without error checking. The settings help to reduce reading errors. **Min. bars** and **Max. bars** define the range of the allowed number of bars. **Min. value** and **Max. value** set the range of numbers allowed. **Reverse decoding** sets the direction of decoding.

UPC/EAN

Expansion enables or disables UPC E expansion.

Supplement 2 enables or disables decoding of 2-digit supplemental code for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies.

Supplement 5 enables or disables decoding of 5-digit supplemental code for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies.

Add space to supplement enables the space character between the UPC/EAN code and the 2 or 5 digit supplemental code for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies. Applicable only when the decoding of the corresponding supplemental code is enabled.

Supplement required requires that the supplemental code can be decoded for UPC-A, UPC-E, EAN-13 and EAN-8 symbologies.

Strip number enables or disables stripping the number system digit in decoded data for UPC-A and UPC-E symbologies.

Strip character enables or disables stripping the check digit for UPC-A, UPC-E, EAN13 and EAN8 symbologies.

Code 2of5

Codes with content shorter than **Min. length** will not be accepted.

Checksum checking can be **Disable**d, or **Enable**d, **Enable and strip** will check but strip the checksum characters from the result.

Reject partial enables or disables rejection of partial 2of5 decoding. Enabling this feature makes it less likely to partially decode a code when it is damaged or truncated because of its placement in the image (e.g. part of the barcode is missing).

Aztec

Polarity can be set to **Dark on Light**, **Light on Dark** or **Both** to specify the codes that are accepted.

If **Mirror** is checked, codes seen from behind are also decoded.

Data Matrix

Polarity can be set to **Dark on Light**, **Light on Dark** or **Both** to specify the codes that are accepted.

If **Mirror** is checked, codes seen from behind are also decoded.

Rectangular enables or disables rectangular codes.

QR Code

Send ECC Codewords specifies, whether error correction codewords will be available in results.

Parameters

The necessary image **Contrast** needed by the decoder may be affected by checking **Low** or **Enhance**, which will increase the performance at the cost of runtime.

The **Accuracy** can be set in four steps between **Precise** and **Fast**.

DPM marking can be used to influence the decoding of directly marked codes.

No DPM (Direct Part Marking): no special decoding of directly marked codes. **Laser/Etched**: reads laser or chemically etched directly marked codes. **Inkjet/Dotpeen**: reads directly marked codes printed with inkjet printers.

If these simple methods are not sufficient, there are advanced options with the ID pro license:

Pro: DotPeen Dark on Light: robust method for dark on light dotpeen directly marked codes. **Pro: DotPeen Light on Dark**: robust method for light on dark dotpeen directly marked codes. **Pro: Laser/Chemical Etched**: robust method for lasered and etched directly marked codes. **Pro: DotPeen and Etched**: robust method for dotpeen, lasered and etched directly marked codes (but maybe a bit slower than the specialized methods).

The verification method for 2D codes can be selected with **Verification**:

None: no verification is performed. **ISO/IEC 15415**: verification is carried out based on the ISO/IEC 15415 standard. **ISO/IEC TR29158**: a verification based on the ISO/IEC TR29158 standard is carried out.

The two verification modes are only available if a **Pro** method is selected for **DPM marking** at the same time.

NoReadText: this is the text that is output if no code could be read.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects**

(Open) and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the barcode is decoded and used for location. In addition, the decoded content is displayed.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

Fig. 89: The locate barcode tool gui.

4.3.16 Locate Edge

Fig. 90: The **Locate Edge** tool icon.

The **Locate Edge** tool is used to locate an edge in a scene. This tool is only available for the **CS-60**.

When the **Locate Edge** tool can locate a part, it determines the position and optionally rotation of the edge and forwards this to downstream tools. The primary result of the **Locate Edge** tool is a **Pose** that combines position and rotation.
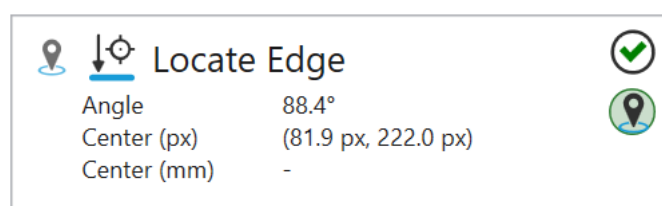
Fig. 91: The **Locate Edge** tool tile.

The **Locate Edge** tool shows the **Angle** as well as the **Pose** of the edge.

Tools in the pipeline can use this pose. You can have multiple location tools in one pipeline and in any tool you can choose the pose by selecting it from the context menu shown when you click the pin.

Calibration

Parameters

The **Scan direction** is used to select the direction of the scans. You can choose **Right**, **Down**, **Left** or **Up**.

The **Polarity** is used to select the brightness transition of the edge. You can choose **Dark -> Light**, **Light -> Dark** or **Both**. This specifies the type of edge in combination with the **Scan direction**.

The **Edge Selection** chooses the **First**, **Last** or **Best** edge with respect to the scan direction.

Smoothing selects the sharpness of the edge. The slider allows you to select gradual values between **Blurred** and **Sharp**.

The **Edge Strength** display a gradient curve of the edge and allows you to select the desired strength of edges.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs

a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top.

The ROI (region of interest) can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

The ROI must be as perpendicular as possible to the edges to be measured, i.e. optimally under 90°. If the search lines become close to 0° the edges are more smoothed by averaging and the results become less precise.

Inside the ROI, the scanlines are visualized with dotted lines and from the points determined along the scanlines, a line is fitted and displayed.In addition, the angle of the line is displayed.
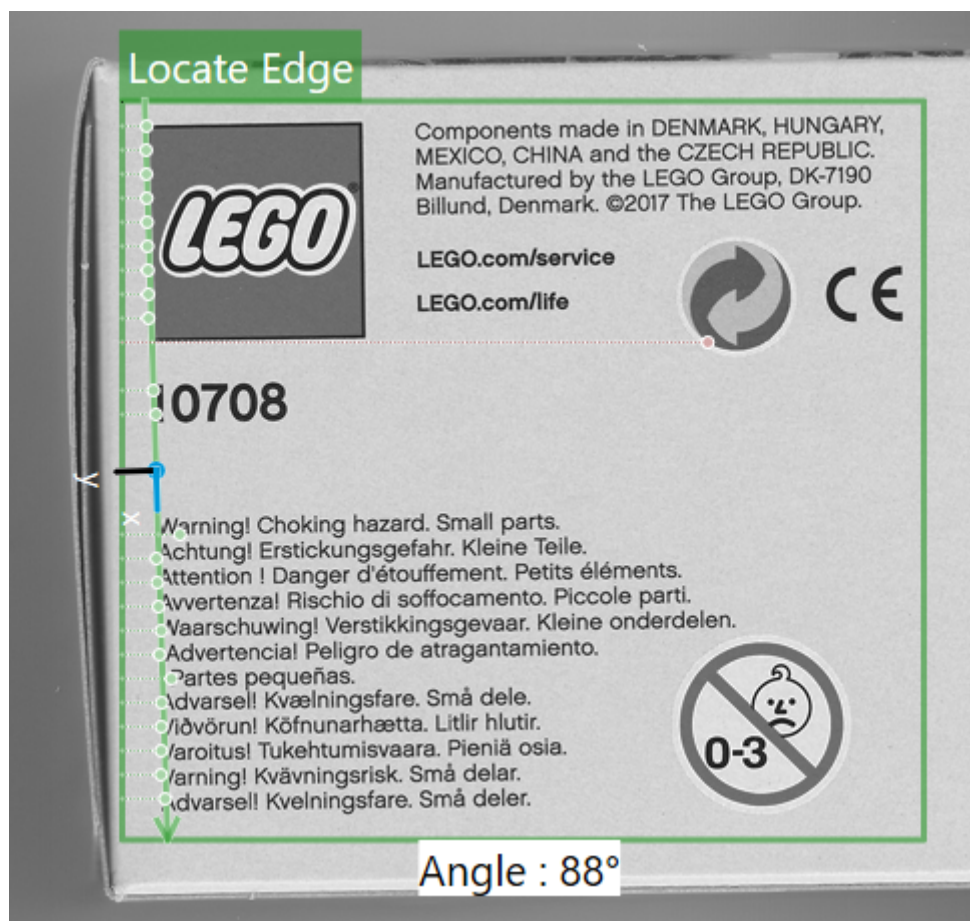
Fig. 92: The **Locate Edge** tool gui.

4.3.17 Locate Shape

Fig. 93: The **Locate Shape** tool icon.

The **Locate Shape** tool is used to locate a part in a scene using pattern matching. This tool is only available for the **CS-60**.

When the **Locate Shape** tool can locate a part, it determines the position and rotation of the part and forwards this to downstream tools. The main result of the **Locate Shape** tool is a **Pose** that combines position and rotation.

Fig. 94: The **Locate Shape** tool tile.

The **Locate Shape** tool shows the **Pose** as well as the **Score** of the match in percent used to find the pose.

Tools in the pipeline can use this pose. You can have multiple location tools in one pipeline and in any tool you can choose the pose by selecting it from the context menu shown when you click the pin.

Parameters

Under **Method** there are three different methods for localizing image areas to choose from.

With the **Correlation** method, areas are in the foreground. The method can be used with a uniform background.

The **Geometric** method focuses on edges, and is more reliable on uneven backgrounds and on partial occlusions.

Both **Geometric** and **Correlation** should be constrained in rotation as memory requirements are significant.

The **Keypoint** method looks for prominent pixels in the template and compares them with those in the search image. As a result, the method recognizes different scaling to a certain extent and is not susceptible to partial occlusions.

The **Method** is only selectable in Training mode.

The different methods area explained in more detail in the chapter **Pattern Matching** in the manual.

Allowed Rotation allows you to constrain the rotation of the parts. This parameter is only visible in teach mode.

The **Template** used for counting can be created with a click on the **Teach pattern** button. Once you have teached a pattern, you are in search mode, where the ROI (region of interest) for the search region is selected. A click on the **New Pattern** button brings you back to teach mode.

Match quality can be used to select gradual settings between **Precise** and **Fast**. Faster matching is less precise and more precise matching is slower. This allows you to select the proper tradeoff for your application.

Accept threshold sets the minimum threshold for match scores in percent. Any matches below this treshold are ignored. A threshold of 50 % is in most cases too low for good matches.

Filter background excludes uniform background regions from processing, which can be used to speed up processing. If the background is cluttered wich many details, this option should not be used, since it can increase processing time in such cases.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. In teach mode, the ROI is used to select the pattern.
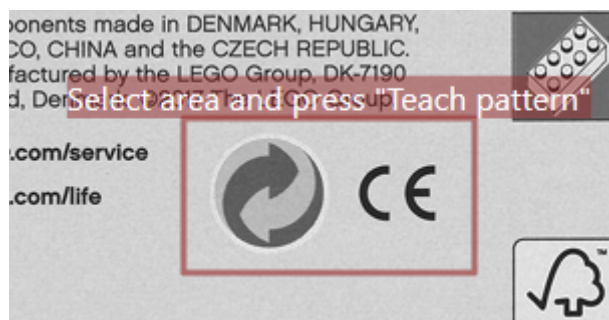
Fig. 95: The **Locate Shapes** tool teach gui.

In search mode, the ROI is used to select the search area. Inside the search area the pattern is searched, counted and displayed.
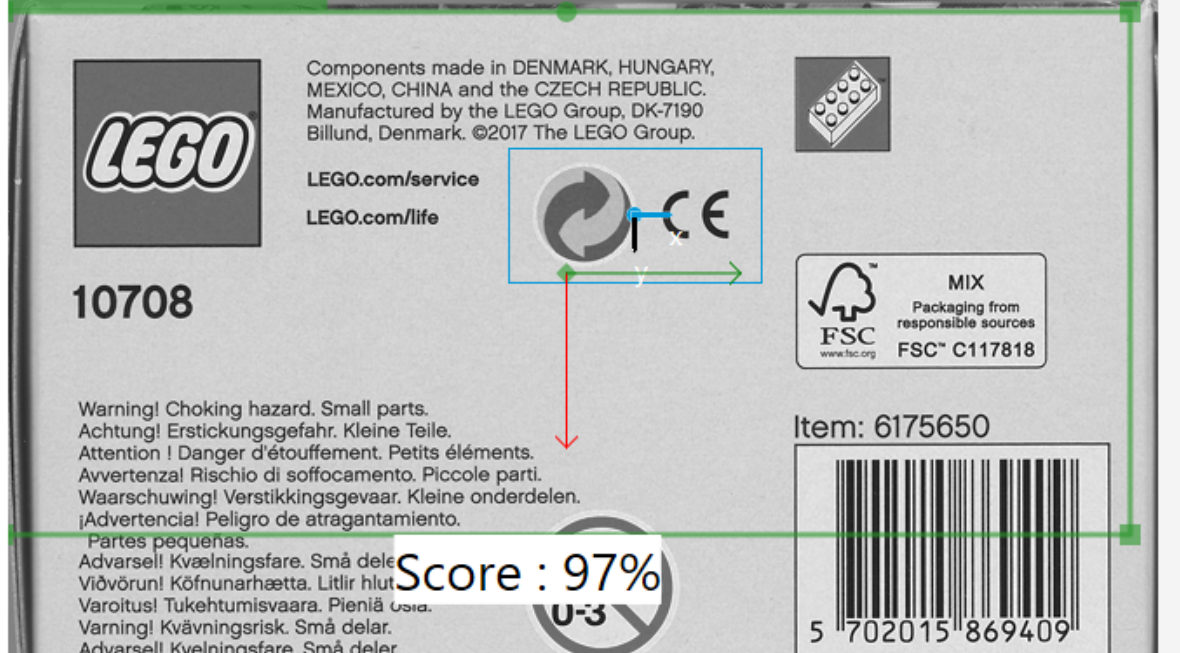
Fig. 96: The **Locate Shapes** tool search gui.

4.3.18 Locate Text

Fig. 97: The **Locate Text** tool icon.

The **Locate Text** tool is used to locate a text in a scene. This tool is available for the **CS-60** and the **ID-600**.

When the **Locate Text** tool can locate a barcode, it determines the position and optionally rotation of the barcode and forwards this to downstream tools. The main result of the **Locate Text** tool is a **Pose** that combines position and rotation.
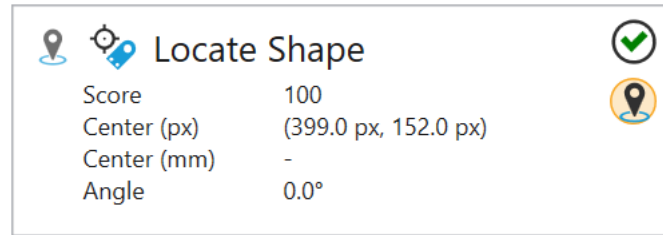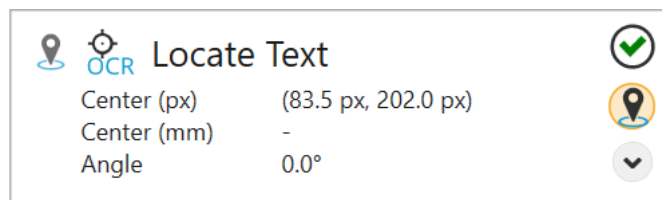

Fig. 98: The **Locate Text** tool tile.

The **Locate Text** tool shows the **Pose** of the barcode.

Tools in the pipeline can use this pose. You can have multiple location tools in one pipeline and in any tool you can choose the pose by selecting it from the context menu shown when you click the pin.

Font

The font can be selected from a list.

There are several groups of fonts. The members of the individual groups differ as they contain different. The content of a font is described by its name. For the names of the pre-trained fonts, the following nomenclature applies:

The name begins with the group name, e.g. **Document** or **DotPrint**, followed by indicators for the symbol set which are contained in the OCR font. The meaning of the indicators is as follows:

- **0-9**: The OCR font contains the digits 0 to 9.

- **A-Z**: The OCR font contains the capital letters A to Z.

- **+** : The OCR font contains special characters. The list of special characters varies slightly between the individual OCR fonts. It is listed individually for each OCR font below.

If the name of the OCR font does not contain any of the above characters, the OCR font normally contains the digits 0 to 9, the upper case letters A to Z, the lower case letters lower case letters a to z and special characters. Some OCR fonts do not contain lowercase letters (e.g., DotPrint). This is explicitly mentioned in the description of the respective font explicitly mentioned.

To set the character set by linking, it must be transferred as text. The text must correspond exactly to the character set, as it can be found under **Font**.

Document

The font `Document` can be used to read characters printed in fonts such as Arial, Courier or Times New Roman. These are typical fonts for printing documents or letters.

Note that the characters I and l of the Arial font cannot be distinguished. This means that an `l` can be mistaken for an `I` and vice versa.

Available special characters: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

The DotPrint OCR font can be used to read characters printed with dot printers. It does not contain lower case letters.

Available special characters: - / . * :

Fig. 99: Examples of dotted font.

HandWritten

The HandWritten_0-9 font can be used to read handwritten numbers.

It contains the digits 0-9.

Available special characters: none
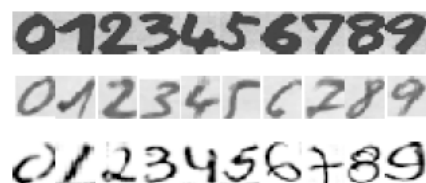
Fig. 100: Examples of handwritten digits.

Industrial

The Industrial font can be used to read characters written in fonts such as Arial, OCR-B or other sans serif fonts. other sans serif fonts. These fonts are normally used for printing z. labels, for example.

Available special characters: - / + . $ % * e £ ¥
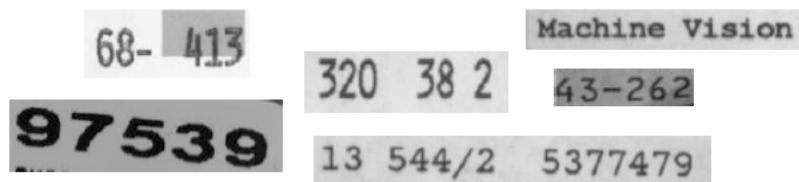
Fig. 101: Examples of industrial prints.

OCR-A

The OCR-A font can be used to read characters that are printed in the OCR-A font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

Fig. 102: Selected characters of the OCR-A font.

OCR-B

The OCR-B font can be used to read characters that are printed in the OCR-B font.

Available special characters: - ? ! / { } = + < > . # $ % & ( ) @ * e £ ¥

Fig. 103: Selected characters of the OCR-B font.

Pharma

The Pharma font can be used to read characters printed in fonts such as Arial, OCR-B and and other fonts typically used in the pharmaceutical industry.

This OCR font does not contain lowercase letters.

Available special characters: - / . ( ) :

Fig. 104: Examples of pharmaceutical prints.

SEMI

The SEMI font can be used to read characters that are printed in the SEMI font, which consists of characters that are designed in such a way that they can be easily distinguished from each other. It has a limited set of characters.

This OCR font does not contain lowercase letters.

Available special characters: - .

Fig. 105: Character set of the SEMI font.

Universal

The Universal OCR font can be used to read a variety of different characters. The training of this CNN-based font is based on the character sets 'Document', 'DotPrint', 'SEMI' and 'Industrial'.

Available special characters: - / = + : < > . # $ % & ( ) @ * e £ ¥

Fig. 106: Example for the Universal character set.

Segmentation

Min. contrast determines the minimum contrast between letters and their background. The value can be set between 1 and 255 the default value is 15.

The **Polarity** can be set to **Dark on Light** for dark characters on a light background, **Light on Dark** for light characters on a dark background or **Both** if both types of text are to be segmented.

If **Eliminate border** is selected, regions that touch the border of the ROI will be discarded.

If **Add fragments** is selected, fragments such as the dot on an "i" are added to the segmented letters. However, this can also lead to errors being added to the segmented letters.

Touching characters determines the separation of pairs or small groups of adjacent characters that are segmented as a single region. segmented as a single region. If **Standard** is selected, such regions are recognised and split into two or more individual characters. If the value is set to **false**, no separation of touching characters is performed.

Character size

Minimum character height determines the minimum height of a character in pixels. In order to segment letters of any height, the value value 0 can be passed. **Minimum character height** restricts the height of letters, but not the height of punctuation or separators. separators.

Maximum character height determines the maximum height of a letter in pixels. In order to segment letters of any height, the value value 0 can be passed. **Maximum character height** restricts the height of letters, but not the height of punctuation or separators. separators.

Minimum character width determines the minimum width of a letter in pixels. To segment letters of any width, the value 0 can be passed. **Minimum character width** restricts the width of letters, but not the width of punctuation or separator characters. punctuation marks or separators.

Maximum character width determines the maximum width of a letter in pixels. To segment letters of any width, the value 0 can be passed. **Maximum character width** restricts the width of letters, but not the width of punctuation or separators. punctuation or separator characters.

Minimum stroke width determines the minimum stroke width of a letter in pixels. If the minimum stroke width is to be estimated automatically during text segmentation, the value 0 can be transferred. **Minimum stroke width** restricts the stroke width of letters, but not the stroke width of punctuation marks or separators.

Maximum stroke width determines the maximum stroke width of a letter in pixels. If the maximum stroke width is to be estimated automatically during text segmentation, the value 0 can be transferred. **Maximum stroke width** restricts the stroke width of letters, but not the stroke width of punctuation marks or separators.

Special characters

If **Punctuation** is selected, small punctuation marks that are close to the baseline of the text line (such as full stops or commas) are added to the segmented characters. comma) are added to the segmented letters.

If **Separator** is selected, separators such as a minus or the equal sign are added to the segmented letters. are added to the segmented letters.

Dot print

Dot print must be selected if the text to be segmented contains dot print letters.

Tight characters must be selected if the gaps between neighbouring letters are smaller than the gap between two dots within a letter. dots within a letter. If **Dot print** is not selected, this parameter has no effect. If the exact size of the smallest gap between neighbouring letters is known, this can be set using **Minimum character gap** instead. instead. In this case, the setting of **Tight characters** is ignored.

Minimum character gap is the size of the smallest gap between two letters in pixels. This parameter can be used in cases cases where the gaps between letters are smaller than those between dots within the letters, this parameter can lead to a more accurate segmentation. segmentation. If the minimum size is not known or is larger than the maximum gap between the dots within

a letter, the value 0 can be passed. If **dot print** is not selected, this parameter has no effect. If letters are printed very close to each other but the size of the minimum gap is unknown, **Tight characters** can be set instead.

Fig. 107: Here the smallest gap between two letters is 8 pixels

Maximum dot gap is the size of the largest gap between two dots within a letter in pixels. To segment any dot print letters, the value 0 can be passed. If **Dot print** is not selected, this parameter has no effect. parameter has no effect. If the largest gap between two dots is larger than the smallest gap between two letters, **Tight characters** should also be selected or **Minimum character gap** should be set accordingly. If **Maximum dot gap** is set, the performance of text recognition can be significantly increased.
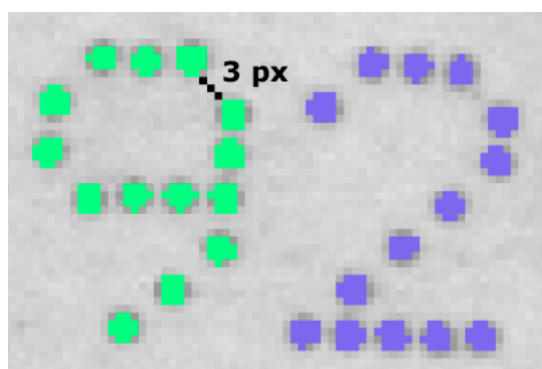
Fig. 108: Here the largest gap between two dots is 3 pixels

Structure

Line structure defines the structure of the text being searched for. The distance between consecutive letters is calculated for each line. Based on these distances, the line is divided into individual letter blocks. Very small characters such as ., _ and - are ignored and interpreted as spaces. Additional separators can also be specified, which should also be ignored, see **Line separators**. The resulting letter blocks are then checked to see whether they correspond to one of the of the transferred structures.

A structure consists of a string that contains the number of letters/digits per letter block, which are separated by spaces. For example, if a date with two digits each for the day, month and year is to be found, the structure would be 2 2 2. If the year can also consist of four digits, the

structure would be 2 2 2-4, so that the last block can consist of two to four letters/digits can consist of two to four letters/digits. If the structure corresponds to an empty text, the text to be found can consist of any structure.

Each structure found is saved as a single line. If the complete line containing a structure is to be returned instead is to be returned instead, **Whole line** can be selected.

Line separator is a string of separators that should be ignored when searching for structures, see **Line structure**. Please note that such a separator must always be included in the classifier. If, for example, the characters : and \ are to be treated as separators, the string :\\ can be transferred. Please note that the preceding \ prevents the use of the symbol \ as a literal.

Whole line if only the structures found are to be returned as lines, 'false' if every line containing a matching structure is to be returned in full.

Parameters

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect Objects (Close)** performs a morphological closing, **Reduce Objects (Erode)** performs an erosion, **Enlarge Objects (Erode)** performs an erosion. Erosion, **Enlarge objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations are to the brightness of the objects: **Divide objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Divide objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. are complementary. The **Gradient filter (Sobel)** enhances the edges and the **Noise filter (Lowpass)** reduces the noise.

If calibration data has been saved for the sensor, this data can be used for a conversion of image coordinates (px) to world coordinates (mm). With **Apply calibration** the conversion from image coordinates (px) to world coordinates (mm) is switched on.

These control elements are only displayed if calibration data has been saved.

Image display with overlay

The image is displayed on the right-hand side, overlaid with a rectangular ROI. Within the ROI, the number of characters is counted and displayed with coloured graphic overlays. The **number** is also displayed.

The region of interest can be moved (by dragging inside) or resized (by dragging the border lines or corners) or rotated (by dragging the pivot point in the centre of the upper edge).

4.3.19 Measure Angle

Fig. 109: The **Measure Angle** tool icon.

The **Measure Angle** tool is used to measure the angle of a line in the scene. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Measure Angle** tool is the **Angle**.

Fig. 110: The **Measure Angle** tool tile.

The **Measure Angle** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **Scan direction** is used to select the direction of the scans. You can choose **Left -> Right**, **Right -> Left**, **Top -> Bottom** or **Bottom -> Top**.

The **Polarity** is used to select the brightness transition of the edge. You can choose **Dark -> Light**, **Light -> Dark** or **Both**. This specifies the type of edge in combination with the **Scan direction**.

The **Edge selection** chooses the **Best**, **First** or **Last** edge with respect to the scan direction.

Smoothing selects the sharpness of the edge. The slider allows you to select gradual values between **Blurred** and **Sharp**.

The **Edge Strength** display a gradient curve of the edge and allow you to select the desired strength of edges.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Measurement

The primary tool result is **True/Ok** if the angle is within the **Tolerance (+/-)** in degrees, with respect to the **Expected angle**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the search direction is visualized with dotted lines, the found edge points are visualized with green dots, and the fitted line is displayed. The angular tolerance is displayed with a butterfly-shaped range.

The ROI (region of interest) can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

The ROI must be as perpendicular as possible to the edges to be measured, i.e. optimally under 90°. If the search lines become close to 0° the edges are more smoothed by averaging and the results become less precise.

Fig. 111: The **Measure Angle** tool gui.

In addition, a "butterfly" visualizes the angular tolerance.
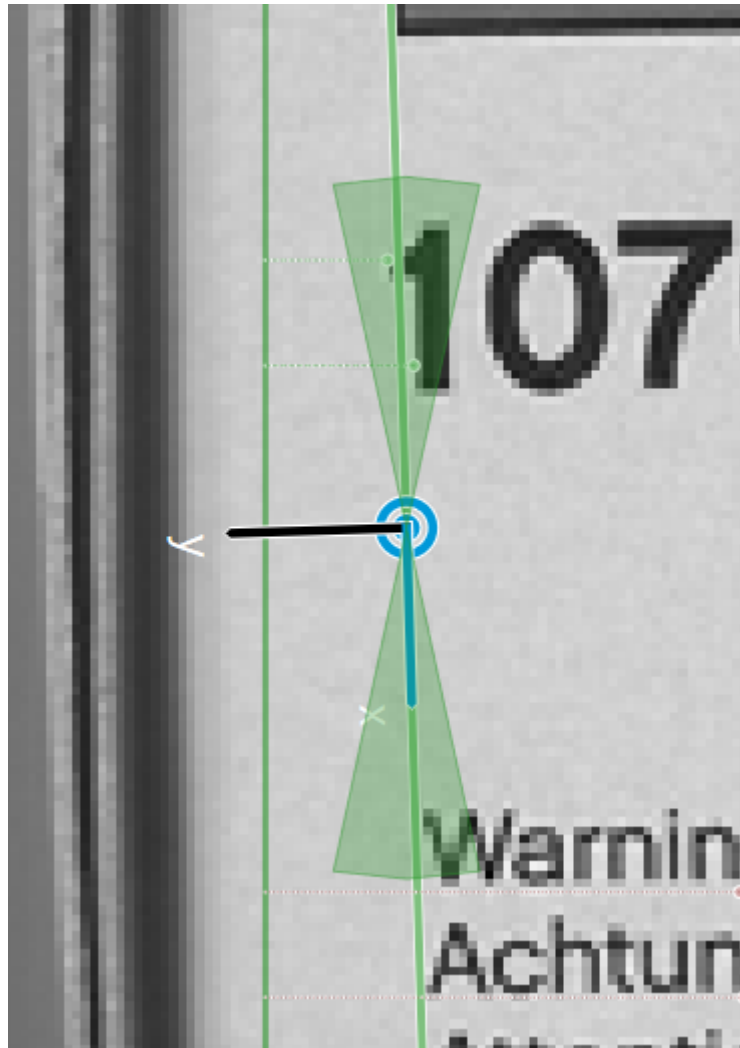
Fig. 112: The **Measure Angle** tool "butterfly".

4.3.20 Detect Brightness

Fig. 113: The **Detect Brightness** tool icon.

The **Detect Brightness** tool is used to measure the average brightness inside a region of interest. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Detect Brightness** tool is the **Average**, which is an average value normalized to 100 %.
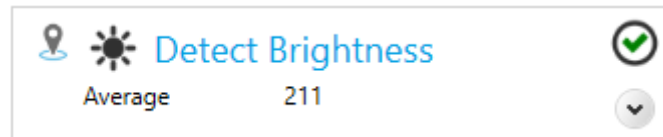
Fig. 114: The **Detect Brightness** tool tile.

The **Detect Brightness** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **shape of the ROI** can be selected as **rectangle**, **circle** or **ring**. Depending on the selection, the shape of the ROI set interactively on the right side will be adjusted.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or, in the case of the shape selection **rectangle**, also rotated (by dragging the rotation point in the middle of the top border).

The average brightness is measured inside this region of interest. A logarithmically scaled histogram of the greyvalues inside the ROI is displayed to give you an idea of the grey value distribution.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

The primary tool result is **True/Ok** if the brightness is between **Minimum** and **Maximum**, otherwise the result is **False/Not Ok**.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the average brightness is determined and displayed.

Fig. 115: The **Detect Brightness** tool gui.

4.3.21 Measure Circle

Fig. 116: The **Measure Circle** tool icon.

The **Measure Circle** tool is used to measure a circular part in the scene. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, secondary results of the **Measure Circle** tools are the **Diameter** and the **CenterPoint** of the circle.

Fig. 117: The **Measure Circle** tool tile.

The **Measure Circle** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **Scan direction** is used to select the direction of the scans. You can choose **Out** or **In**.

The **Polarity** is used to select the brightness transition of the edge. You can choose **Dark -> Light**, **Light -> Dark** or **Both**. This specifies the type of edge in combination with the **Scan direction**.

The **Edge selection** chooses the **First**, **Last** or **Best** edge with respect to the scan direction.

Search rays selects the number of radial search rays used to find the circle. A number of 36 would position a search ray every 10 degrees.

Smoothing selects the sharpness of the edge. The slider allows you to select gradual values between **Blurred** and **Sharp**.

The **Edge strength** display a gradient curve of the edge and allow you to select the desired strength of edges.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Circle

The primary tool result is **True/Ok** if the diameter is between **Diameter minimum** or **Diameter maximum** and the **Minimum completeness** is exceeded, otherwise the result is **False/Not Ok**. The completeness can only be calculated properly, if the center of the region of interest is close to the center of the circle.

Image display with overlay

At the right, the image is displayed with an ROI on top. Inside the ROI, the found edge points are visualized with green or red (outlier) dots, and the fitted circle is displayed.

The region of interest can be moved (by dragging it inside), or resized (by dragging the border lines).
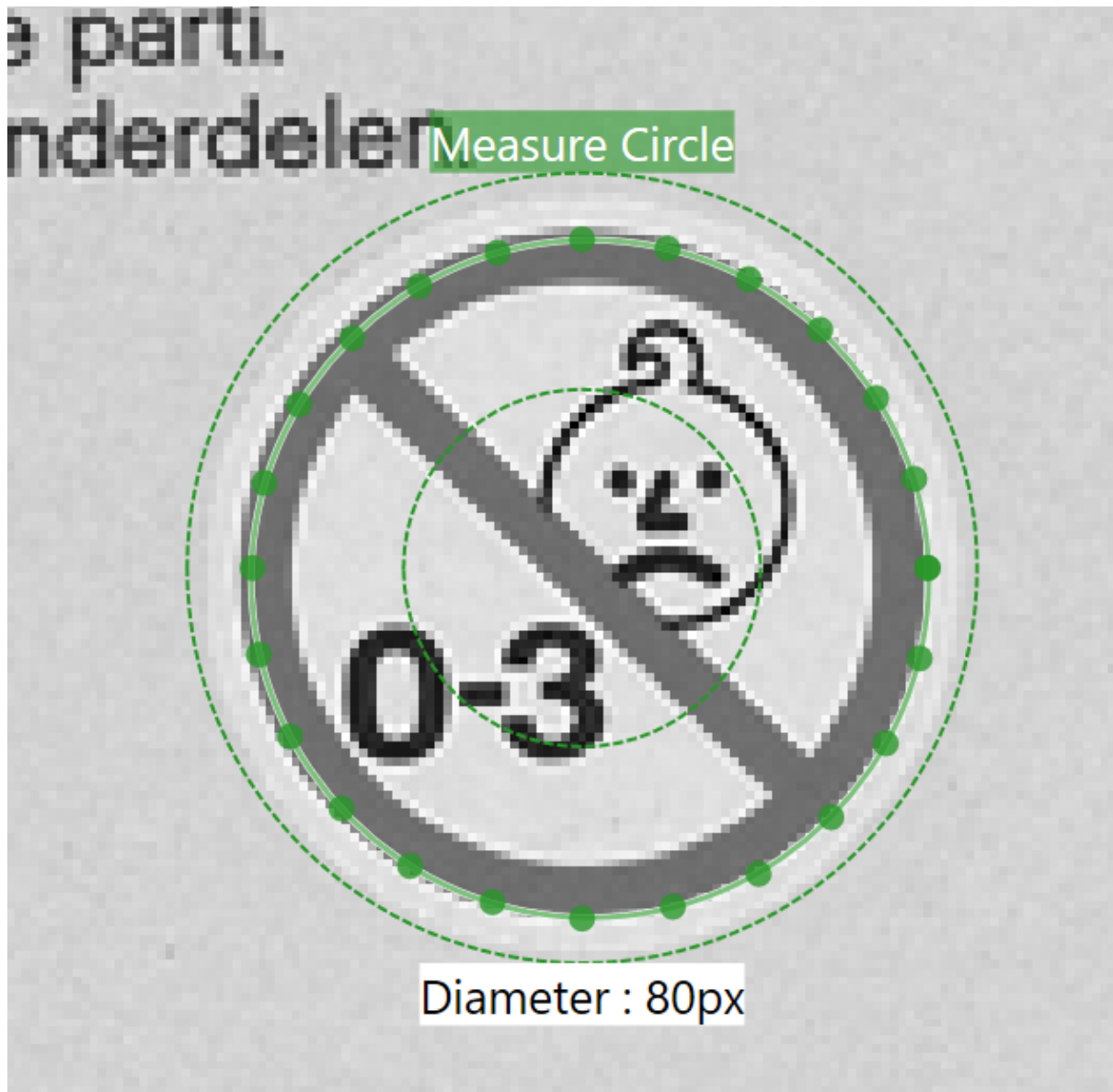
Fig. 118: The **Measure Circle** tool gui.

4.3.22 Detect Contrast

Fig. 119: The **Detect Contrast** tool icon.

The **Detect Contrast** tool is used to measure the contrast inside a region of interest.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Detect Contrast** tool is the **Contrast** value.

Fig. 120: The **Detect Contrast** tool tile.

The **Detect Contrast** tool can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **shape of the ROI** can be selected as **rectangle**, **circle** or **ring**. Depending on the selection, the shape of the ROI set interactively on the right side will be adjusted.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or, in the case of the shape selection **rectangle**, also rotated (by dragging the rotation point in the middle of the top border).

The contrast is measured inside this region of interest.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

The region of interest can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

The primary tool result is **True/Ok** if the contrast is between **Minimum constrast** or **Maximum constrast**, otherwise the result is **False/Not Ok**.

With the parameter **sensitivity** a certain number of darkest and lightest pixels can be excluded from the calculation of the contrast. The lower the sensitivity, the more dark and more light pixels are ignored.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the contrast is determined and displayed.

Fig. 121: The **Detect Contrast** tool bui.

4.3.23 Measure Distance

Fig. 122: The **Measure Distance** tool icon.

The **Measure Distance** tool is used to measure a distance in the scene. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Measure Distance** tools is the **Distance**.
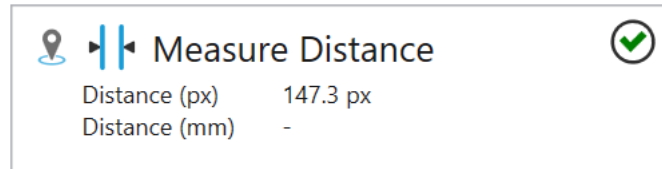
Fig. 123: The **Measure Distance** tool icon.

The **Measure Distance** tool uses can use the pose from any locate tool in the pipeline, by selecting a specific locate tool using the pin.

Parameters

The **Scan direction** is used to select the direction of the scans. You can choose **Left -> Right**, **Right -> Left**, **Top -> Bottom** or **Bottom -> Top**.

The **Polarity** is used to select the brightness transition of the edge. You can choose **Dark -> Light**, **Light -> Dark** or **Both**. This specifies the type of edge in combination with the **Scan direction**.

Smoothing selects the sharpness of the edge. The slider allows you to select gradual values between **Sharp** and **Blurred**.

The **Edge Strength** displays a gradient curve of the edge and allow you to select the desired strength of edges.

Image preprocessing can be used to select an optional preprocessing operation in the tool. **Separate objects (Open)** performs a morphological opening, **Connect objects (Close)** performs a morphological closing, **Shrink objects (Erode)** performs an erosion, **Grow objects (Dilate)** performs a morphological dilation. It should be noted that the morphological operations depend on the brightness of the objects: **Separate objects (Open)** applies to bright objects, applied to dark objects the effect is opposite, i.e. it connects dark objects. Just like **Separate objects (Open)** and **Connect objects (Close)**, **Shrink objects (Erode)** and **Dilate objects (Dilate)** are also complementary. The **Gradient filter (Sobel)** enhances edges and **Smooth noise (Lowpass)** reduces noise.

If calibration data has been saved for the sensor, this data can be used for conversion from Image coordinates (px) in world coordinates (mm). With **Apply Calibration** the conversion of image coordinates (px) to world coordinates (mm) is turned on.

These controls only appear if calibration data has been saved.

Distance

The primary tool result is **True/Ok** if the **Distance** is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**. The comparison can be done in pixels (px) or millimeters (mm) depending on your choice of **Unit**, the latter is only possible if the sensor has been calibrated.

Image display with overlay

At the right, the image is displayed with a rectangular ROI on top. Inside the ROI, the measured line is displayed and the length in pixels is shown.

The ROI (region of interest) can be moved (by dragging it inside), resized (by dragging the border lines or corners) or rotated (by dragging the rotation point in the middle of the top border).

The ROI must be as perpendicular as possible to the edges to be measured, i.e. optimally under 90°. If the search lines become close to 0° the edges are more smoothed by averaging and the results become less precise.
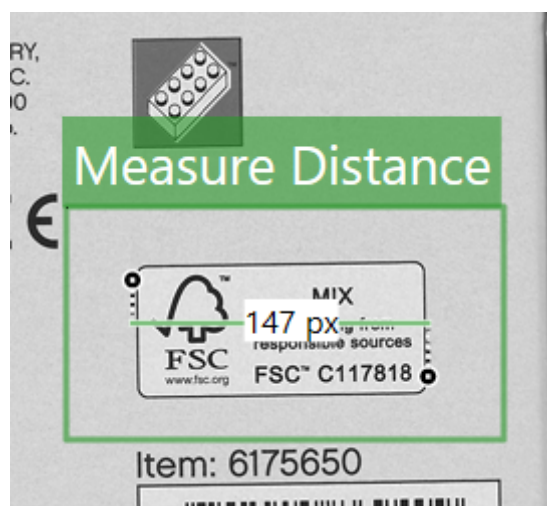
Fig. 124: The **Measure Distance** tool gui.

4.3.24 Measure Point to Line Distance

Fig. 125: The **Measure Point to Line Distance** tool icon.

The **Measure Point to Line Distance** tool is used to measure the distance between a point and a line. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Measure Point to Line Distance** tool is a **Distance**.
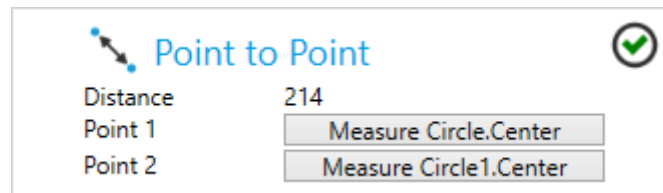
Fig. 126: The **Measure Point to Line Distance** tool tile.

The **Measure Point to Line Distance** tool uses a point and a line from tools that measure points or lines(i.e. Measure Circle, Locate Edge).

The two points can be selected by clicking on the respective buttons.

Distance

The primary tool result is **True/Ok** if the count is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**.
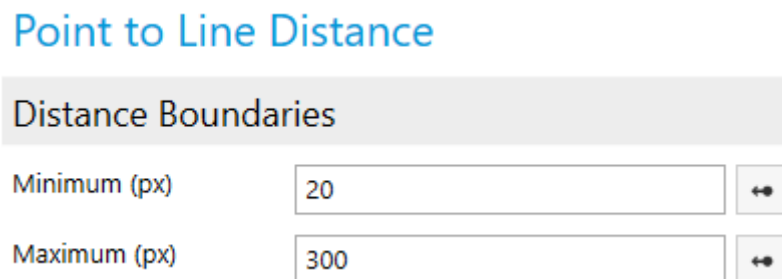
Fig. 127: The **Measure Point to Line Distance** tool parameters.

At the right, the image is displayed with a line display on top. In addition to the line, the length in pixels is shown.

Fig. 128: The **Measure Point to Line Distance** tool gui.

4.3.25 Measure Point to Point Distance

Fig. 129: The **Measure Point to Point Distance** tool icon.

The **Measure Point to Point Distance** tool is used to measure the distance between two points. This tool is only available for the **CS-60**.

Besides the primary **Ok/Not Ok** result, a secondary result of the **Measure Point to Point Distance** tool is a **Distance**.
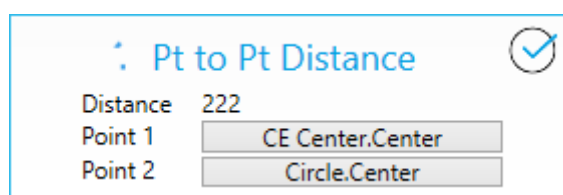

Fig. 130: The **Measure Point to Point Distance** tool tile.

The **Measure Point to Point Distance** tool uses two points from tools that measure points (i.e. Measure Circle, Locate Edge).

The two points can be selected by clicking on the respective buttons.

Distance

The primary tool result is **True/Ok** if the count is between **Minimum** or **Maximum**, otherwise the result is **False/Not Ok**.
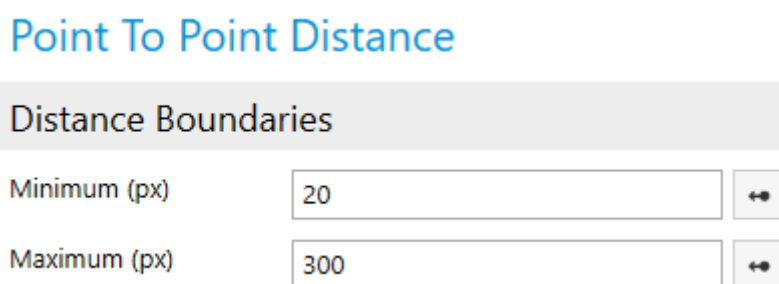
Fig. 131: The **Measure Point to Point Distance** tool parameters.

At the right, the image is displayed with a line display on top. In addition to the line, the length in pixels is shown.
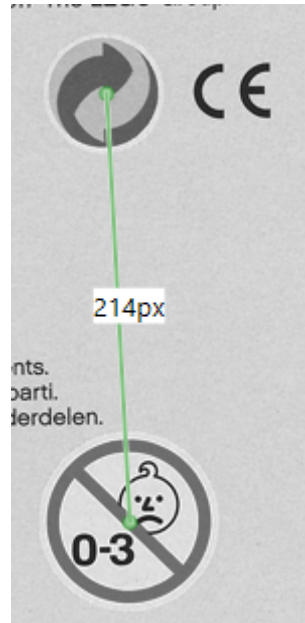
Fig. 132: The **Measure Point to Point Distance** tool gui.

4.3.26 Logic - Results

Fig. 133: The **Logic** tool icon.

The **Logic** tool is used to define the logic and derive an output from the results of different tools.

The **Logic** tool is special: it is always available and it is always the last tool in the pipeline.

Fig. 134: The **Logic** tool pane.

The **Logic** tool allows you to produce inspection results by creating a function block diagram.

4.3.27 Results

By default, the overall **Result** of the Logic tool is a combination of the results of all tools in the pipeline. Only when all tools return **Ok** (or logic **1**), the result of the **Logic** tool itself - that is the result of the overall inspection - is **Ok**.

However, you can override this by implementing your own logic on the **Results** tab. To override, select a tool on the **Tools** menu to add it to the function plan canvas. You can then use the predefined **Result** of the tool. Here is an example:
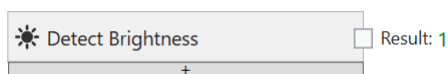
Fig. 135: A tool node.

Alternatively, you can create a comparison expression of any secondary tool result:
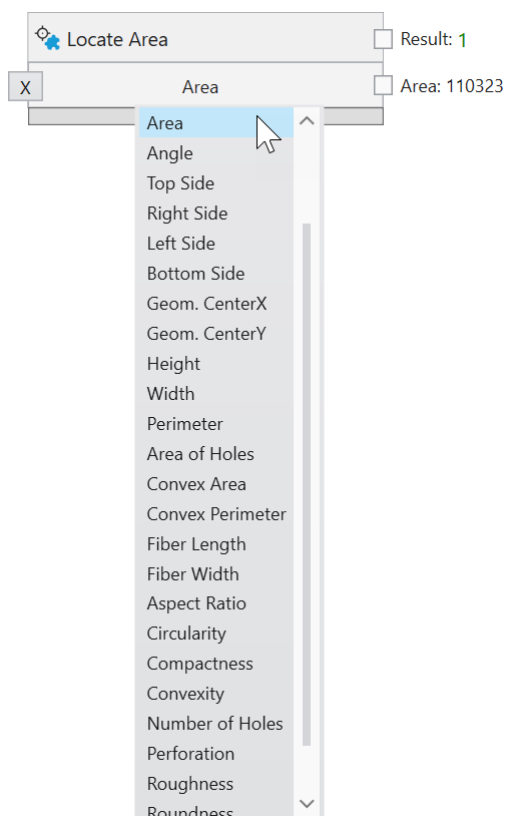
Fig. 136: Create additional boolean outputs with secondary tool results.

Any of these results of different tools can be combined with logical AND, OR or NOT nodes that you can select from the **Logic Operators** menu:
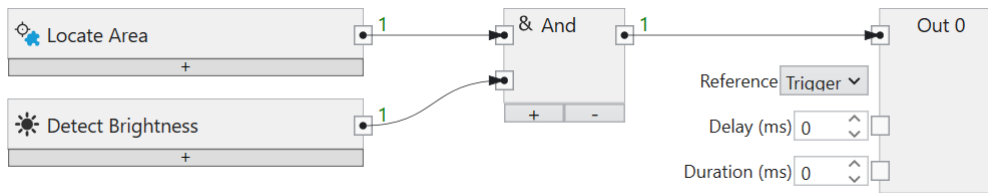
Fig. 137: Logical combination of several tools.

The logical combination can be arbitrarily complex.

Finally, the result of the logical combination can be connected to the **Result**-node, and will then be used as the overall inspection result.

4.3.28 Logic - Digital Outputs

The **Logic** tool allows you to produce inspection results by creating a function block diagram. Function block diagrams are standardized in IEC EN 61131-3.

Digital Outputs

By default, the four digital outputs of the **CS-60/ID-600** are displayed as nodes on the canvas.
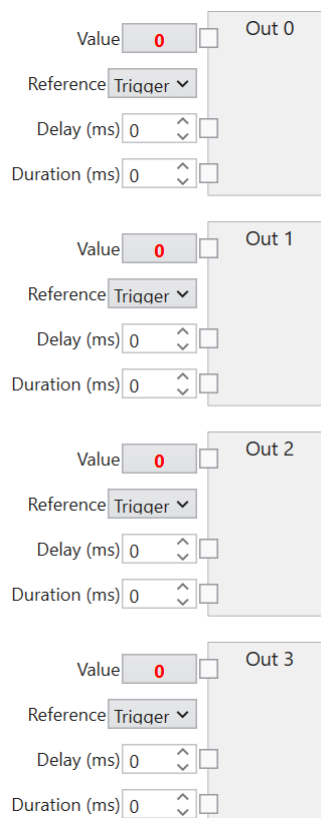
Fig. 138: The **Logic** tool digital output nodes.

The **Delay** and the **Duration** of a digital output signal can be specified (in milliseconds). The

delay is measured from the start of the image acquistion (**Reference: Trigger**) or from the end of the inspection (**Reference: EndOfInspection**), a duration of 0 means infinite duration.
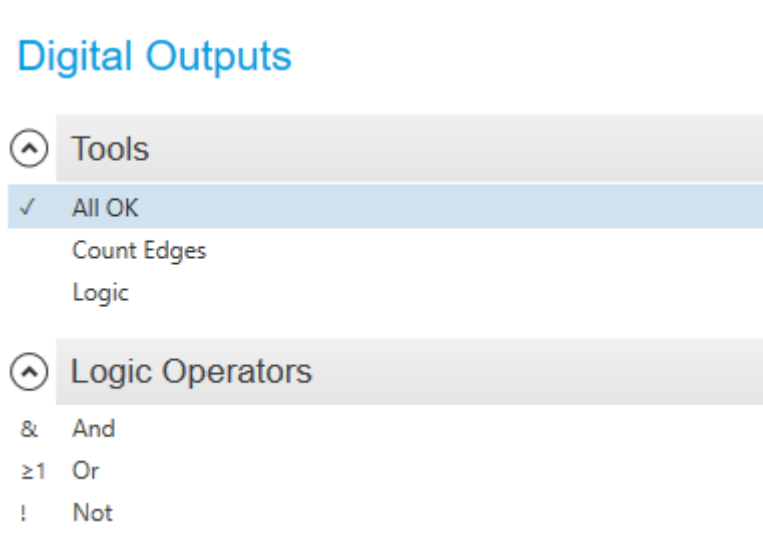
Fig. 139: The context menu of the Digital Outputs.

Tools can be dragged in from the tool list or can be selected in the context menu on the left, to create tool nodes that provide values that can be combined logically to create the desired outputs. Here is an example:
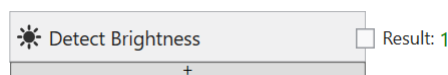
Fig. 140: A tool node.

The primary result of the tool is **True** or **False** and can be connected to any of the four outputs (by dragging from the little square output pin named **Result** of the tool to the respective input pin of the output named **Value**.

Fig. 141: Connecting a tool and an output.

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added to the canvas by selecting them in the context menu on the left or by clicking with the right mouse button.
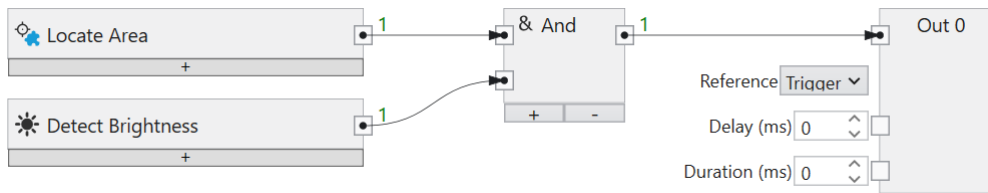
Fig. 142: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node on the canvas or a connection between two nodes can be selected and deleted with the **Del** key.

A click on the little **+** button at the bottom of a tool node adds an additional row to the tool node, where you can create a logic expression of any secondary tool result.
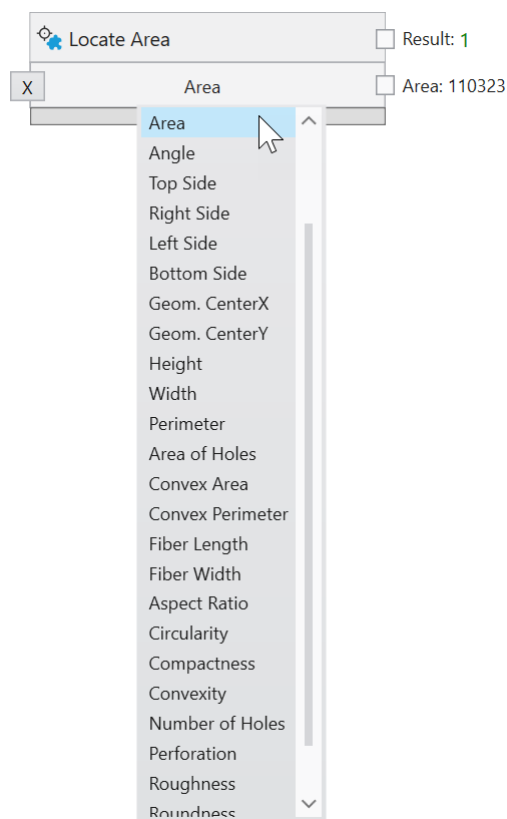
Fig. 143: Create additional boolean outputs with secondary tool results.

Enter minimum or maximum boundaries and create additional logic results from secondary numeric tool results.

The **X** button deletes a row, if it is no longer needed.

4.3.29 Logic - Fieldbus Outputs

The **Logic** tool allows you to produce inspection results by creating a function block diagram. Function block diagrams are standardized in IEC EN 61131-3.

Inspection results can be transferred to a PLC via PROFINET by linking to PROFINET outputs.

A value is linked in several steps on the **PROFINET Outputs** or **EtherNet/IP** tabs of the **Logic** tool: 1. A tool node is connected to a write node. 2. The write node is connected to a PROFINET or EtherNet/IP module in a slot.

Output data are transmitted at the end of an inspection cycle.

Linking Fieldbus Outputs

Note: the screenshots show PROFINET, but the procedures for EtherNet/IP are exactly the same, unless where explicitly noted.

Fig. 144: The context menu of the PROFINET Outputs.

Fieldbus outputs must be defined by by selecting them in the context menu on the left or by clicking with the right mouse button into the canvas and chosing **Write Bool**, **Write Byte**, **Write Int32**, **Write Float** or **Write String**.

Tools can be selected in the context menu on the left, to create tool nodes that provide values that can be connected to the PROFINET output nodes. Here is an example:
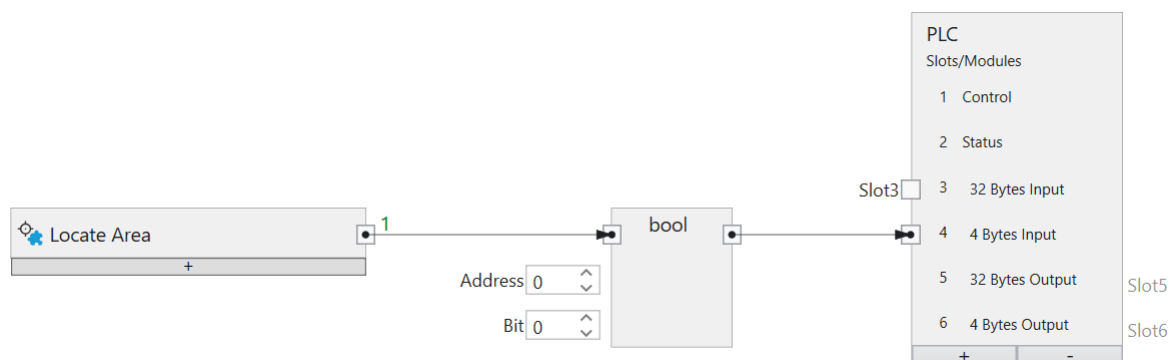
Fig. 145: A tool node for PROFINET.

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added to the canvas by clicking with the right mouse button.
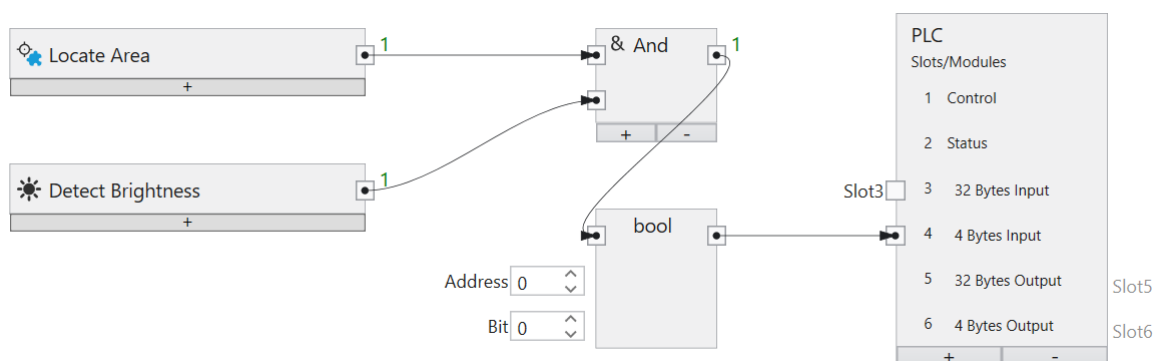
Fig. 146: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node or a connection between two nodes can be selected and deleted with the **Del** key.

In addition to logical values, PROFINET or EtherNet/IP can also transfer numerical values or texts.

The PROFINET or EtherNet/IP output nodes (**Write Bool**, **Write Byte**, **Write Int32**, **Write Float**, **Write String**) determine at which address within a module the values are written.
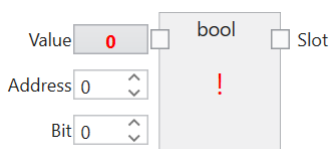
Fig. 147: A Boolean output.

The size of a boolean output is 1 byte (but you can allocate up to 8 in one byte).

Value is the value to be transferred.

Address specifies the address in the module.

Bit indicates the bit to be addressed, where 0 is the rightmost bit and 7 is the leftmost bit in the byte.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
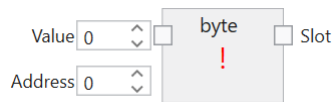
Fig. 148: A byte output.

Value is the value to be transferred.

Address specifies the address in the module.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
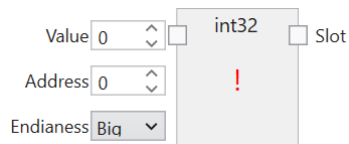
Fig. 149: An integer output.

The size of an integer output is 4 bytes.

Value is the value to be transferred.

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
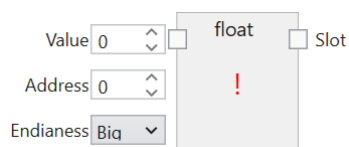
Fig. 150: A floating point output.

The size of a floating point output is 4 bytes.

Value is the value to be transferred.

Address specifies the address in the module.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).
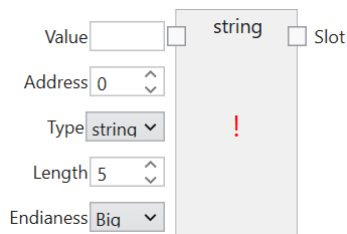
Fig. 151: A text output.

Profinet: The size of a string output is (2 + n) * s, where n is the maximum number of characters and s = 1 (for an Ansi STRING) or s = 2 (for a Unicode WSTRING). The first byte/word (Ansi/Unicode) contains the maximum number of characters the string can hold. The second byte/word (Ansi/Unicode) contains the actual number of characters the string holds.

EtherNet/IP: The size of a text input is (4 + n), where n is the maximum number of characters. The first 4 bytes are interpreted as an integer and represent the actual number of characters in the string.

Value is the value to be transferred.

Type specifies if the string is Ansi (STRING) or Unicode (WSTRING). This input is omitted with EtherNet/IP, as EtherNet/IP only supports Ansi texts.

Address specifies the address in the module.

Length specifies the number of characters in the string. A length of 1 indicates a single character.

Endianness specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first. With EtherNet/IP, this input is omitted as it is only relevant for Unicode texts.

Slot connects the node with an input module (**CS-60/ID-600** -> PLC).

The **PLC** node is used to define modules in specific slots. This setup must exactly match the PROFINET or EtherNet/IP setup on the PLC, to enable communication between the PLC and the **CS-60/ID-600**. The PLC node is shown both on the Acquisition tool and the Logic tool. Inside the PLC node, the names Input and Output for the modules are used as seen from the PLC. A PLC Output is transferred from the PLC via PROFINET or EtherNet/IP to the **CS-60/ID-600**, where it is seen as an input. A PLC Input is transferred from the **CS-60/ID-600** via PROFINET or EtherNet/IP to the PLC, on the **CS-60/ID-600** this is seen as an output.

Fig. 152: The PLC node with six modules.

As an example, six modules are selected here.

The modules are displayed graphically under the heading **Slots**. The used regions within the modules are also displayed.

C4G Mode

For the C4G mode, the predefined modules **Control** and **Status** must be removed and replaced manually by a C4G compatible setup (see chapter C4G Mode Status and Control Signal Setup). Any further modules can be attached to these modules.
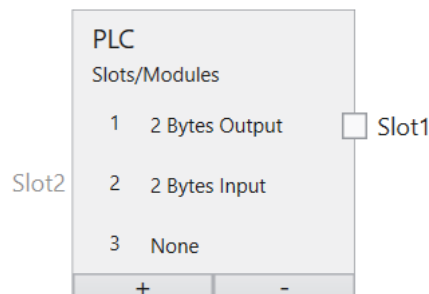
Fig. 153: C4G Mode

4.3.30 Logic - TCP/IP Command Outputs

The **Logic** tool allows you to produce inspection results by creating a function block diagram. Function block diagrams are standardized in IEC EN 61131-3.

Inspection results can be transferred via TCP/IP commands to any counterpart by linking to TCP/IP variables outputs.

A value is linked on the **TCP/IP Varibles Outputs** tab of the **Logic** tool: A tool node is connected to a TCP/IP variable in the Outputs node.

Output data are transmitted at the end of an inspection cycle.

Linking TCP/IP Variables Outputs

TCP/IP Variables Outputs

Tools

All OK

Acquisition

Locate Area

Detect Brightness

Logic

Output

Logic Operators

Arithmetic

Text

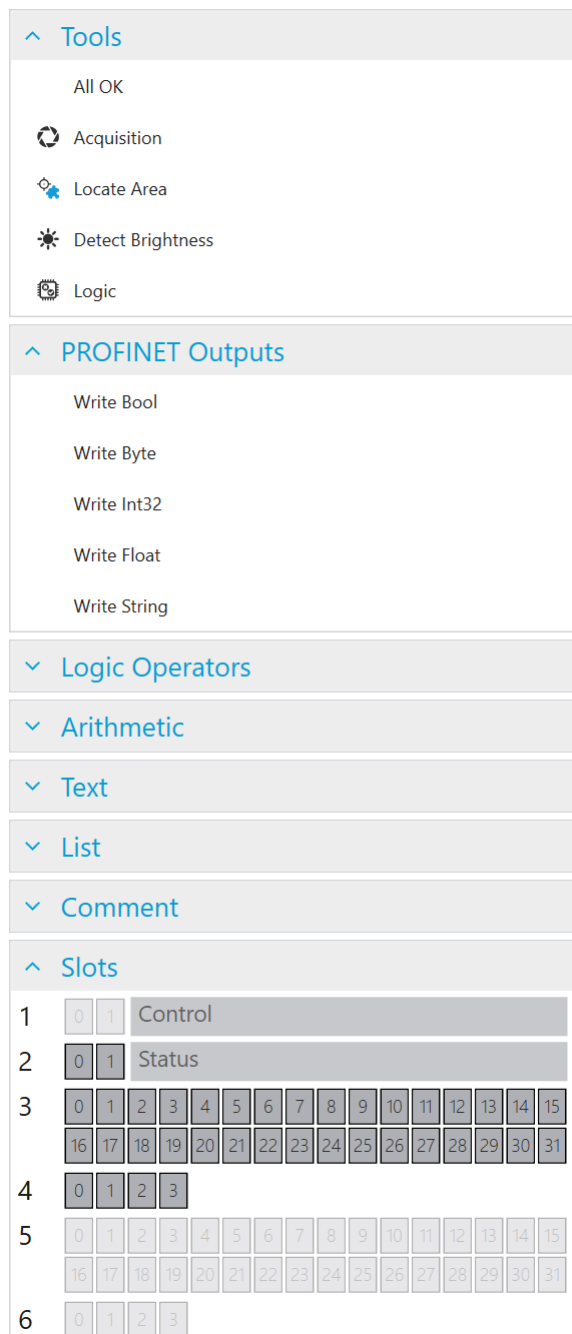**List**

Comment
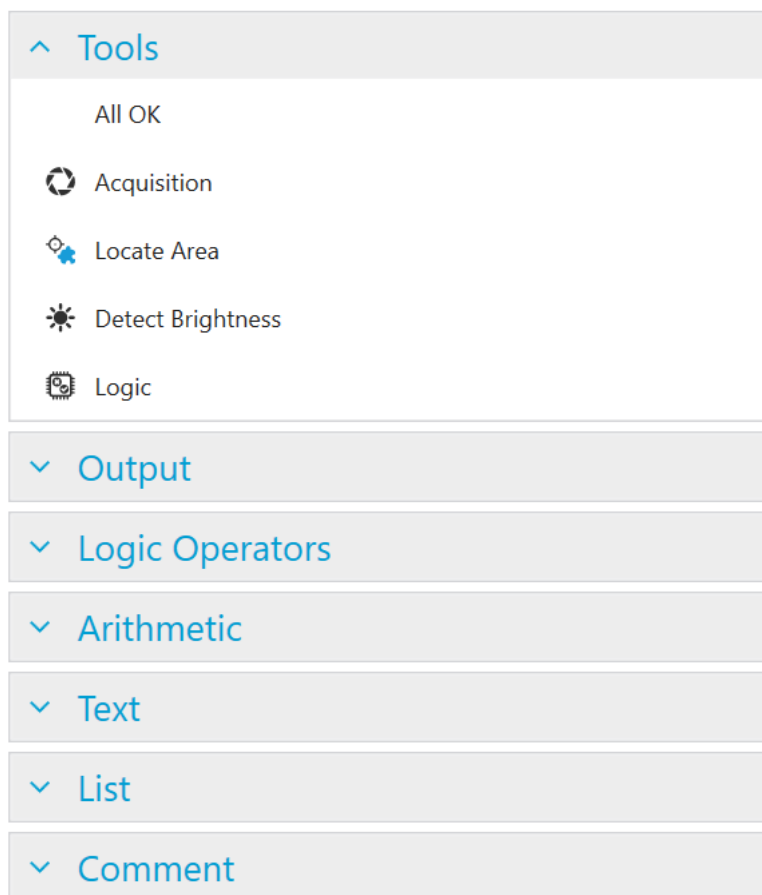
Fig. 154: The context menu of the TCP/IP Variables Outputs.

PROFINET outputs must be defined by by selecting them in the context menu on the left or by clicking with the right mouse button into the canvas and chosing **Write Bool**, **Write Byte**, **Write Int32**, **Write Float** or **Write String**.

Tools can be selected in the context menu on the left, to create tool nodes that provide values that can be connected to the TCP/IP Outputs node. Here is an example:

Detect Brightness — Result: 1 — 239.9 — Average — Variables — Brightness
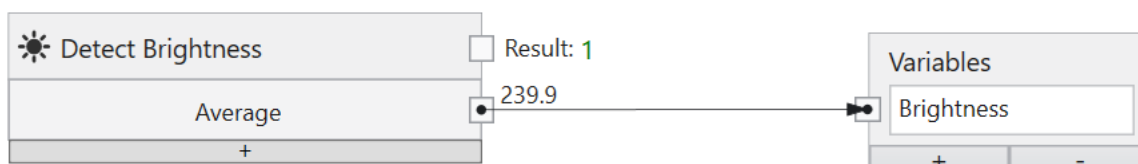
Fig. 155: A tool node for TCP/IP Variables Outputs.

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added to the canvas by clicking with the right mouse button.
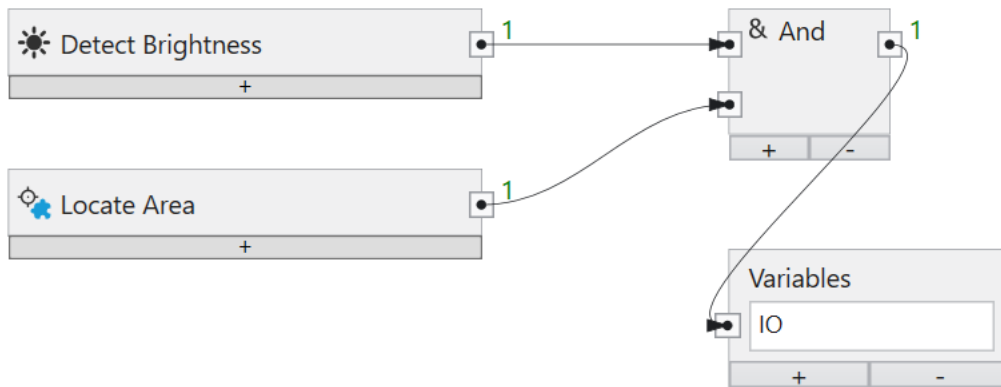
Fig. 156: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node or a connection between two nodes can be selected and deleted with the **Del** key.

In addition to logical values, TCP/IP commands variables can also transfer numerical values or texts.

4.3.31 Logic - FTP

The **Logic** tool allows you to produce inspection results by creating a function block diagram. Function block diagrams are standardized in IEC EN 61131-3.

FTP/SFTP

The **CS-60/ID-600** vision sensor can optionally save images to a server using FTP (File Transfer Protocol) or SFTP (Secure File Transfer Protocol). The FTP option must be activated (see **Protocol settings** of the **CS-60/ID-600**) and the address and port of the FTP server must be defined.

The FTP or SFTP protocol is selected by prefixing the host with `ftp: //` or `sftp: //`.

The user credentials (username and password) must also be provided so that a connection to the FTP/SFTP server can be established. Communication between the built-in optional FTP/SFTP client takes place with FTP via port 21, with SFTP via port 22, the standard ports for FTP and SFTP.
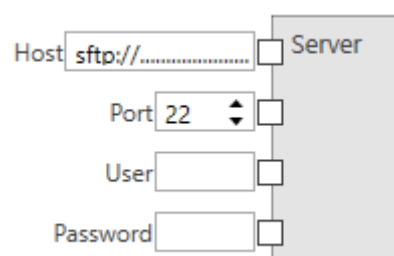
Fig. 157: The FTP/SFTP server settings.

When FTP is enabled, a job can upload images to an FTP/SFTP server. The address and login information of the FTP/SFTP server must be configured in the job on the FTP tab of the logic tool.

Uploading Images

Images of inspections can be saved into two directories: **OK folder** is used for inspections that were Ok, **NOK folder** is used for failed inspections. If the same name is used for both directories, both types of images are uploaded to the same directory. The directory and/or the file name (**OK file name** and **NOK file name**) of the **Image Upload** node can be entered as fixed values or kept variable by inserting the **Text** node (see description of **Text Node**). If using the **Text** node, connect the output of the **Text** node with the **OK folder**, **NOK folder**, **OK file name** or **NOK file name** inputs of the **Image Upload** node.

If the FTP/SFTP server is not available for uploading, the images are cached on the **CS-60/ID-600** until the maximum number of images has been reached. If the maximum number has been reached, first older OK pictures and then older NOK pictures are overwritten, so that these pictures are lost forever for the FTP/SFTP upload. This behavior gives priority to NOK images.

All images that are in the FTP cache and for some reason could not be uploaded can be saved on the PC with nVision-i.

Uploading Text

Besides the option to transfer images via FTP, there is also the option to attach text to a file (or to create a new file if it does not exist yet). To do this, select the **Text** node under **Upload** in the menu on the left. This node can be configured exactly like the **Image Upload** node (see **Uploading Images**). Additionally the **Text Upload** node has an input named **Text** where any text can be appended to the specified file. For this purpose the **Text** node described under **Text Node** can be used. The output of the **Text** node has to be connected to the corresponding input of the **Text Upload** node.

Text Node

Add a **Text** node to the function block diagram and type a name or use the dropdown list to select amoung **Acquisition ID**, **Job name**, **Job slot**, **SensorName**, **Sensor serialnumber**, **Year**, **Month**, **Day**, **Hour**, **Minute**, **Second** and **Millisecond**. The options **Carriage return**, **Line feed**, **Carriage return & Line feed**, and **Tab** are available for formatting. If you want to remove one of the predefined items, choose the first item **Disconnect** from the dropdown list. You can also connect results of tools to the **Text** node by connecting a corresponding output of a tool to one of the inputs of the **Text** node. You can add as many inputs to the **Text** node as desired by pressing the "+" button or remove the last one by pressing the "-" button.

4.4 Graphical Programming

4.4.1 Graphical Programming

Some aspects of job creation are carried out in the **CS-60/ID-600** vision sensor using graphical programming. Examples are the linking of the logical digital inputs or input values transmitted via fieldbusses or TCP/IP in the Acquisition tool. Graphical programming is also used in the Logic tool to calculated output values for the digital outputs or for PROFINET and TCP/IP or FTP.

Pipelines

Pipelines are the programs that make up nVision-i **Jobs**. Data flows through pipelines, much like water flows through pipes.

Nodes are the building blocks of pipelines. Nodes have **Pins** to which nodes are connected using **Connections**.

Here is an example of a pipeline made up of two nodes:

Fig. 158: A pipeline made up of two nodes.

A pipeline represents a directed graph. Data flows in the direction indicated by the arrows at the end of the links. Cycles in the graphs are prohibited and are cancelled automatically.

Pipelines run live. Each time a value changes, the pipeline is executed again and generates new outputs. Nodes only run when they need the values they produce.

Nodes

Nodes are the basic building blocks in pipelines.

A node can have **Input Pins** and **Output Pins**, although not every node has both. The pins are used to connect nodes using **Connections**.
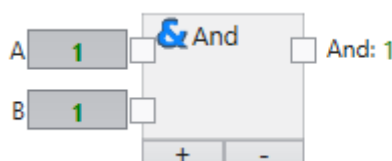
Fig. 159: A node with input and output pins.

Pins

Pins are typed. Connections are only possible between pins with compatible **Types**.

Pins are compatible if their types are the same, or if the type flowing on the connection can be converted automatically.

Pins of incompatible types cannot be connected.

Types

The data flowing in pipelines is typed.

Type	Description
`Boolean`	A Boolean value that can have the two binary states `True` (1) and `False` (0).
`Byte`	An integer value between 0 and 255 (inclusive).
`Integer`	An integer value between -2147483648 and 2147483647 (inclusive).
`Single`	A floating point value between -3.40282347 x 10^38 and 3.40282347 x 10^38.
`Double`	A floating point value between -1.7976931348623157 x 10^308 and 1.7976931348623157 x 10^308.
`String`	A string of characters, e.g. `Text`.

Conversion

The connections between pins represent the paths for the data flow. The data can only flow if the types of the pins are the same or if they can be converted automatically.

Extending conversions can be performed automatically. The conversion of an Int32 value into a float value is expanding and is carried out automatically.
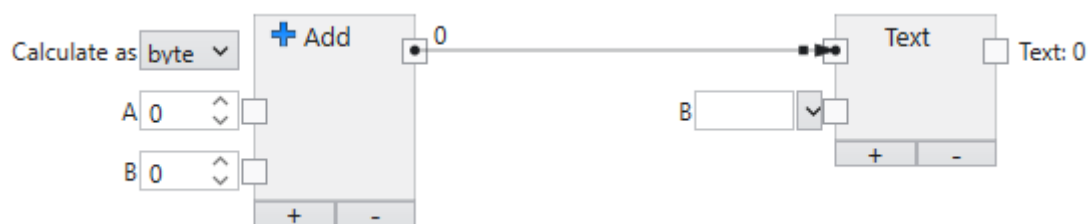
Fig. 160: Example of convertible types.

A numeric value can be automatically converted to text. The conversion is visualized by a little dot near the end of the connection arrow.

Narrowing conversions cannot be carried out automatically. The conversion of a float value into an Int32 value would cut off the decimal places and is therefore not carried out automatically.
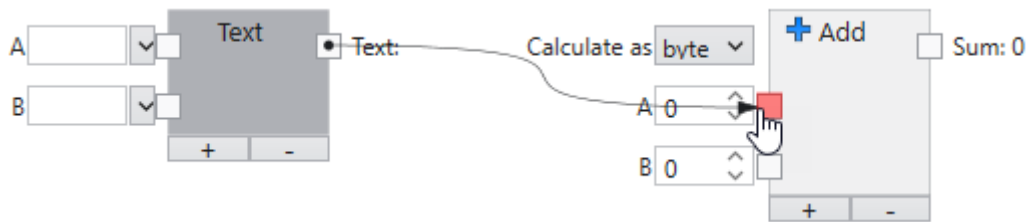
Fig. 161: Example of incompatible types.

A text cannot be automatically converted into a numeric value. This is visualized by a pin filled in red.

4.5 Nodes

4.5.1 Add

Adds two of more numbers.
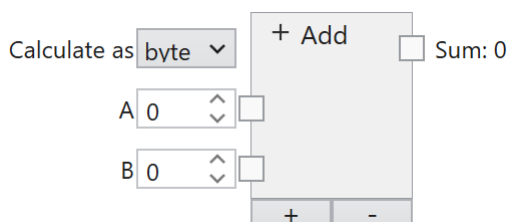
The numbers are added. The addition takes place with a specified data type.

Inputs

Calculate as (Type: `String`)

The calculation takes place in the specified data type: byte, int or float. Byte allows integer values between 0 and 255, int allows integer values between -2.147.483.648 and 2.147.483.647 and float allows floating point numbers between -3.4 x 10^38 to 3.4 x 10^38.

A...Z (Type: `Double`)

The numbers to add. All numeric data types are converted to double as needed.

Outputs

Sum (Type: `Byte|Int32|Single`)

The sum of the numbers, according to the specified data type. If the result cannot be represented in the specified data type, the node does not produce a numerical result.

4.5.2 All Ok

Provides the summarized result of all tools.
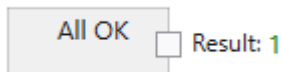
Outputs

Result (Type: `Boolean`)

The AND combination of the results of all tools.

4.5.3 And

Logical AND operation of two or more boolean values.
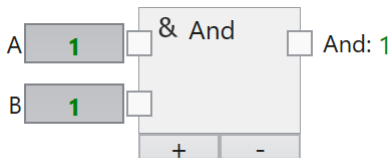
The logical AND of the boolean input values.

Inputs

A...Z (Type: `Boolean`)

The boolean input values.

Outputs

And (Type: `Boolean`)

The AND operation of the inputs. The result is True/1 if all inputs are True/1, otherwise the result is False/0.

4.5.4 BinarySelector

Chose one of two inputs, depending on the state of a boolean input.

Inputs

Selector (Type: `Boolean`)

The selector. If this input is `True`, the value at the *TrueValue* input is shown at the output. If this input is `False`, the value at the *FalseValue* input is shown at the output.

TrueValue (Type: `Object`)

This value is shown at the Value output, when the value at the *Selector* input is `True`.

FalseValue (Type: `Object`)

This value is shown at the Value output, when the value at the *Selector* input is `False`.

Outputs

Value (Type: `Object`)

The value of either the value at the *TrueValue* input or the value at the *FalseValue* input, depening on the value at the *Selector* input.

Comments

The types of the inputs and the outputs can be anything, i.e. they must at least be of type `Object`. If any type is more concrete, the other types must be of the same type (or null, i.e. not connected).

The most straightforward use of the **BinarySelector** node is to select from two values,

Another use is to inhibit execution of a part of a pipeline by feeding `null` into the rest of the pipeline. If the input is left open, `null` is propagated to the output, and in consequence the nodes that depend on it cannot be executed.

4.5.5 Comment

A comment in a function block diagram.

The comment is used to describe the purpose of a functional diagram or a part of it. The comment field can be expanded or collapsed.
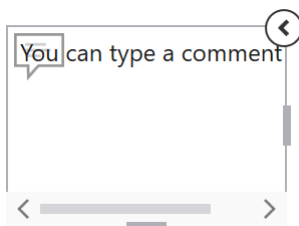
4.5.6 Compare

Compares two numerical values.
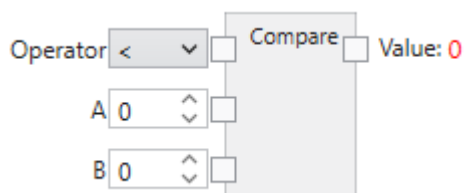
The numbers are compared with different operators. If the operator applies, True/1 is returned or if it does not apply, False/0 is returned.

Inputs

Operator (Type: `String`)

The comparison operator. '<', '<=', '==', '!=', '>=' and '>' are available.

A,B (Type: `Double`)

The numbers to be compared. All numeric data types are converted to double as needed.

Outputs

Value (Type: `Boolean`)

The comparison result.

4.5.7 Contains

Checks whether a text is contained in another text.
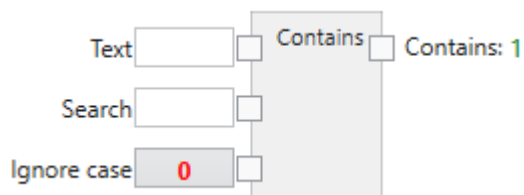
It is checked whether the search text is contained in the text and True/1 or False/0 is returned.

Inputs

Text (Type: `String`)

The text in which the search text is searched for.

Search (Type: `String`)

The search text.

Ignore case (Typ: `Boolean`)

Upper or lower case is ignored if this value is set to True/1.

Outputs

Contains (Type: `Boolean`)

The search result: True/1 if the search text is included in the text or False/0 if it is not included. An empty search text is contained in all texts (including an empty text).

4.5.8 In

Reads data from a digital input.

In 0 ☐ Value: 0
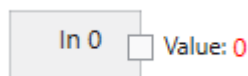
The Vision Sensor **CS-60/ID-600** has two digital inputs that can be linked. The input nodes are displayed on the Digital Inputs page of the Acquisition tool. In order to use them for linking, they need to be named. An input is named by adding a link node (selection in the menu on the left or with a right-click on the drawing area and selection of link), connecting the input node with the link node and entering the corresponding name in the link node.

The digital input is read in at the beginning of the inspection cycle, roughly at the time of the trigger signal.

Outputs

Value (Type: `Boolean`)

The logical value of the input at the time of the trigger signal.

4.5.9 Out

Writes data to a digital output.

Value 0 ☐ Out 0
Reference Trigger ∨
Delay (ms) 0 ⌄ ☐
Duration (ms) 0 ⌄ ☐
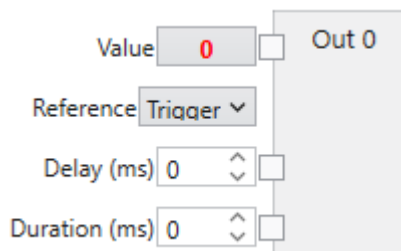
The Vision Sensor **CS-60/ID-600** has four binary digital outputs that can be switched depending on the result of the inspection.

If digital outputs are used to signal the job change, they cannot be switched as inspection results at the same time.

Inputs

Value (Type: `Boolean`)

The logical value to be written to the output.

Reference (Type: `String`)

The delay of the output signal can refer to the **Trigger** time, or to the **End of inspection**.

Delay (ms) (Type: `UInt32`)

The delay of the switch point in relation to the time of the trigger or the end of the inspection. A delay of `0` means immediate switching (here only after the inspection has ended, because the value can only be determined at this point in time).

Duration (ms) (Type: `UInt32`)

The duration of the logic signal. A duration of `0` results in a permanent signal.

4.5.10 Divide

Divides two numbers.
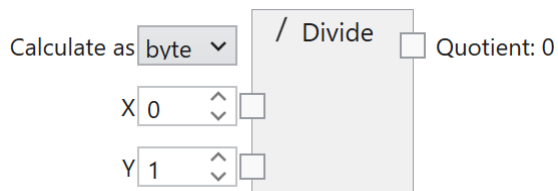
The numbers are divided. The division takes place with a specified data type.

Inputs

Calculate as (Type: `String`)

The calculation takes place in the specified data type: byte, int or float. Byte allows integer values between 0 and 255, int allows integer values between -2.147.483.648 and 2.147.483.647 and float allows floating point numbers between -3.4 x 10^38 to 3.4 x 10^38.

X (Type: `Double`)

The dividend. All numeric data types are converted to double as needed.

Y (Type: `Double`)

The divisor. All numeric data types are converted to double as needed.

Outputs

Quotient (Type: `Byte|Int32|Single`)

The quotient of the numbers, according to the specified data type. If the result cannot be represented in the specified data type, the node does not produce a numerical result.

4.5.11 Image Upload

Configures the upload of images.
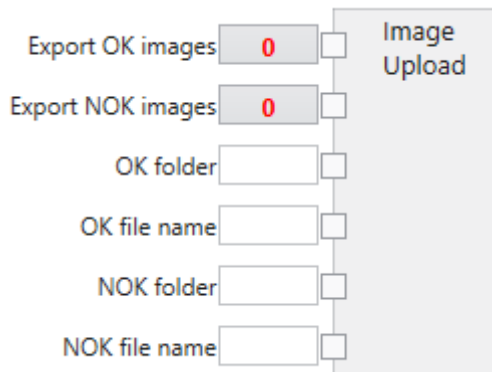
Inputs

Export OK images (Type: `Boolean`)

Configures whether IO pictures should be exported.

Export NOK images (Type: `Boolean`)

Configures whether IO pictures should be exported.

OK folder (Type: `String`)

The folder name to which the IO images are to be exported.

OK filename (Type: `String`)

The filename for IO images.

NOK folder (Type: `String`)
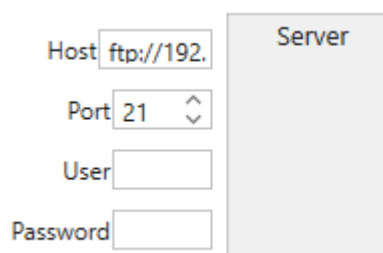
The folder name to which NOK images are to be exported.

NOK filename (Type: `String`)

The filename for IO images.

4.5.12 Server

Configures the FTP server settings.

Inputs

Host (Type: `String`)

The FTP or SFTP host, e.g. `ftp://192.168.3.15` or `sftp://192.168.3.15`. The IP addresses must be adjusted accordingly to the circumstances.

Port (Type: `Int32`)

The FTP or SFTP port, z.B. `21` or `22`.

User (Type: `String`)

The user name for logging into the FTP or SFTP server.

Password (Type: `String`)

The password for logging into the FTP or SFTP server.

4.5.13 Text Upload

Configures the upload of text.

Inputs

Export OK images (Type: `Boolean`)

Configures whether text should be exported to IO images.

Export NOK images (Type: `Boolean`)

Configures whether texts for NOK images should be exported.

OK folder (Type: `String`)

The folder name in which the texts for IO images are to be exported.

OK filename (Type: `String`)

The file name for texts for IO images.

NOK folder (Type: `String`)

The folder name in which texts for NOK images are to be exported.

NOK filename (Type: `String`)

The file name for texts for NIO images.

Text (Type: `String`)

The text to write.

4.5.14 Get Item

Selects an item from a list.
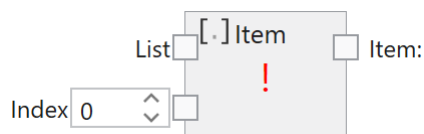
A list consists of several elements in a row. This node selects a specific element from the list according to its position in the list.

Inputs

List (Type: `List<T>`)

The list of items. The type can be anything, but all items in the list have the same type.

Index (Type: `Int32`)

The zero-based index of the item. If the index is out of range, an error will be shown.

Outputs

Item (Type: `T`)

The specific item of the list.

4.5.15 Link

Creates a link.
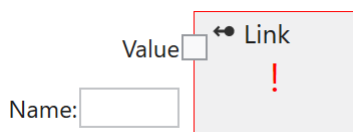
A link represents a relationship between an input value and the usage of this value in subsequent tools. Input values can be provided by digital inputs, Profinet inputs or TCP/IP variable inputs. In order for them to be used in subsequent tools, they must be named.

Inputs

Value (Type: `Object`)

The value of the link. It is usually provided by connecting to a digital input, a Profinet input or a TCP/IP variable input.

Name (Type: `String`)

The name of the link. The name must be unique within a job. The value is available under this name in subsequent tools.

4.5.16 IsNull

Returns `True`, if there is no object (`Null`) at the input.

Inputs

Object (type: `Object`)

If this input is `Null`, the value `True` is displayed at the output. If this input is a valid object of any type, the value `False` is displayed at the output.

Outputs

Value (type: `Boolean`)

`True` or `False`, depending on the value at the *Object* input.

Comments

The **IsNull** node is often used in conjunction with a **BinarySelector**, for example to determine a default value in the event that a value is `Null`.

4.5.17 Join

Combines a list of texts into a string.
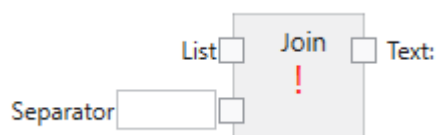
A list of texts is converted into a string.

Inputs

List (Type: `List<String>`)

Die Liste von Texten.

Separator (Type: `String`)

The separators are optionally inserted between the list elements. The following characters have special meanings:

- \t inserts a tab
- \r inserts a carriage return
- \n inserts a line feed

Outputs

Text (Typ: `String`)

The combined text.

4.5.18 Last

Selects the last item from a list.
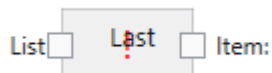
A list consists of several elements in a row. This node selects the last item from the list.

Inputs

List (Type: `List<T>`)

The list of items. It can be of any type, but all items in the list are of the same type.

Outputs

Last (Type: `T`)

The last item on the list.

4.5.19 Revert

Reverses the order of items in a list.

A list consists of several elements in a row. This node reverses the order of the list items.

Inputs

List (Type: `List<T>`)

The list of items. It can be of any type, but all items in the list are of the same type.

Outputs

Reverse (Type: `List<T>`)

The list of items in reverse order.

4.5.20 Sort

Sorts a list.
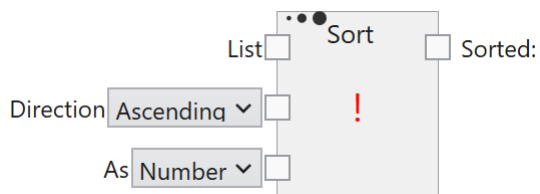
A list consists of several elements in a row. This node sorts the list according to its content.

Inputs

List (Type: `List<T>`)

The list of items. It can be of any type, but all items in the list are of the same type.

Direction (Type: `String`)

Ascending or **Descending** specifies the sorting direction.

Outputs

Sorted (Type: `List<T>`)

The list sorted in ascending or descending order.

4.5.21 Multiply

Multiplies two or more numbers.
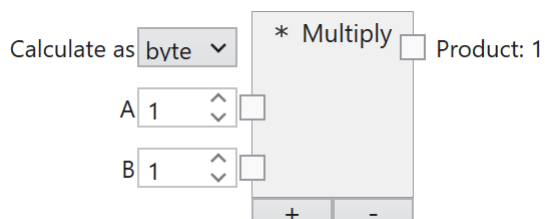
The numbers are multiplied. The multiplication takes place with a specified data type.

Inputs

Calculate as (Type: `String`)

The calculation takes place in the specified data type: byte, int or float. Byte allows integer values between 0 and 255, int allows integer values between -2.147.483.648 and 2.147.483.647 and float allows floating point numbers between -3.4 x 10^38 to 3.4 x 10^38.

A...Z **(Type:** `Double`**)**

The numbers to multiply. All numeric data types are converted to double as needed.

Outputs

**Product (Type:** `Byte|Int32|Single`**)**

The product of the numbers, according to the specified data type. If the result cannot be represented in the specified data type, the node does not produce a numerical result.

4.5.22 Not

Logic negation of a boolean value.
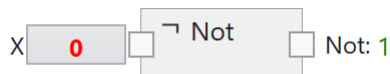
The logical value is negated.

Inputs

X(Type: `Boolean`**)**

The boolean input value.

Outputs

Not (Type: `Boolean`**)**

The negated input. The result is True/1 if the input is False/0, otherwise the result is False/0.

4.5.23 Or

Logical OR operation of two or more boolean values.
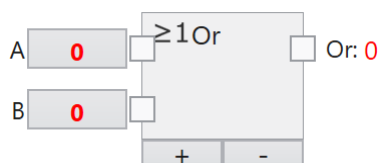
The logical OR of the boolean input values.

Inputs

A...Z (Type: `Boolean`)

The boolean input values.

Outputs

Or (Type: `Boolean`)

The OR operation of the inputs. The result is True/1 if at least one input is True/1, otherwise the result is False/0.

4.5.24 Read Bool

Reads in a boolean value from a PLC.
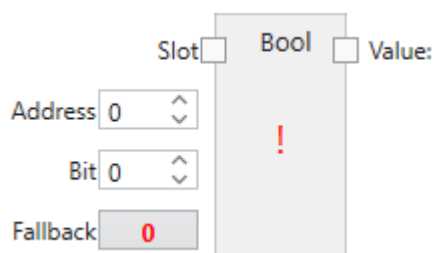
A boolean input is connected to a fieldbus output slot and then named via a link for use in subsequent tools.
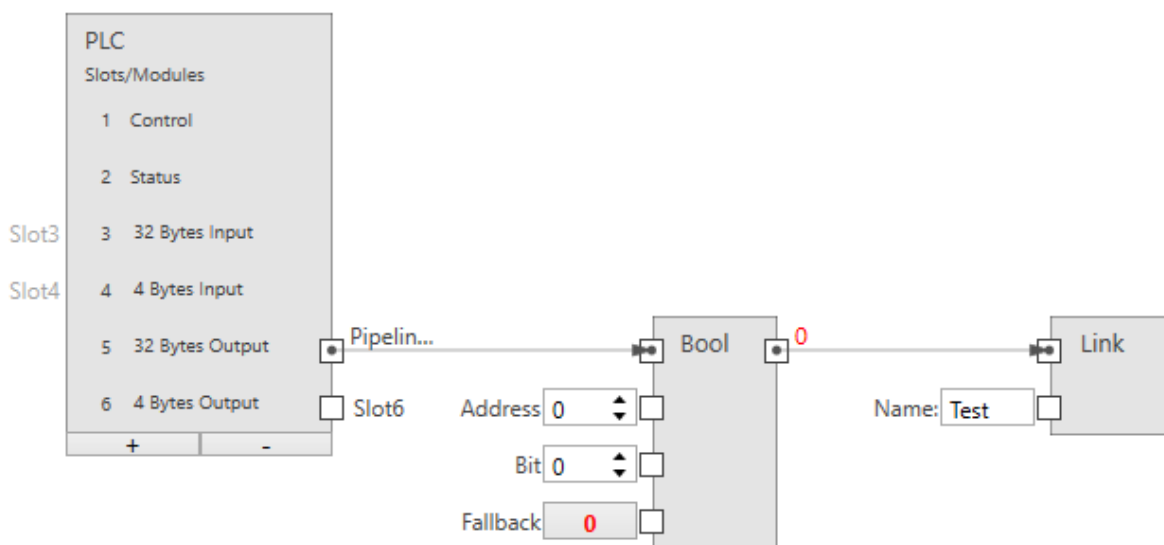
Fig. 162: A boolean input.

The size of a boolean input is 1 byte (but you can allocate up to 8 in one byte).

Inputs

Slot (Type: `ProfinetModule`)

Connects the node with an output module (PLC -> **CS-60/ID-600**).

Address (Type: `UInt32`)

Specifies the byte address in the module.

Bit (Type: `Byte`)

Indicates the bit to be addressed, where 0 is the rightmost bit and 7 is the leftmost bit in the byte.

Fallback (Type: `Boolean`)

Specifies the boolean value that is used, when the fieldbus data is not available. Fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Outputs

Value (Type: `Boolean`)

The boolean value read in.

4.5.25 Read Byte

Reads a byte value from a PLC.
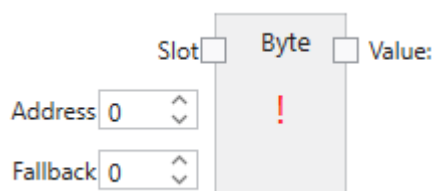
A byte input is connected to a fieldbus output slot and then named via a link for use in subsequent tools.
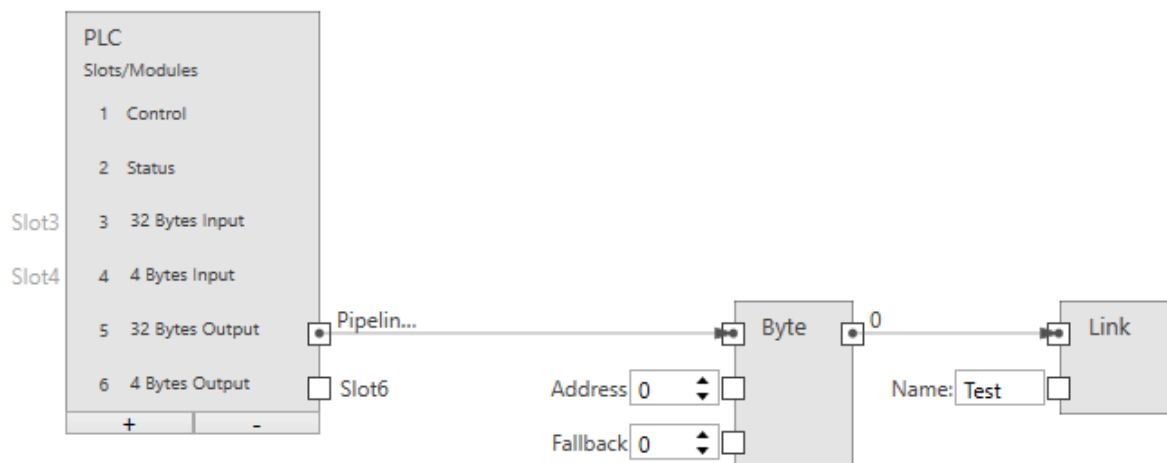
Fig. 163: A byte input.

Inputs

Slot (Type: `ProfinetModule`)

Connects the node with an output module (PLC -> **CS-60/ID-600**).

Address (Type: `UInt32`)

Specifies the byte address in the module.

Fallback (Type: `Byte`)

Specifies the byte value that is used, when the fieldbus data is not available. Fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Outputs

Value (Type: `Byte`)

The byte value read in.

4.5.26 Read Float

Reads in a floating point value from a PLC.
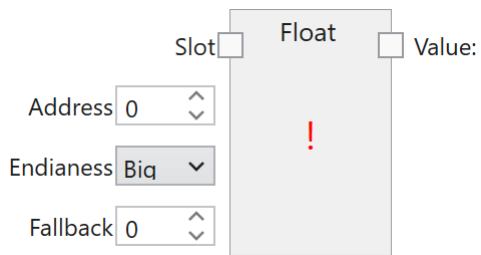
A float input is connected to a fieldbus output slot and then named via a link for use in subsequent tools.
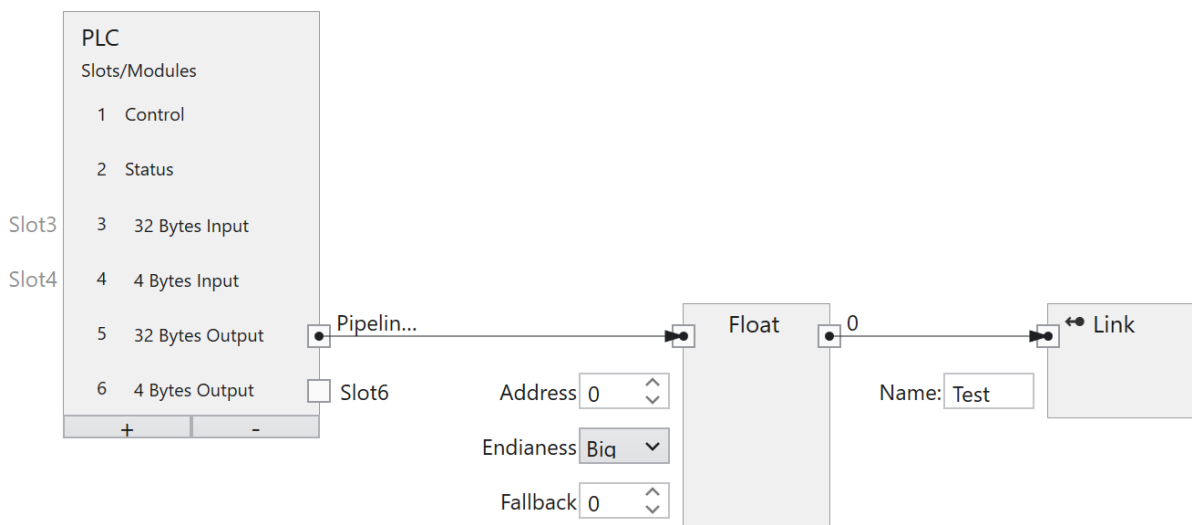
Fig. 164: A floating point input.

The size of a floating point input is 4 bytes.

Inputs

Slot (Type: `ProfinetModule`)

Connects the node with an output module (PLC -> **CS-60/ID-600**).

Address (Type: `UInt32`)

Specifies the byte address in the module.

Endianness (Type: `String`)

Specifies the byte order of the four bytes that comprise the 32 bit float. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback (Type: `Single`)

Specifies the float value that is used, when the fieldbus data is not available. Fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Outputs

Value (Type: `Single`)

The floating point value read in.

4.5.27 Read Int32

Reads an integer value from a PLC.
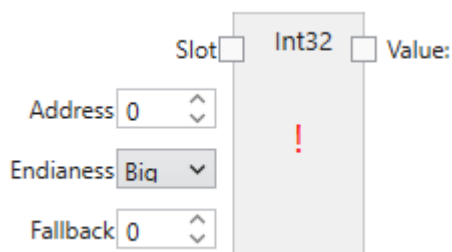
An Int32 input is connected to a fieldbus output slot and then named via a link for use in subsequent tools.
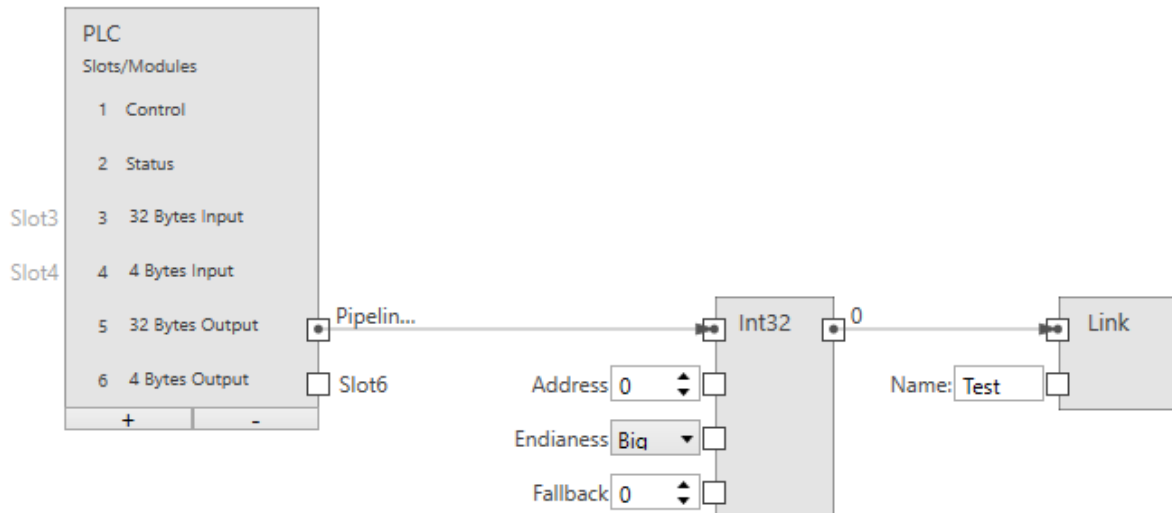
Fig. 165: An integer input.

The size of an integer input is 4 bytes.

Inputs

Slot (Type: `ProfinetModule`)

Connects the node with an output module (PLC -> **CS-60/ID-600**).

Address (Type: `UInt32`)

Specifies the byte address in the module.

Endianness (Type: `String`)

Specifies the byte order of the four bytes that comprise the 32 bit integer. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback (Type: `Int32`)

Specifies the integer value that is used, when the fieldbus data is not available. Fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Outputs

Value (Typ: `Int32`)

The integer Int32 value read in.

4.5.28 PLC

Represents the configuration of a PLC.
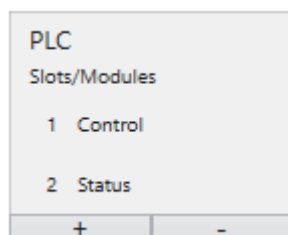
The **PLC** node is used to define modules in specific slots. The **+** and **-** buttons can be used to add or delete modules. This setup must exactly match the setup on the PLC, to enable communication between the PLC and the **CS-60/ID-600**. The PLC node is shown both on the **Acquisition** tool and the **Logic** tool. Inside the PLC node, the names Input and Output for the modules are used as seen from the PLC. A PLC Output is transferred from the PLC to the **CS-60/ID-600**, where it is seen as an input. A PLC Input is transferred from the **CS-60/ID-600** to the PLC, on the **CS-60/ID-600** this is seen as an output.
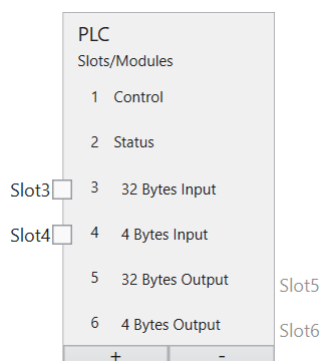
Fig. 166: The PLC node with six modules.

4.5.29 Read String

Reads text from a PLC.
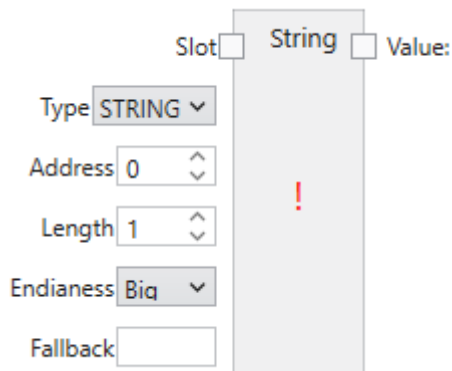
A string input is connected to a fieldbus output slot and then named via a link for use in subsequent tools.
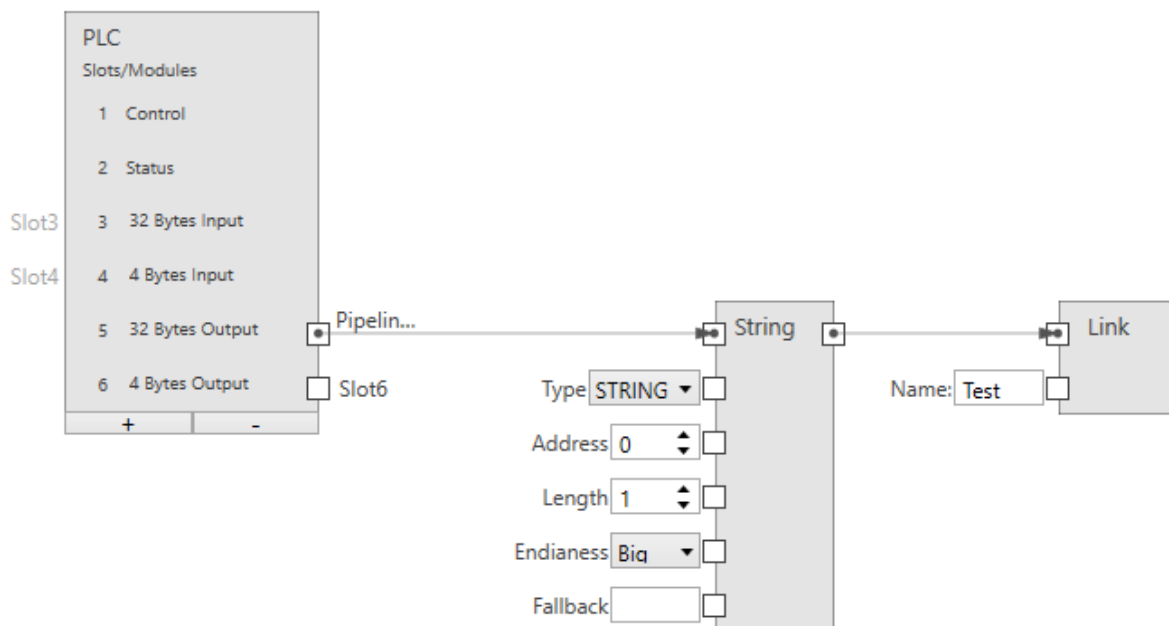
Fig. 167: A text input.

The size of a string input is $(2 + n) * s$, where n is the maximum number of characters and $s = 1$ (for an Ansi STRING) or $s = 2$ (for a Unicode WSTRING). The first byte/word (Ansi/Unicode) contains the maximum number of characters the string can hold. The second byte/word (Ansi/Unicode) contains the actual number of characters the string holds.

Inputs

Slot (Type: `ProfinetModule`)

Connects the node with an output module (PLC -> **CS-60/ID-600**).

Type (Type: `String`)

Specifies if the string is Ansi (STRING) or Unicode (WSTRING).

Address (Type: `UInt32`)

Specifies the byte address in the module.

Length (Type: `UInt32`)

Specifies the number of characters in the string. A length of 1 specifies a single character.

Endianness (Type: `String`)

Specifies the byte order of the two bytes that comprise a 16 bit Unicode character. **Big** endian means that higher valued bytes are stored first, i.e. at lower addresses. **Little** endian is the opposite and means that lower valued bytes are stored first.

Fallback (Type: `Int32`)

Specifies the string value that is used, when the fieldbus data is not available. Fieldbus data may not be available for several reasons: - because PROFINET or EtherNet/IP is disabled on the sensor, - because the job is running in the emulator, etc.

Outputs

Wert (Type: `String`)

The text read in.

4.5.30 Replace

Replaces search text in a text.
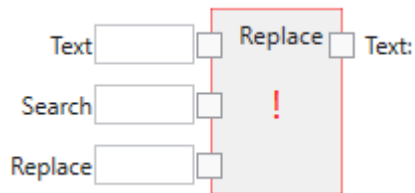
he search text is searched for in a text and replaced by the replacement text.

Inputs

Text (Type: `String`)

The text in which the search text is searched for and the replacement is made.

Search (Type: `String`)

The search text.

Replace (Type: `String`)

The replacement text.

Groß/Kleinschreibung ignorieren (Typ: `Boolean`)

Upper or lower case is ignored if this value is set to True/1.

Outputs

Text (Type: `String`)

The text in which the search text has been replaced by the replacement text.

4.5.31 Split

Separates a text at certain separators.
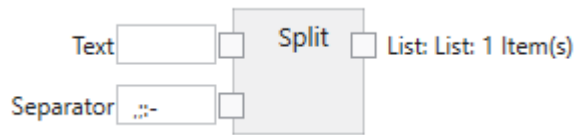
A text is separated at certain separators and converted into a list of texts.

Inputs

Text (Type: `String`)

The text to be split.

Separators (Type: `String`)

Contains the separators. A separation is made at each of the separators.

Outputs

Liste (Type: `List<String>`)

A list of texts.

4.5.32 Text

Assembles a text from one or more inputs.
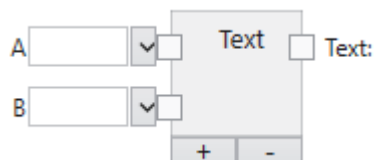
Texts can be entered at the inputs of the Text node, selected from a menu or connected by other nodes.

With the **+** and **-** buttons, inputs can be added at the end or deleted from the end.

Inputs

A...Z (Type: `String`)

The text modules from which an overall text is put together.

There are three ways to specify the text:

1. The text can be entered directly.

2. A macro can be selected from a list.

3. The input can be connected to the output of another node. In this case, the transferred value is automatically converted into a string, if it is not already of the String type.

The following macros are available:

- Acqusition ID

- Job name

- Job slot

- Sensor name

- Sensor serialnumber

- Year

- Month

- Day

- Hour

- Minute

- Second

- Millisecond

- Line feed ()

- Carriage return ()

- Carriage return & Line feed ()

- Tab ()

Not all macros are available in all editors. An entry can be deleted with the **Disconnect** command.

Outputs

Text (Type: `String`)

The assembled text.

4.5.33 Compare

Compares two texts.
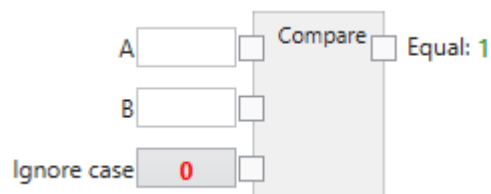
The texts are compared and if they are equal True/1 or if they are not equal False/0 will be returned.

Inputs

A,B (Type: `String`)

Die Texte, die verglichen werden sollen.

Ignore case (Type: `Boolean`)

Upper or lower case is ignored if this value is set to True/1. In this case the two texts 'A' and 'a' are treated as the same. If the value is set to False/0, 'A' and 'a' are not the same.

Outputs

Equal (Typ: `Boolean`)

The result of the comparison: True/1 if the texts are the equal or False/0 if the texts are not equal. Two empty strings are considered equal.

4.5.34 Subtract

Subtracts two numbers.
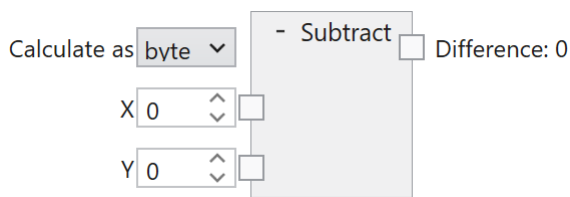
The numbers are subtracted. The subtraction is done with a specified data type.

Inputs

Calculate as (Type: `String`)

The calculation takes place in the specified data type: byte, int or float. Byte allows integer values between 0 and 255, int allows integer values between -2.147.483.648 and 2.147.483.647 and float allows floating point numbers between -3.4 x 10^38 to 3.4 x 10^38.

X (Type: `Double`)

The minuend. All numeric data types are converted to double as needed.

Y (Type: `Double`)

The subtrahend. All numeric data types are converted to double as needed.

Outputs

Difference (Type: `Byte|Int32|Single`)

The difference of the numbers, according to the specified data type. If the result cannot be represented in the specified data type, the node does not produce a numerical result.

4.5.35 Output

Configuration of the TCP/IP outputs.
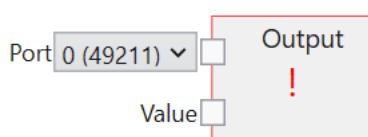
The **Output** node is used to define outputs on a TCP/IP channel.

Inputs

Port (Typ: `String`)

The TCP/IP port used for the output stream. Up to four ports can be used, depending on the number preselected in **Protocol Settings**. The port numbers are predefined from 49211 to 49214.

Value (Typ: `String`)

The output value that is output once on the TCP/IP port after each image evaluation.

Tools can be selected from the tool list to create tool nodes. These provide the values that can be connected to the TCP/IP **Output** node. Here is an example:
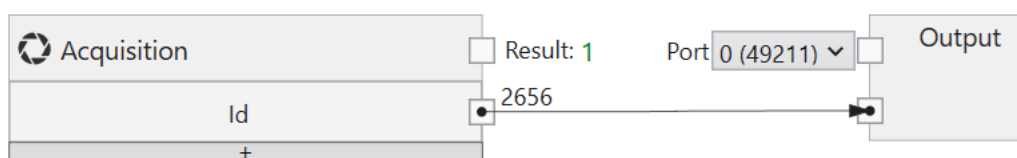
Fig. 168: A tool node for a TCP/IP output stream.

In addition to logical and arithmetic combinations, a combination of texts is also possible with the **Text** node, which can be added to the drawing area by clicking the right mouse button.
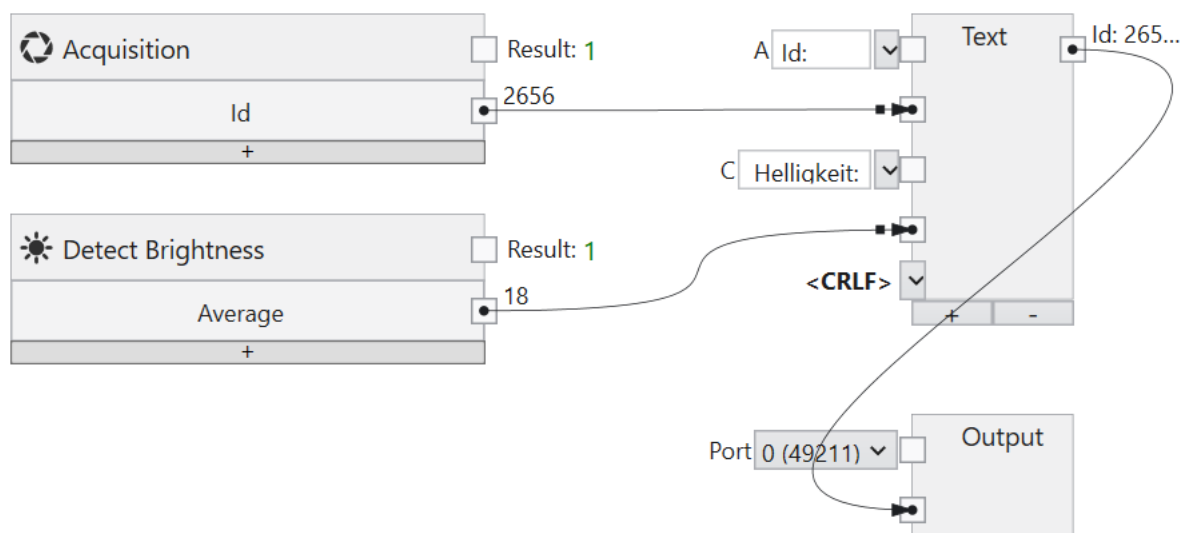
Fig. 169: Text combination for a TCP/IP output stream.

A node on the canvas or a connection between two nodes can be deleted with the **Del** key.

4.5.36 Read Bool

Reads in a boolean value from a TCP/IP input variable.
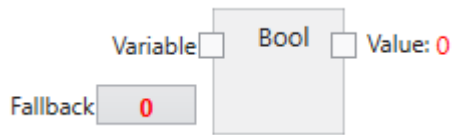
A boolean input is connected to a TCP/IP input variable and then named via a link for use in subsequent tools.
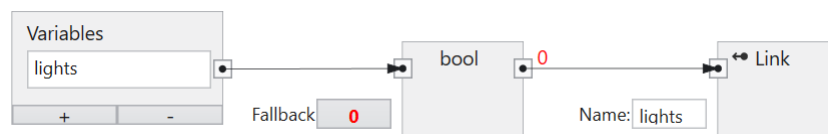
Fig. 170: A boolean input.

Inputs

Variable (Type: `TcpIpVariable`)

Connects the node with a TCP/IP input variable (TCP/IP -> **CS-60/ID-600**).

Fallback (Type: `Boolean`)

Specifies the boolean value that is used, when the TCP/IP data is not available. TCP/IP data may not be available for several reasons: - because the command server on the sensor is deactivated, - because the TCP / IP input variable was not set.

Outputs

Value (Type: `Boolean`)

The boolean value read in via TCP/IP.

4.5.37 Read Byte

Reads a byte value from a TCP/IP input variable.
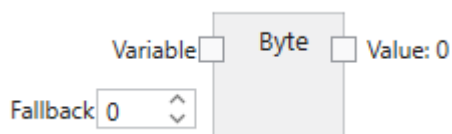
A byte input is connected to a TCP/IP input variable and then named via a link for use in subsequent tools.
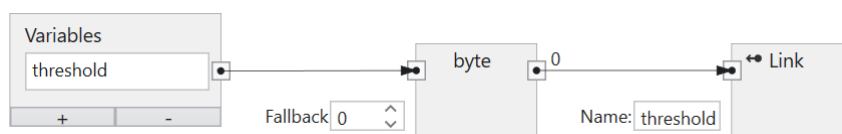
Fig. 171: A byte input.

Inputs

Variable (Type: `TcpIpVariable`)

Connects the node with a TCP/IP input variable (TCP/IP -> **CS-60/ID-600**).

Fallback (Type: `Byte`)

Specifies the byte value that is used, when the TCP/IP data is not available. TCP/IP data may not be available for several reasons: - because the command server on the sensor is deactivated, - because the TCP / IP input variable was not set.

Outputs

Value (Type: `Byte`)

The byte value read in via TCP/IP.

4.5.38 Read Float

Reads in a floating point value from a TCP/IP input variable.
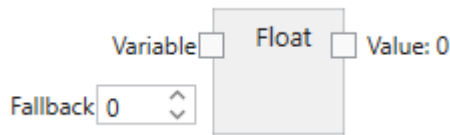
A float input is connected to a TCP/IP input variable and then named via a link for use in subsequent tools.

Fig. 172: A floating point input.

Inputs

Variable (Type: `TcpIpVariable`)

Connects the node with a TCP/IP input variable (TCP/IP -> **CS-60/ID-600**).

Fallback (Type: `Single`)

Specifies the float value that is used, when the TCP/IP data is not available. TCP/IP data may not be available for several reasons: - because the command server on the sensor is deactivated, - because the TCP / IP input variable was not set.

Outputs

Value (Type: `Single`)

The floating point value read in via TCP/IP.

4.5.39 Read Int32

Reads an integer value from a TCP/IP input variable.
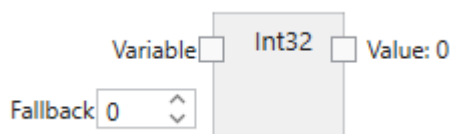
An Int32 input is connected to a TCP/IP input variable and then named via a link for use in subsequent tools.
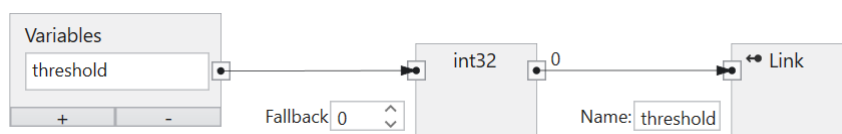
Fig. 173: An integer input.

Inputs

Variable (Type: `TcpIpVariable`)

Connects the node with a TCP/IP input variable (TCP/IP -> **CS-60/ID-600**).

Fallback (Type: `Int32`)

Specifies the integer value to use when no TCP/IP data is available. TCP/IP data may not be available for several reasons: - because the command server on the sensor is deactivated, - because the TCP / IP input variable was not set.

Outputs

Value (Typ: `Int32`)

The integer Int32 value read in via TCP/IP.

4.5.40 Inputs

Represents the configuration of the TCP / IP command inputs.

The **Inputs** node is used to define variables. The **Inputs** node is displayed in the acquisition tool. Variables can be added or deleted using the **+** and **-** buttons.

Many tool parameters can be changed by selecting or entering values. Most parameters can also be **Linked** - e.g connected with TCP/IP inputs.

Linking a value is done in multiple steps the tab **TCP/IP Variables Inputs** of the tool **Acquisition**:

1. A TCP/IP variable is named and connected to a Read node.

2. The output of the read node is connected to a Link node.

3. The Link node defines a name that can be used as a paramter using the button.

The names under points 1 and 3 can be different, but ultimately refer to the same object. Under point 1 the name is defined with which the value of the TCP/IP command `set name = value` is identified. Under point 3 the name is defined under which the value is identified with the links in the job.

Input data is captured whenever an image is captured. This happens at the beginning of an inspection cycle.

Linking TCP/IP Command Variable Inputs

Fig. 174: The context menu on the left.

The **Inputs** node is used to name variables in the TCP/IP namespace.
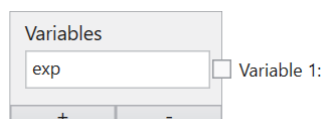
Fig. 175: The inputs node with one variable.

As an example one variable is defined here. The name must begin with a letter and can optionally be followed by one or more letters, numbers or _ (underscore). This name is valid in the TCP/IP namespace, i.e. when used with the TCP/IP command `set name = value`. For example, the command `set exp=100` would assign the value `100` to the variable `exp`.

TCP/IP command variables can provide input values of different types: boolean (Bool), numeric (Byte, Int32, Float) and text (String). In order to use them for linking they must be defined and named. An input is defined by adding a read node of the desired type (by selecting them in the context menu on the left or with a right click on the canvas and selecting **Read Bool**, **Read Byte**, **Read Int32**, **Read Float** or **Read String**) and connecting it to a specific variable defined in the Inputs node. An input is named by adding a **Link** node (with a right click on the canvas and selecting **Link**), connecting a read node with the **Link** node and typing the respective name in the **Link** node.

4.5.41 Variables

Represents the configuration of the TCP/IP command variable outputs.
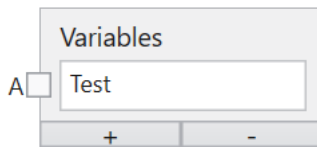
The **Variables** node is used to define variables. The **Variables** node is displayed in the Lgoic tool. Variables can be added or deleted using the **+** and **-** buttons.

Tools can be selected from the tool list to create tool nodes. These provide the values that can be connected to the TCP/IP **Variables** node. Here is an example:
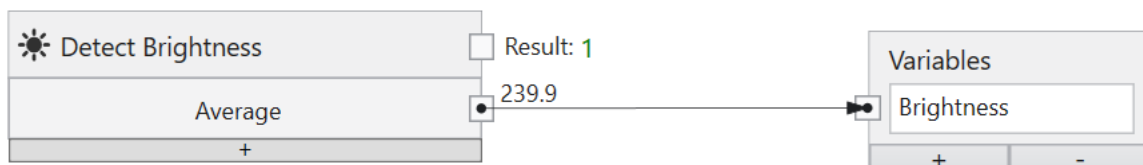
Fig. 176: A tool node for TCP/IP.

The results of several different tools can be combined with logical AND, OR or NOT nodes, which are added to the canvas by clicking with the right mouse button.
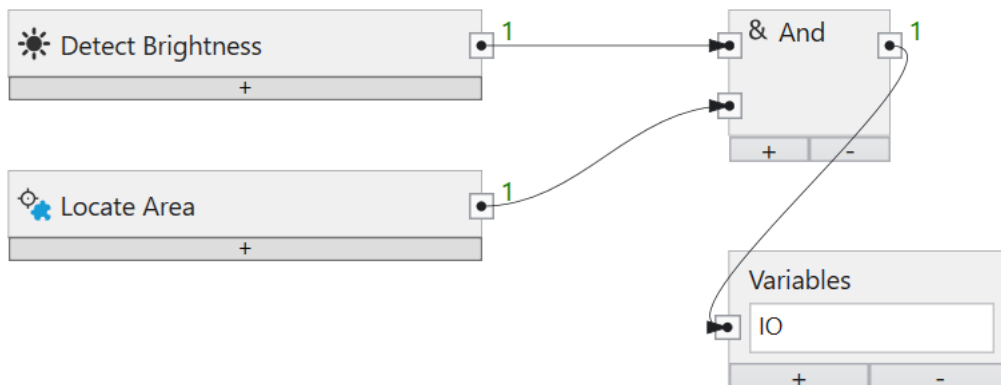
Fig. 177: Logical combination of several tools.

The logic combination can be arbitrary complex.

A node on the canvas or a connection between two nodes can be selected and deleted with the **Del** key.

In addition to logical values, TCP/IP commands variables can also transfer numerical values or texts.

4.5.42 Read String

Reads text from a TCP/IP input variable.
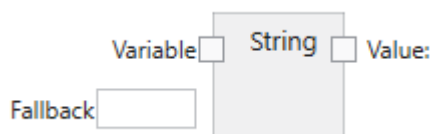
A string input is connected to a TCP/IP input variable and then named via a link for use in subsequent tools.

Fig. 178: A text input.

Inputs

Variable (Type: `TcpIpVariable`)

Connects the node with a TCP/IP input variable (TCP/IP -> **CS-60/ID-600**).

Fallback (Type: `Int32`)

Specifies the string value that is used, when the TCP/IP data is not available. TCP/IP data may not be available for several reasons: - because the command server on the sensor is deactivated, - because the TCP / IP input variable was not set.

Outputs

Wert (Type: `String`)

The text read in via TCP/IP.

4.5.43 Tool Result

Provides results from individual tools.
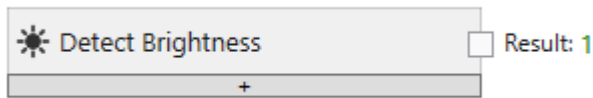
All tools deliver Boolean results.

In addition to these primary results, some tools also provide secondary numerical results, which can be converted into Boolean values via comparisons.
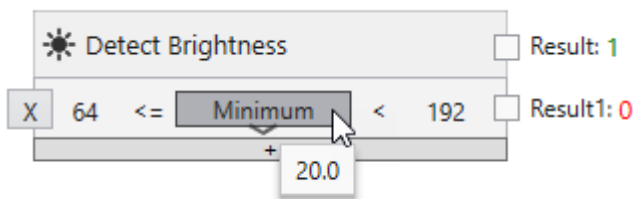
Additional lines can be added using the **+** button and specific lines can be removed using the **X** button.

Here is an example of how the minimum value of the brightness tool is evaluated.

4.5.44 Tool value

Returns values from individual tools.
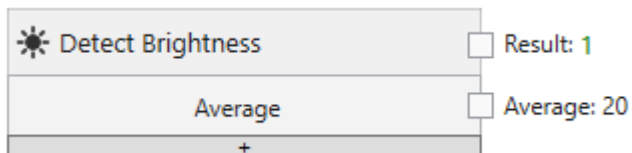
All tools provide primary Boolean results.

In addition to these primary results, some tools also provide secondary numerical or textual results that can be further processed graphically.
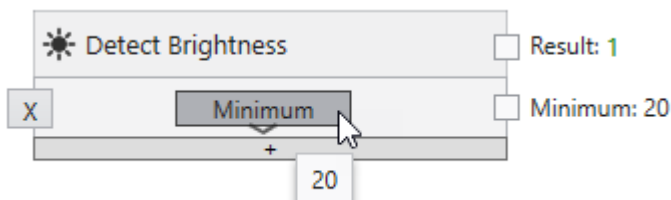
Additional lines can be added using the **+** button and specific lines can be removed using the **X** button.

Here is an example of how the minimum value of the brightness tool is evaluated.

4.5.45 Xor

Logical XOR operation of two boolean values.
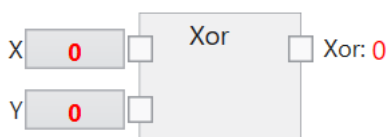
The logical XOR of the boolean input values.

Inputs

X,Y (Type: `Boolean`)

The boolean input values.

Outputs

Xor (Type: `Boolean`)

The XOR operation of the inputs. The result is True/1 if exactly one input is True/1, otherwise the result is False/0.

4.5.46 Within Range

Checks if a value is within a specified range.
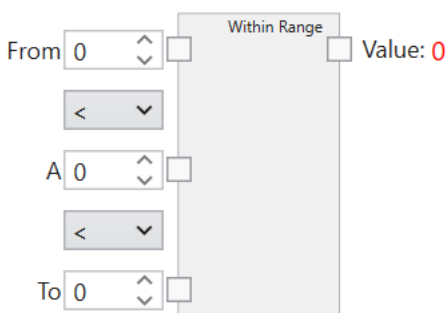
The given numbers are compared within the range defined by two operators. If the operators apply, True/1 is returned or if it does not apply, False/0 is returned.

Inputs

Operators

The comparison operators to span the range. '<' and '<=' are available.

From,A,To (Type: `Double`)

The numbers to be compared. All numeric data types are converted to double as needed.

Outputs

Value (Type: `Boolean`)

The comparison result.

4.6 Protocols

4.6.1 UDP

nVision-i and the **60** vision sensor communicate via UDP (User Datagram Protocol). UDP uses a simple connectionless communication model with a minimum of protocol mechanism.

When **nVision-i** is running, it checks for **CS-60/ID-600** sensors on the network. To do so, it broadcasts identification requests via UDP on port `1069`. These identification requests are sent regularly in intervals of about 4 seconds.

A properly configured **CS-60/ID-600** sensor will see the identification request of **nVision-i** and answer with a UPD message on port `1069` and identify itself with it's name, serial number and IP address.

This mechanism can only work, if your firewall (on the PC) does not inhibit outbound and inbound UDP communication on port `1069`. If in doubt, check with your IT department.

4.6.2 TCP

When a connection has been established using UDP, **nVision-i** and the **CS-60/ID-600** vision sensor communicate via TCP (Transmission Control Protocol). TCP provides communication services between two hosts.

The communication will be carried out on port `50053`.

This mechanism can only work, if your firewall (on the PC) does not inhibit TCP communication on port `50053`. If in doubt, check with your IT department.

4.6.3 SSH

The **nVision-i** software and the **CS-60/ID-600** vision sensor communicate using SSH (Secure Shell). SSH is a cryptographic network protocol for the secure operation of network services over unsecured networks.

Certain functions, such as firmware upgrade, take place via SSH. Port 22 is used for SSH.

This mechanism only works if the firewall (on the PC) does not block connections on port 22. If in doubt, contact your IT department

4.6.4 FTP/SFTP

The **CS-60/ID-600** vision sensor can optionally save images and texts to a server using FTP (File Transfer Protocol) or SFTP (Secure File Transfer Protocol). The FTP option must be activated (see **Protocol settings** of the **CS-60/ID-600**) and the address and port of the FTP server must be defined.

The FTP or SFTP protocol is selected by prefixing the host with `ftp: //` or `sftp: //`.

The user credentials (username and password) must also be provided so that a connection to the FTP/SFTP server can be established. Communication between the built-in optional FTP/SFTP client takes place with FTP via port 21, with SFTP via port 22, the standard ports for FTP and SFTP.
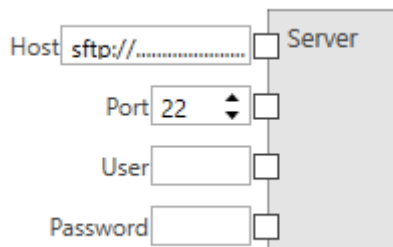
Fig. 179: The FTP/SFTP server settings.

When FTP is enabled, a job can upload images and text files to an FTP/SFTP server. The address and login information of the FTP/SFTP server must be configured in the job on the FTP tab of the logic tool.

Images and texts of inspections can be saved into two directories: **FolderOK** is used for inspections that were Ok, **FolderNOK** is used for failed inspections. If the same name is used for both directories, both types of images are uploaded to the same directory.

The directory and/or file name of the saved files can be specified in a flexible way: add a **Text** node to the function block diagram and type a name of use the dropdown list to select among **Acquisition ID**, **Job name**, **Job slot**, **SensorName**, **Sensor serialnumber**, **Year**, **Month**, **Day**, **Hour**, **Minute**, **Second** and **Millisecond**. Typed names and preselected items can be freely mixed to build arbitrarily complex filenames. If you want to remove one of the predefined items, choose the first item **Disconnect** from the dropdown list. Once the name is built as you

like, connect the output of the **Text** node with the **FolderOK**, **FolderNOK**, **FileNameOK** or **FileNameNOK** inputs of the file node.

If the FTP/SFTP server is not available for uploading, the images and texts are cached on the **CS-60/ID-600** until the maximum number of files has been reached. If the maximum number has been reached, first older OK pictures and then older NOK pictures are overwritten, so that these pictures are lost forever for the FTP/SFTP upload. This behavior gives priority to NOK images.

All files that are in the FTP cache and for some reason could not be uploaded can be saved on the PC with nVision-i.

4.6.5 Telnet

PLCs or PCs or other devices can communicate with a **CS-60/ID-600** vision sensor via a text-based TCP/IP protocol via the Telnet port 23.

This allows commands to be sent to the **CS-60/ID-600** vision sensor and corresponding results to be read back.

Further information can be found in the chapter **TCP/IP Command Server**.

4.6.6 TCP/IP Command Server

The **CS-60/ID-600** sensor can be addressed and controlled by external devices via TCP/IP (Telnet) commands.

In order to address the **CS-60/ID-600** via TCP/IP, the command protocol must be activated. This can be done from the nVision-i start page. (Expand the device tile and click on **Protocol Settings** and **Enable Command Server**). The **CS-60/ID-600** is restarted when the command server is activated or deactivated.

Connection

If the command server is activated, the **CS-60/ID-600** waits for connections on port 23.

A connection can be made, for example, with the Windows 10 terminal with this command:

```
telnet 192.168.3.15
```

As soon as a connection has been established, the **CS-60/ID-600** sends this text:

```
command server
```

From this point on, commands can be sent via the interface. The **CS-60/ID-600** executes these commands and responds accordingly. The API is asynchronous, i.e. the corresponding response from the **CS-60/ID-600** must be awaited before the next command can be sent. This is typically not a problem when entering text via Telnet, but must be taken into account when using a PLC or a control PC.

Commands

ActiveJob

set the active job

$\rightarrow$ `activejob slot<CRLF>`

Makes the job in a specific numeric slot the active job. If the active job could be set, the **CS-60/ID-600** answers with `Ok!`, otherwise the **CS-60/ID-600** answers with `Error!`.

$\leftarrow$ `(Ok!|Error!)<CRLF>`

ActiveJob?

query the active job

$\rightarrow$ `activejob?<CRLF>`

The numeric slot of the active job is returned. If no job is active, the **CS-60/ID-600** answers with `Error!`.

$\leftarrow$ `(slot|Error!)<CRLF>`

Blink

flash the LEDs three times

$\rightarrow$ `blink<CRLF>`

This command serves to locate a **CS-60/ID-600** device visually. This can interfere with an inspection cycle and should therefore not be issued during a running inspection. The **CS-60/ID-600** answers with `Ok!`.

$\leftarrow$ `Ok!<CRLF>`

BootJob

set the bootjob

$\rightarrow$ `bootjob slot<CRLF>`

Makes the job in a specific numeric slot the boot job. If the boot job could be set, the **CS-60/ID-600** answers with `Ok!`, otherwise the **CS-60/ID-600** answers with `Error!`.

$\leftarrow$ `(Ok!|Error!)<CRLF>`

BootJob?

query the boot job

→ `bootjob?<CRLF>`

The numeric slot of the boot job is returned. If no job is set as boot job, the **CS-60/ID-600** answers with `Error!`.

← `(slot|Error!)<CRLF>`

Clock

set the sensor clock

→ `clock datetime<CRLF>`

Sets the date and time of the sensor clock. The date and time should be provided in the following form: `1/15/2021 8:30:00 AM`. If the clock could be set, the **CS-60/ID-600** answers with `Ok!`, otherwise the **CS-60/ID-600** answers with `Error!`.

← `(Ok!|Error!)<CRLF>`

Clock?

retrieve the sensor clock

→ `clock?<CRLF>`

Retrieves the date and time from the sensor clock.

← `1/1/2021 8:00:00 AM<CRLF>`

CloseInspectionGate

close the gate for continuous inspection

→ `closeinspectiongate<CRLF>`

The acquisition mode in the job must be set to gated, otherwise this command will be ignored. If the gate could be closed the **CS-60/ID-600** answers with `Ok!`, otherwise the **CS-60/ID-600** answers with `Error!`.

← `(Ok!|Error!)<CRLF>`

Cls

clear the screen

→ `cls<CRLF>`

Clears the screen by sending VT100 terminal escape sequences: `Esc[1J` clear the screen from cursor up. `Esc[H` move cursor to upper left corner.

← `Esc[1JEsc[H`

Exit

exit the session

→ `exit<CRLF>`

Logs out of the command server and closes the connection.

← `Ok!<CRLF>`

Get

get the value of a symbol

→ `get[?] name<CRLF>`

Get the value of the symbol name. The question mark is optional. The symbolname must start with an uppercase or lowercase letter [A-Za-z] and can be followed by uppercase or lowercase letters, numbers or an underscore. The associated value (potentially multiple lines) is returned if the symbol exists, otherwise `Error!` is returned.

← `(...value...|Error!)<CRLF>`

Help

show command help

→ `help[?][ command]<CRLF>`

The question mark and the command are optional. If no command (or a wrong command) is given, help shows a list of commands. If a command is given, help shows the specific help of the command.

← `...multiline text... ← Ok!<CRLF>`

Jobs?

list the jobs

→ `jobs[?]<CRLF>`

The question mark is optional. The jobs are listed with their slot followed by their name, separated by a space. If no jobs are loaded, the **CS-60/ID-600** answers with `Error!`

← `...multiline job list... ← Ok!<CRLF>`

or

← `Error!<CRLF>`

License?

return the license

→ `license[?]<CRLF>`

The question mark is optional. Returns the license. Possible license strings are `Basic`, `Basic & Measure`, `Basic & Barcode` or `Basic, Measure & Barcode`.

← `...license description...<CRLF>`

MacAddress?

return the MAC address

→ `macadress[?]<CRLF>`

The question mark is optional. Returns the MAC address. A MAC address looks like this `8C:0F:A0:00:10:69`.

← `...MAC address...<CRLF>`

Model?

return the model

→ `model[?]<CRLF>`

The question mark is optional. Returns the model description..

← `...model description...<CRLF>`

Name?

return the **CS-60/ID-600** name

→ `name[?]<CRLF>`

The question mark is optional. Returns the name.

← `...name...<CRLF>`

Offline

set the **CS-60/ID-600** offline

→ `offline<CRLF>`

Tries to set the **CS-60/ID-600** offline. If the **CS-60/ID-600** could go offline, the response is `Ok!`, otherwise it is `Error!`. It may take some time to put the **CS-60/ID-600** offline, since any current execution cycle must be terminated.

← `(Ok!|Error!)<CRLF>`

Online

set the **CS-60/ID-600** online

→ `online<CRLF>`

Tries to set the **CS-60/ID-600** online. If the **CS-60/ID-600** could to online, the response is `Ok!`, otherwise it is `Error!`. It may take some time to put the **CS-60/ID-600** online. If no job is loaded onto the **CS-60/ID-600**, the **CS-60/ID-600** cannot go online.

← `(Ok!|Error!)<CRLF>`

Online?

return the online status

→ `online?<CRLF>`

Queries the online status. The response is `Ok!` if the is online, and `Error!` if the **CS-60/ID-600** is offline.

← `(Ok!|Error!)<CRLF>`

OpenInspectionGate

open the gate for continuous inspection

$\rightarrow$ `openinspectiongate<CRLF>`

The acquisition mode in the job must be set to gated, otherwise this command will be ignored. If the gate could be opened the **CS-60/ID-600** answers with `Ok!`, otherwise the **CS-60/ID-600** answers with `Error!`.

$\leftarrow$ `(Ok!|Error!)<CRLF>`

Prompt

set the interactive prompt

$\rightarrow$ `prompt[ ...characters...]<CRLF>`

Sets the interactive prompt. An interactive prompt (for example " > ") is very useful in interactive terminal sessions. In PLC sessions the prompt might be a problem, thus it is set to an empty string by default.

$\leftarrow$ `Ok!<CRLF>` $\leftarrow$ `...characters...<CRLF>`

Push

push the value of a variable after each inspection

$\rightarrow$ `push[ variable]<CRLF>`

After each inspection the value of the given variable is pushed, push without variable removes the request.

$\leftarrow$ `Ok!<CRLF>`

Ready?

query the ready status

$\rightarrow$ `ready?<CRLF>`

The question mark is optional. The ready status of the **CS-60/ID-600** is retured. The **CS-60/ID-600** is ready when it has finished an inspection cycle. The **CS-60/ID-600** is not ready when it is in the middle of an inspection cycle. If the **CS-60/ID-600** is offline, it also is not ready. The **CS-60/ID-600** answers with `Ok!` if it is ready, `Error!` otherwise.

$\leftarrow$ `(Ok!|Error!)<CRLF>`

Resolution?

return the camera resolution

$\rightarrow$ `resolution[?]<CRLF>`

The question mark is optional. Returns the **CS-60/ID-600** camera resolution.

$\leftarrow$ `...resolution...<CRLF>`

Results?

list the last inspection results

$\rightarrow$ `results[?]<CRLF>`

The question mark is optional. Returns the results of the last inspection. If no results are available, the **CS-60/ID-600** answers with `Error!`.

$\leftarrow$ `...multiline results...` $\leftarrow$ `Ok!<CRLF>`

or

$\leftarrow$ `Error!<CRLF>`

Serial?

return the serial number

$\rightarrow$ `serial[?]<CRLF>`

The question mark is optional. Returns the **CS-60/ID-600** serial number. A serial number looks like i.e. `10030`.

$\leftarrow$ `...serial number...<CRLF>`

Set

set the value of a symbol

$\rightarrow$ `set name=value<CRLF>`

Set the symbol `name` to `value`. The symbol name must start with an uppercase or lowercase letter [A-Za-z] and can be followed by uppercase or lowercase letters, numbers or an underscore. The **CS-60/ID-600** answers with `Ok!` if it could set the value, `Error!` otherwise.

$\leftarrow$ `(Ok!|Error!)<CRLF>`

Trigger

acquire an image

→ trigger<CRLF>

Starts the image acquisition and inspection. The response is Ok! if the the **CS-60/ID-600** accepted the trigger, and Error! if the **CS-60/ID-600** was not ready and did not accept the trigger. The trigger mode in the acquisiton tool must be set to Virtual, otherwise the trigger is not accepted and the answer will be Error!.

← (Ok!|Error!)<CRLF>

Variables?

list the job variables

→ variables[?]<CRLF>

The question mark is optional. All job variables are listed together with their actual values.

← ...multiline variables... ← Ok!<CRLF>

or

← Error!<CRLF>

Version?

return the firmware version

→ version[?]<CRLF>

The question mark is optional. Returns the **CS-60/ID-600** firmware version. A firmware version looks like 20.4.1205, where 20 is the year, 4 is an ordinal number within the year, and 1205 is an overall numeric build number.

← ...firmware version...<CRLF>

Examples

Job Change

A job change is executed by taking the **CS-60/ID-600** offline, setting the active job to a new slot, and taking the **CS-60/ID-600** back online.

→ offline<CRLF> ← Ok!<CRLF> → activejob 2<CRLF> ← Ok!<CRLF> → online<CRLF> ← Ok!<CRLF>

Image Acquisition

In order to acquire an image, you should check the `ready?` flag before you issue a trigger.

$\rightarrow$ `ready?<CRLF>` $\leftarrow$ `Ok!<CRLF>` $\rightarrow$ `trigger<CRLF>` $\leftarrow$ `Ok!<CRLF>`

4.6.7 PROFINET

The **CS-60/ID-600** sensor may be used as a PROFINET device in a PROFINET environment.

In order to use the **CS-60/ID-600** as a PROFINET device, the PROFINET communication protocol must be enabled. This can be done from the nVision-i start page. (Extend the device tile, click on **Protocol Settings** and **Enable Profinet**). The **CS-60/ID-600** will reboot when PROFINET will be enabled or disabled. When PROFINET is enabled, the **CS-60/ID-600** will start in offline mode.

The PROFINET Gsd file is stored in the subdirectory `Profinet` of the nVision-i installation directory (`%PROGRAM_FILES%\di-soric\nVision-i\`).

PROFINET Identity

A PROFINET device is uniquely identified by a **Vendor Id** and a **Device Id**. The di-soric Vendor Id is 0x0221. The Device Id is 0x0001.

Connection

A PROFINET master and a PROFINET device exchange data with a cyclic messaging protocol. The Vision Sensor cycle update time is 8 ms.

The data block that is exchanged is maximal 1024 bytes for both input and output. The actual sizes of the exchanged blocks are controlled by the PLC setup. The PLC can put input and output modules of varying size (1 Byte, 2 Bytes, etc.) into up to 256 slots. The actual slot setup determines how many bytes are exchanged between the PLC and the **CS-60/ID-600**.

Prepare the PLC

Integrate the **CS-60/ID-600** into the PLC environment with the TIA Portal software. Locate the **CS-60/ID-600** GSD file (`GSDML-V2.42-di-soric-CS-60/ID-600-...`) in the `Profinet` subdirectory of the nVision-i installation directory (`%PROGRAM_FILES%\di-soric\nVision-i\`) and use the command `Options -> Manage general station description files (GSD)` to add the GSD file to the PLC.
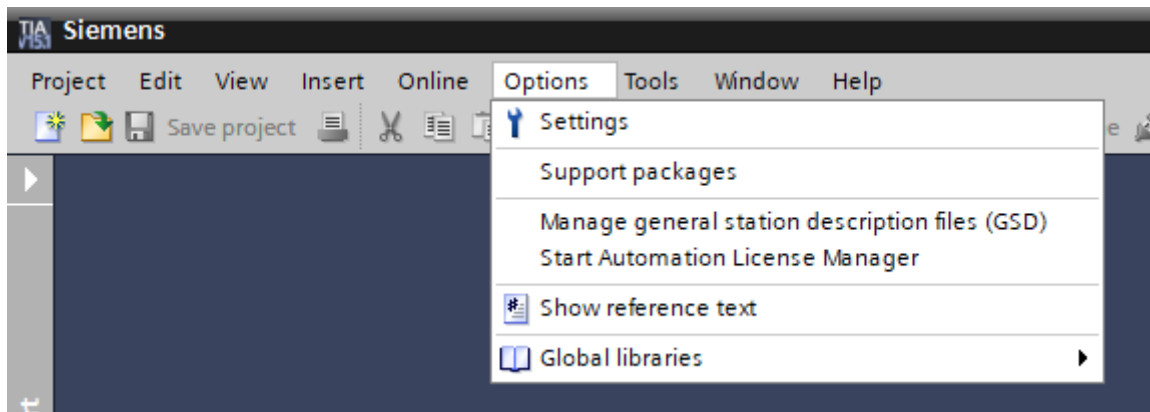
Fig. 180: Find the GSD file of the **CS-60/ID-600**
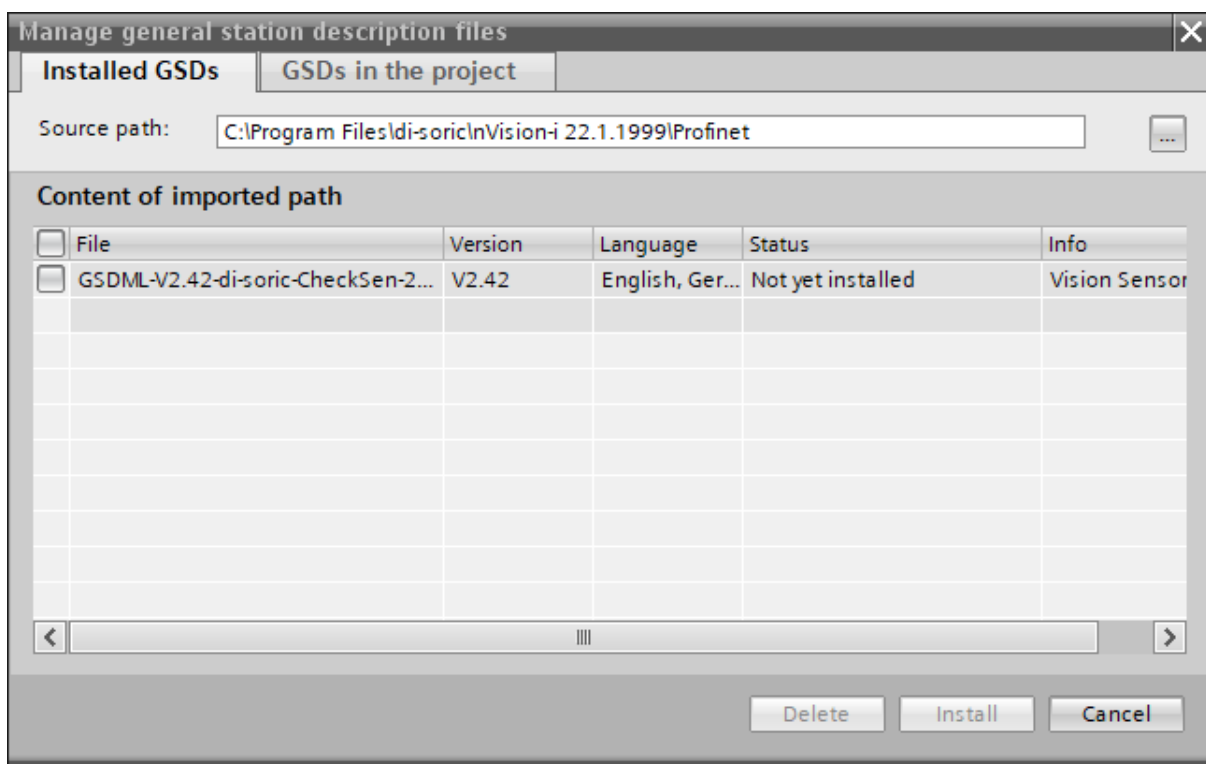
Fig. 181: Add the GSD file of the **CS-60/ID-600** to TIA

After you have added the PLC to the TIA project, add the **CS-60/ID-600** to the project. The **CS-60/ID-600** is listed in the hardware catalog, under `Other field devices -> PROFINET IO -> General -> di-soric -> Vision Sensor -> CS-60/ID-600`.
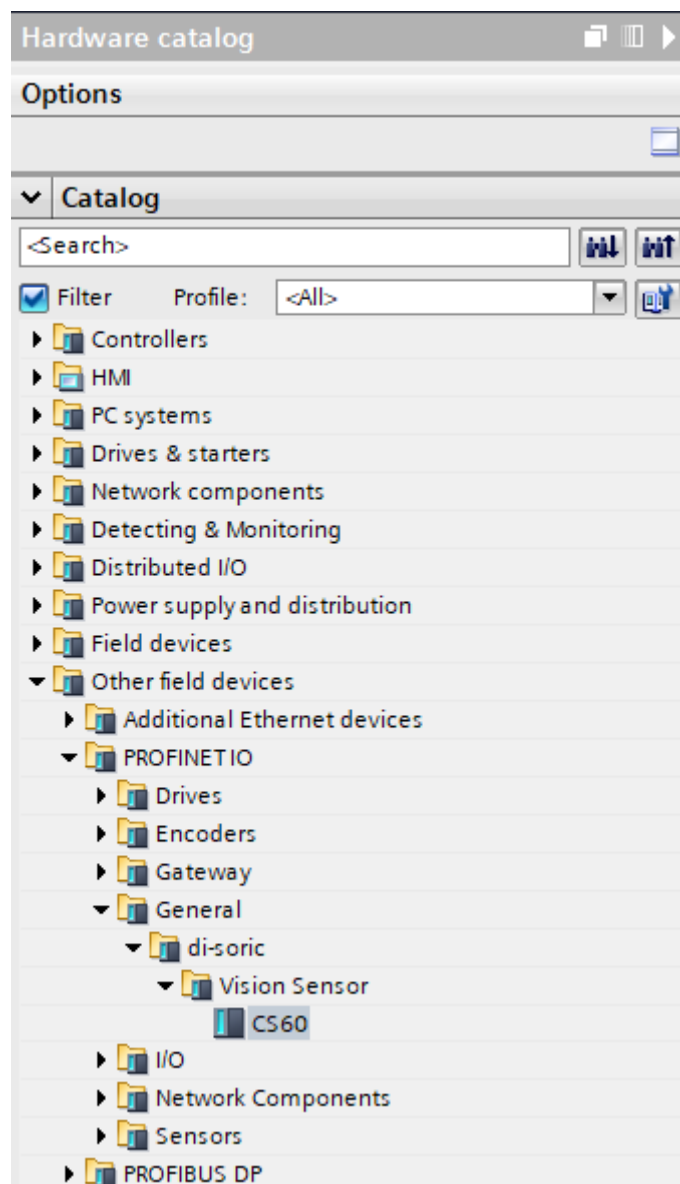
Fig. 182: The TIA hardware catalog

After the **CS-60/ID-600** device has been added, it must be connected to the PLC.
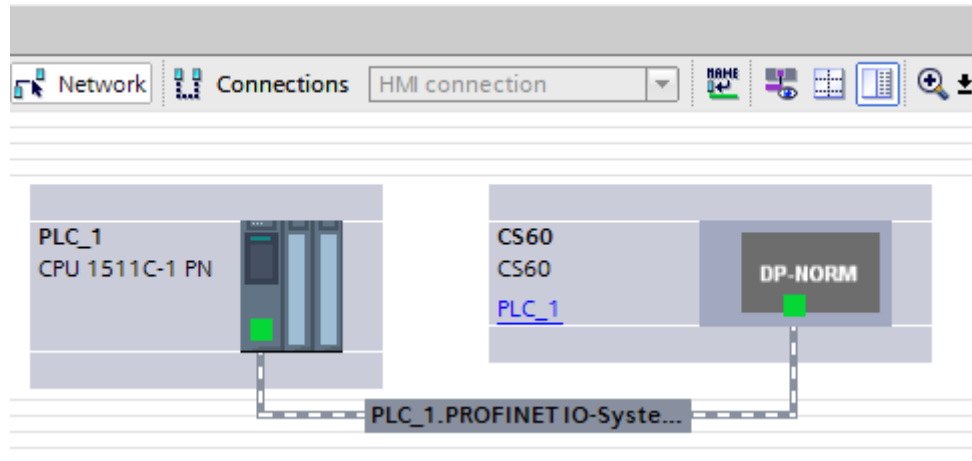
Fig. 183: Connect the PLC to the **CS-60/ID-600**

Then, you need to configure the Input and Output Modules to the slots of the **CS-60/ID-600**. Here is an example:
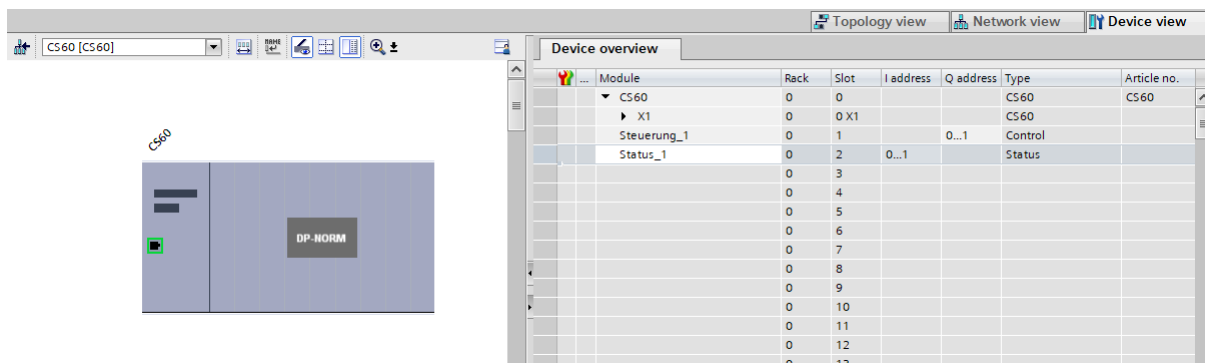
Fig. 184: Add modules to **CS-60/ID-600** slots

At least 2 byte inputs and outputs must be configured for the control and status signals.

This setup must be finally compiled and downloaded to the PLC.

EtherNet/IP

The **CS-60/ID-600** sensor can be used as an EtherNet/IP device in an EtherNet/IP environment.

To use the **CS-60/ID-600** as an EtherNet/IP device, the EtherNet/IP communication protocol must be activated. This can be done from the nVision-i start page. (Expand the device tile and click on **Protocol Settings** and **Enable EtherNet/IP**). The **CS-60/ID-600** is restarted when EtherNet/IP is activated or deactivated. If EtherNet/IP is activated, the **CS-60/ID-600** starts in offline mode.

The EtherNet/IP EDS file is located in the `EthernetIP` subdirectory in the nVision-i installation directory (`%PROGRAMME%\di-soric\nVision-i\`).

EtherNet/IP Identity

An EtherNet/IP device is uniquely identified by a **Vendor Id** and a **Product Code**. The di-soric Vendor Id is 1663 and the product code is 1 (CS60) or 2 (ID600).

Connection

An EtherNet/IP Scanner and an EtherNet/IP Adapter exchange data in a cyclic message protocol.

The exchanged data block is a maximum of 400 bytes for both input and output.

Preparation of the PLC (Allen-Bradley CompactLogix 5380)

Integrate the **CS-60/ID-600** into the PLC environment with the Logix Designer Studio 5000. Locate the **CS-60/ID-600** EDS file in the `EthernetIP` subdirectory of the nVision-i installation directory (`%PROGRAMME%\di-soric\nVision-i\`) and use the command `Tools -> EDS Hardware Installation Tool` to register the EDS file in the PLC.
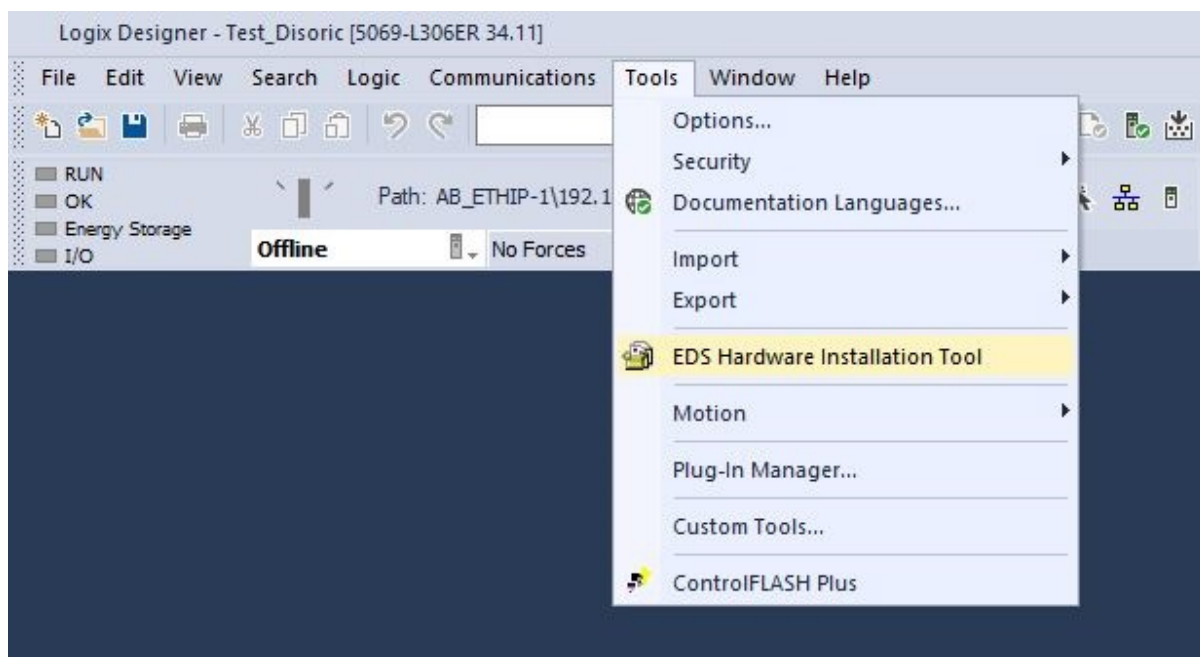
Fig. 185: Find the EDS file of the **CS-60/ID-600**
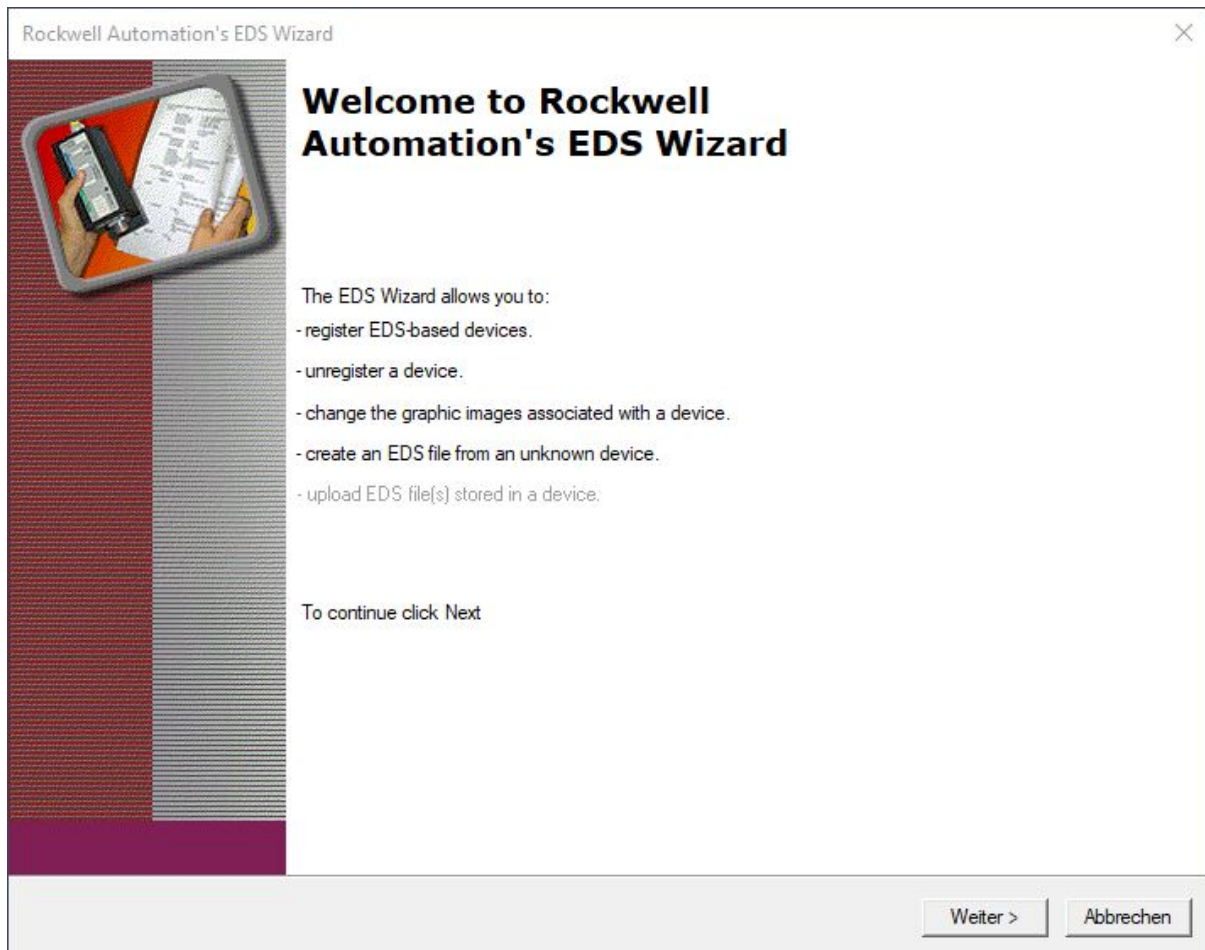
Fig. 186: Add the EDS file of the **CS-60/ID-600** to Studio 5000

After you have created a new project, also add the **CS-60/ID-600** to the project. The **CS-60/ID-600** can be found in the catalogue under CS-60/ID-600.
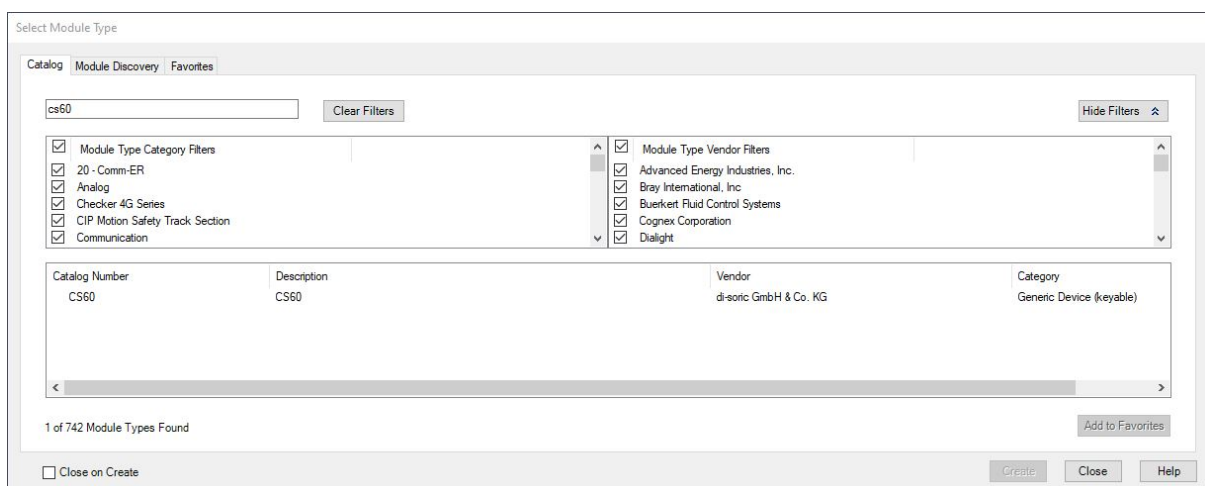
Fig. 187: Add the CS-60/ID-600 to the project

After the **CS-60/ID-600** has been added, it must be configured with a name CS-60/ID-600 and IP address.
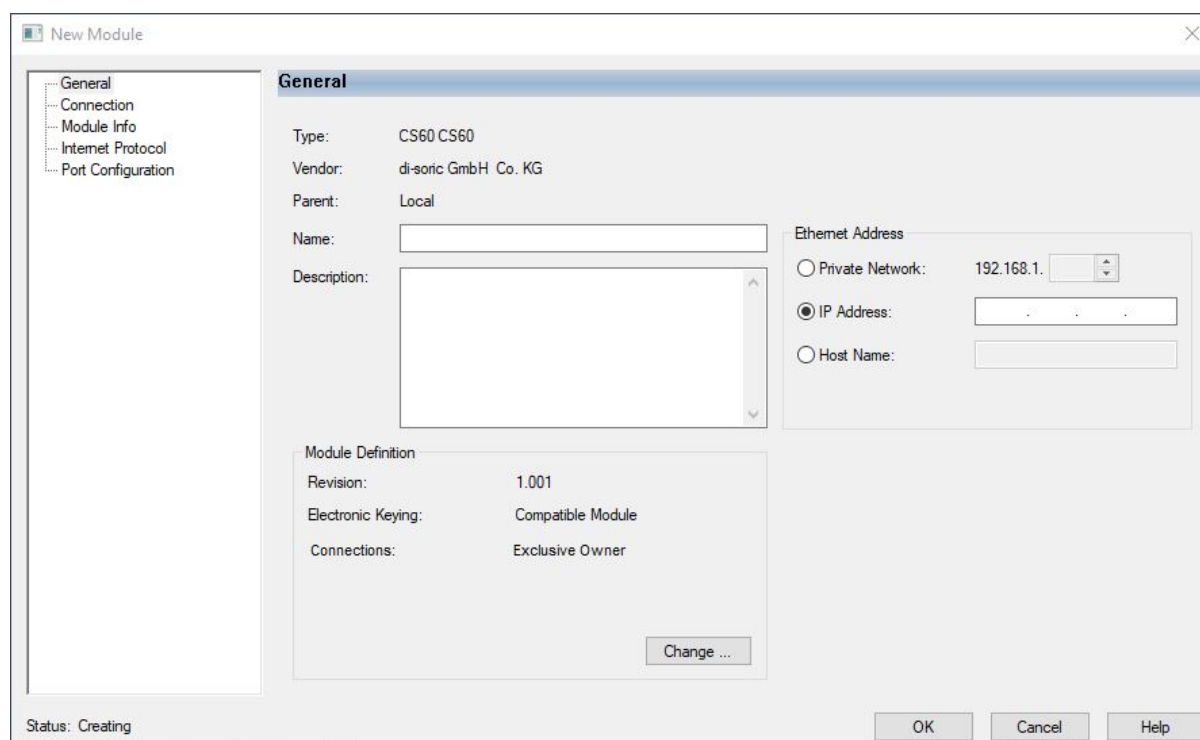
Fig. 188: Configuration of **CS-60/ID-600**

4.6.8 Standard Status and Control Signal Setup for PROFINET and EtherNet/IP

The **CS-60/ID-600** status information is sent from the **CS-60/ID-600** to the PLC. By default, the assignment is as follows:

Byte Offset	Bits	Data	Description
0	0	Online	True if the **CS-60/ID-600** is online, false if the **CS-60/ID-600** is offline.
0	1	AckSe-tOnline	Acknowledge toggles of the `SetOnline` control bit.
0	2		
0	3	Ack-JobLoad	Acknowledge toggles of the `JobLoad` control bit.
0	4	JobLoadOk	True if a job could be loaded, false if loading failed.
0	5	Ready	True if the **CS-60/ID-600** can accept a trigger, false if it is busy.
0	6	AckTrig-ger	Acknowledge Trigger. True after rising edge, false after falling edge of Trigger.
1	0-7	JobId	The id of the current job (0 .. 255).

Control information is sent from the PLC to the **CS-60/ID-600**. By default, the assignment is as follows:

Byte Off-set	Bits	Data	Description
0	0	SetOn-line	Set to true to put the **CS-60/ID-600** online, false to put it offline.
0	1	JobLoad	Execute a job load command.
0	2	BootJob	True to make the loaded job the boot job, false otherwise.
0	3	Trigger	Trigger an acquisition.
1	0-7	JobLoa-dId	The id of the job that should be loaded (0 .. 255).

The Vision Sensor starts in offline mode, when PROFINET is enabled.

Online Status

The online of offline status of the Vision Sensor is reflected in the **Online** bit (status byte 0, bit 0). If this bit is set, the Vision Sensor is online, otherwise it is offline.

In online mode a job is executing.

In offline mode, no job is executing, but the Vision Sensor is listening for commands such as changing a job.

To change from offline to online or vice versa, you set or clear the **SetOnline** bit (control byte 0, bit 0).
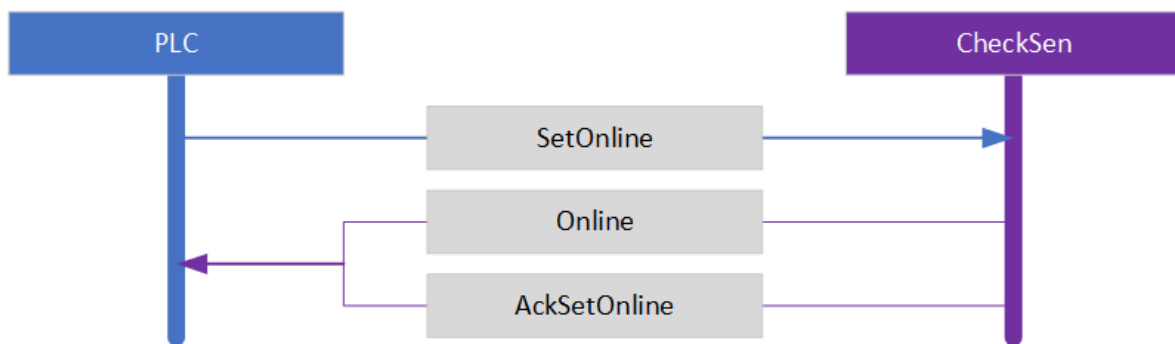
Fig. 189: The SetOnline sequence

Set Online

When the Vision Sensor is offline, it can be put into online mode. To put the Vision Sensor online, set the **SetOnline** bit (control byte 0 bit 0).

The Vision Sensor can only go online, if a job has been loaded first.

To check if the Vision Sensor could go online, wait until the **AckSetOnline** bit (status byte 0, bit 1) is set and check the **Online** bit (status byte 0, bit 0).
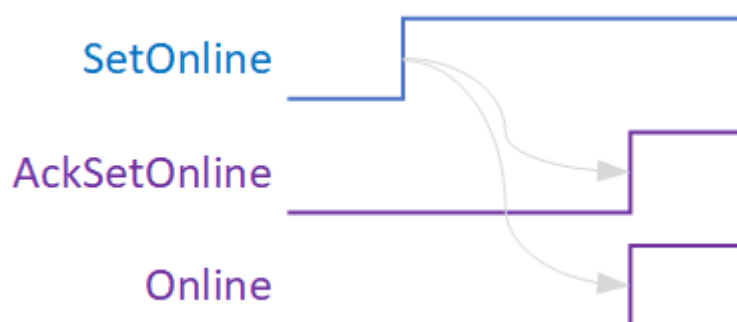
Fig. 190: The SetOnline timing

Set Offline

When the Vision Sensor is online, it can be put into offline mode. To put the Vision Sensor offline, clear the **SetOnline** bit (control byte 0 bit 0).

In order to go offline, the current cycle of the executing job is finished, and the execution of the job is stopped. The job remains loaded. Then, the Vision Sensor goes offline.

To check when the Vision Sensor goes offline, wait until the **AckSetOnline** bit (status byte 0, bit 1) is cleared. The **Online** bit (status byte 0, bit 0) will be cleared as well.
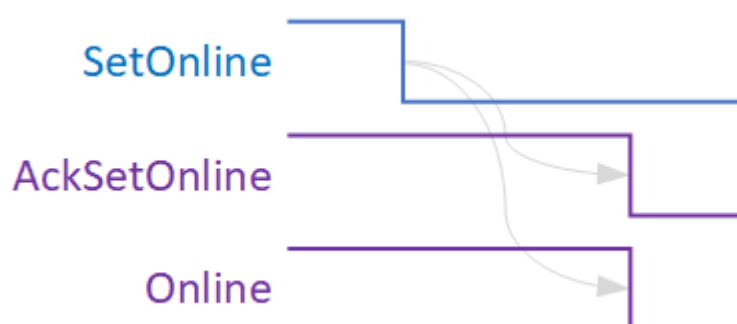
Fig. 191: The SetOffline timing

Job Load

In order to change a job via Profinet, the Vision Sensor must be put offline, the **JobLoadId** number (control byte 1, bits 0..7) must be set to the desired job number and the **JobLoad** bit (control byte 0, bit 1) must be set. Set the **BootJob** bit (control byte 0, bit 2) if you want to make the new job the boot job, otherwise clear the **BootJob** bit. When the loading is finished, the **AckJobLoad** bit (status byte 0 bit 3) is set. The **JobLoadOk** bit (status byte 0 bit 4) is set if the job could be loaded and cleared if loading the job failed. In addition, the **JobId** number (status byte 1, bits 0..7) is set to reflect the new job id. The PLC should then clear the **JobLoad** bit (control byte 0, bit 1) and put the Vision Sensor online.
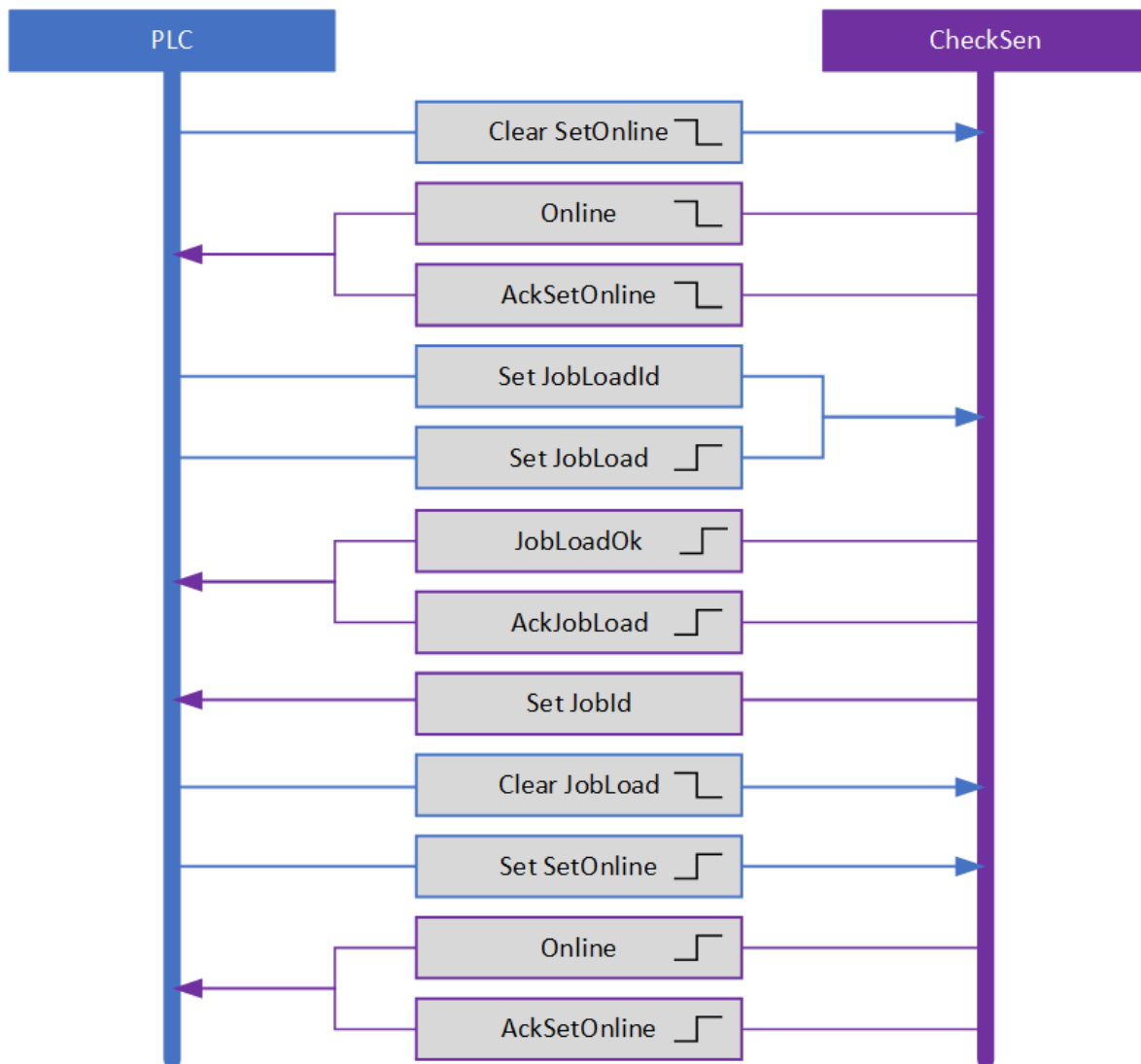
Fig. 192: The JobLoad sequence

Trigger

A **Trigger** (control byte 0, bit 3) starts the image acquisition via PROFINET. Profinet triggers are only executed if - **Ready** (status byte 0, bit 5) is set, and - the trigger type in the acquisition tool is set to PROFINET.

C4G Mode Status and Control Signal Setup

The C4G mode is a Profinet based communication mode that is similar to the communication style of the di-soric C4G sensors. In order to work in C4G mode, you must delete the **CS-60/ID-600** modules (Control, Status) and provide the following 8 modules in the respective slots with their exact sizes:

Slot	Type	Description
1	2 Bytes Output	Device control - controls the **CS-60/ID-600** in C4G mode.
2	2 Bytes Input	Device status - indicates the **CS-60/ID-600** status in C4G mode.
3	1 Byte Output	Acquisition control - controls the image acquisition in C4G mode.
4	3 Bytes Input	Acquisition status - indicates the current acquisition status in C4G mode.
5	1 Byte Output	Results control - controls the processing of result data in C4G mode.
6	3 Bytes Input	Results status - indicates the results tatus status in C4G mode.
7	3 Bytes Output	Input data.
8	3 Bytes Input	Output data.

Additional modules can be freely added.

The definition in the Acquisition and Logic tools must match exactly.
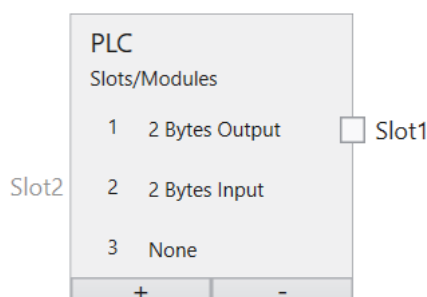
Fig. 193: C4G mode

C4G Device Control

This module consists of data that is sent from the PLC to the **CS-60/ID-600** sensor.

Bit	Name	Description
0	Set Offline	Setting this bit puts the **CS-60/ID-600** in offline mode.
1	Job Change	This bit is used for changing jobs (job number must be filled as well).
2	Retrain	Not supported.
3	Lights Off	Setting this bit to 1 turns off external illumination.
4	Volatile	Not supported.
8-15	Job Number	see Job Change bit.

C4G Device Status

This module consists of data that is sent from the **CS-60/ID-600** sensor to the PLC.

Bit	Name	Description
0	Online	If the **CS-60/ID-600** is online, this bit is 1.
1	Offline Reason	Not supported.
2	Offline Reason	Not supported.
3	Offline Reason	Not supported.
1	General Fault	Always 0.
5	Job Load Complete	If the job change is successfull, this changes to 1.
6	Job Load Failed	If the job change is not successfull, this changes to 1.
7	Retrain Completed	Not supported.
8	Retrain Failed	Not supported.
9	Observer	Not supported.

C4G Acquisition Control

This module consists of data sent from the PLC to the **CS-60/ID-600** sensor.

Bit	Name	Description
0	Trigger	Trigger an acquisition.
1	Trigger Enable	Not supported.

C4G Acquisition Status

This module consists of data that is sent from the **CS-60/ID-600** sensor to the PLC.

Bit	Name	Description
0	Trigger Ready	Indicates when the device is ready to accept a trigger.
1	Trigger Ack	Indicates that the **CS-60/ID-600** has received a trigger.
2	Acquiring	Indicates that the **CS-60/ID-600** is acquiring an image.
3	Missed Ack	Not supported.
8-23	Acquisition ID	Id of the next trigger to be used.

The Acquisition ID in C4G mode is limited to 16 bit.

C4G Results Control

This module consists of data sent from the PLC to the **CS-60/ID-600** sensor.

Bit	Name	Description
0	Results Enable	Enable queuing of result data.
1	Results Ack	Acknowledge that the PLC has read the latest results.

C4G Results Status

This module consists of data that is sent from the **CS-60/ID-600** sensor to the PLC.

Bit	Name	Description
0	Part Detect	Not supported.
1	Inspecting	Its value is 1 when the **CS-60/ID-600** is inspecting.
2	Inspection Complete Toggle	Changes when the inspection is finished.
3	Result Buffer Overrun	Not supported.
4	Result Available	Its value changes to 1 when the inspection is finished.
5	Any Fail	Its value is 1 if at least one of the tools fails.
6	All Pass	Its value is 1 if all the tools pass.
7-22	Result ID	Pair of the Acquisition ID.

The handshake to retrieve data is supported, but only the result from the last inspection is available (no queuing).

C4G Input

This module consists of data sent from the PLC to the **CS-60/ID-600** sensor.

Standard linking can be used here.

C4G Output

This module consists of data that is sent from the **CS-60/ID-600** sensor to the PLC.

Standard linking can be used here. Some signals that are available on the C4G sensor (JobChangeFail/Pass, PartDetect) are not supported.

4.6.9 HTML Webserver

The **CS-60/ID-600** sensor provides an optional built-in web server. The webserver shows a web page and provides a ReST API. Communication between any browser and the built-in webserver is carried out on port `80`, the standard port for HTTP.

Web Page

The **CS-60/ID-600** web page shows information about the vision sensor as well as the loaded jobs. You can also switch jobs and set the boot job. The web page can be opened by typing the IP address of the sensor into the browser address bar, which automatically shows the **Info** page.

Info

Info   Live View   Statistics

Device Information		Jobs

Device Information

Emulator
CS60
Serialnumber: 0000000

Resolution:	any
IP Address:	127.0.0.1
Subnet Mask:	255.0.0.0
Gateway:	---.---.---.---
MAC Address:	00:00:00:00:00:00
Version:	21.2.0.0
License:	Unknown license

Jobs

1   MultiBrightness
Modified: 17/07/2023, 19:00:46
Set as Active Job   Set as Boot Job

4   Pharma
Modified: 23/08/2023, 13:49:45
Set as Active Job   Set as Boot Job

6   ISOIEC 15415
Modified: 27/10/2023, 17:00:55
Set as Active Job   Set as Boot Job

7   Html
Modified: 13/11/2023, 16:05:52
Set as Active Job   Set as Boot Job

Fig. 194: The **CS-60/ID-600 Info** page

You can change jobs with the **Set Active Job** button and set the boot job with the **Set Boot Job** button.

Live View

In addition to the **Info** page, you can also switch to **Live View** to view the inspection results and the captured image. the recorded image.

Fig. 195: The **CS-60/ID-600 Live View** page

On the left you can see a list of the tools in the job and their most important calculation results.

In the middle you can see the image with a graphical visualisation of the tools. You can adjust the desired visualisation by switching the corresponding elements on or off on the right-hand side.

At the bottom you can see the history of the last inspections. You can pause and restart the

history and also click on the individual inspections to study the visualisation and the values at your leisure.

Statistics

You can also use the **Statistics** page to measure the time behaviour of the job and the individual tools.

	Pass	Fail	Mean (ms)	Max (ms)
Job	**46**	**0**	**775.9**	**939.0**
Acquisition	46	0	20.1	108.5
Locate Area	46	0	691.5	850.6
Detect Barcode	46	0	7.6	29.9
Detect Brightness	46	0	8.1	12.5
Measure Circle	46	0	9.3	11.4
Logic	46	0	0.1	0.1
DataHandling	46	0	39.3	89.7
Webserver	46	0	0.1	0.1
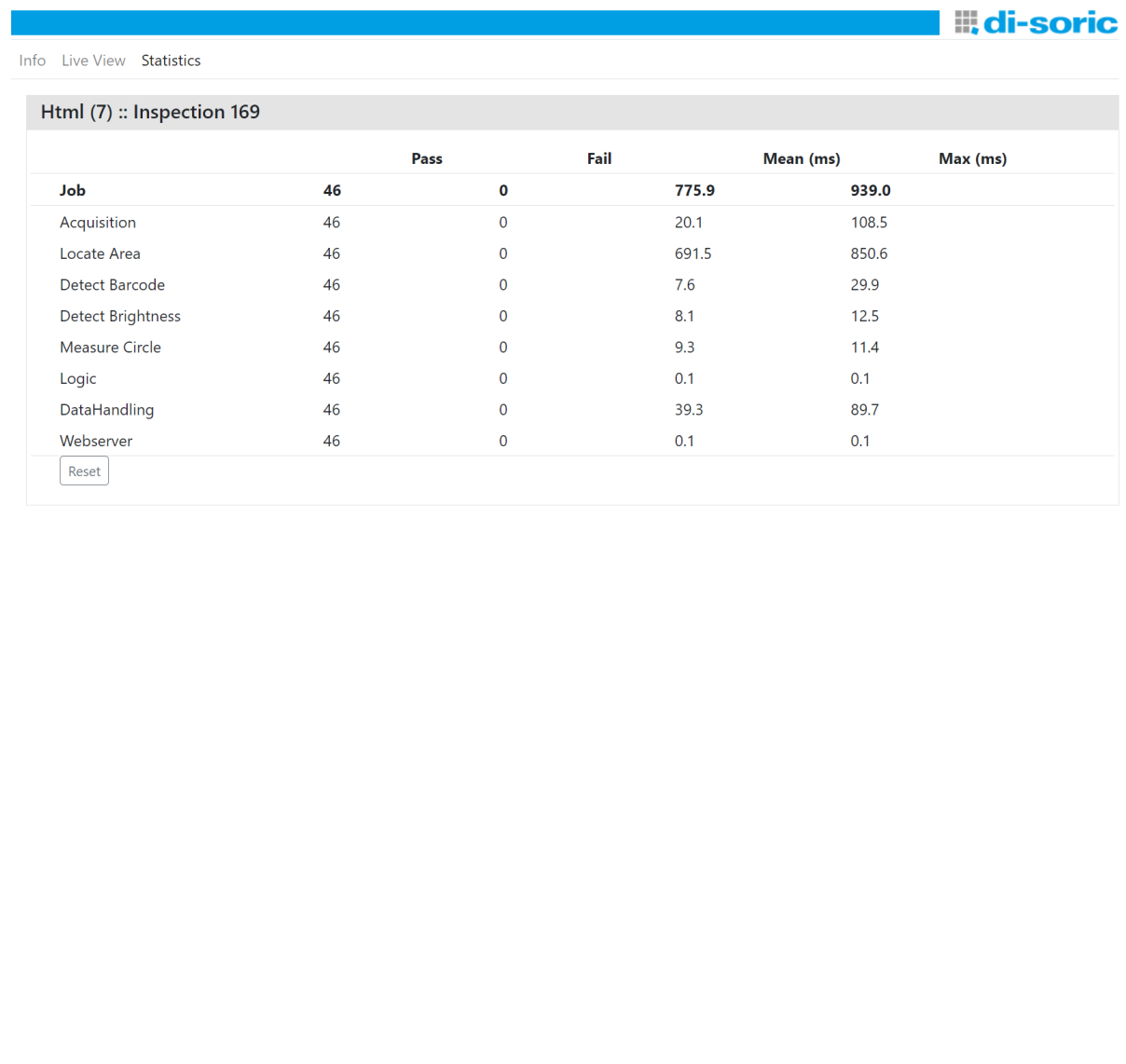
Fig. 196: Die **CS-60/ID-600 Statistics** Seite.

You can reset the statistics to zero using the **Reset** button.

ReST API

The ReST API provides several functions to provide information about the sensor, and to list jobs and change jobs.

Endpoint: `/api/device` HTTP-method: `GET`

Returns information about the device.

Here is an example response in JSON format:

```
{
    "resolution":"736 x 480",
    "userID":"PS-32",
    "serialNumber":"0000032",
    "licenseCode":262143,
    "modelCode":"...",
    "softwareVersion":"22.1.1970.0",
    "ipAddress":"192.168.3.15",
    "macAddress":"8C:0F:A0:00:00:20",
    "subnetMask":"255.255.0.0"
}
```

Endpoint: `/api/jobs` HTTP-method: `GET`

Lists the jobs on the device.

Here is an example response:

```
[
    {
        "slot":1,
        "name":"dauertest",
        "lastModified":"2020-01-31T13:59:49",
        "isBoot":true,
        "isActive":true
    },
    {
        "slot":2,
        "name":"new_job",
        "lastModified":"2020-01-31T13:57:33",
        "isBoot":false,
        "isActive":false
    }
]
```

Endpoint: `/api/jobs/active` HTTP-method: `GET`

Returns the currently active job.

Here is an example response:

```json
{
    "slot":1,
    "name":"dauertest",
    "lastModified":"2020-01-31T13:59:49",
    "isBoot":true,
    "isActive":true
}
```

Endpoint: `/api/jobs/boot` HTTP-method: `GET`

Returns the currently set boot-job.

Example response: see '/api/jobs/active'

Endpoint: `/api/jobs/active?slot=` HTTP-method: `POST`

Sets the active job. The job with the specified slot will be made active.

Endpoint: `/api/jobs/boot?slot` HTTP-method: `POST`

Sets the boot job. The job with the specified slot will be made the boot job.

Endpoint: `/api/results/latest` HTTP-method: `GET`

Returns statistical information for the most recent inspection result.

Here is an example response:

```json
{
"inspectionID": 3467,
"pass": false,
"executionTime": 74.732,
"tools": [
    {
    "name": "Acquisition",
    "pass": true,
    "executionTime": 0.135,
    "secondaryResults": [
        {
```

```
        "value": 3467,
        "name": "Id"
        }
    ]
    },
    {
    "name": "Locate Shape",
    "pass": true,
    "executionTime": 205.8671,
    "secondaryResults": [
        {
        "value": 100,
        "name": "Score"
        },
        {
        "value": {
            "X": 134.48820125311528,
            "Y": 116.50169899415934
        },
        "name": "Center (px)"
        },
        {
        "value": null,
        "name": "Center (mm)"
        },
        {
        "value": 0.6550773065610689,
        "name": "Angle"
        }
    ]
    },
    {
    "name": "Detect Contrast",
    "pass": false,
    "executionTime": 1.1264,
    "secondaryResults": [
        {
        "value": 255,
        "name": "Contrast"
        }
    ]
    },
    {
    "name": "Logic",
    "pass": false,
    "executionTime": 0.0504,
```

```
    "secondaryResults": []
    }
],
"jobName": "new_job",
"jobSlot": 14
}
```

Endpoint: `/api/results/image` HTTP-method: `GET`

Returns the captured image.

4.6.10 Changing jobs with the digital inputs

The digital inputs of the **CS-60/ID-600** can either be used for job changes or for input linking.

The two inputs are used as **Gate** and **Counter**. Input 0 (brown) is the **Gate**, input 1 (blue) is the **Counter**.

If the **Gate** is set to level 1 (high), the rising edges are counted at the counter.

Every rising edge at the **Gate** resets the counter. Every falling edge at the **Gate** takes the number of counted rising edges of the **Counter** and loads the corresponding job.

The **Counter** for the job change may be switched with a maximum of 250 Hz.

4.7 Hardware

4.7.1 Cable Pinout

The **CS-60/ID-600** communicates with the outside world via two cables: an X-coded LAN-cable and an A-coded IO-cable.

A-coded IO-cable

Signal	Con-tact	Color	Description
+UB	7	black	
GND	8	grey	
Digital Out 0	11	grey/p	Digital output 0 (controlled by Logic tool).
Digital Out 1	12	red/blu	Digital output 1 (controlled by Logic tool).
Digital Out 2	3	white	Digital output 2 (controlled by Logic tool).
Digital Out 3	4	green	Digital output 3 (controlled by Logic tool).
Trigger In	10	violet	Digital trigger (allowed only when Ready - wait 100 µs between rising edge of Ready until the next trigger is issued).
Digital In 0	1	brown	Digital input 0 (use for Job Change or linking).
Digital In 1	2	blue	Digital input 1 (use for Job Change or linking).
Common Interface	6	yellow	
Ready	5	pink	Ready output.

4.7.2 LED

The CheckSen **CS-60/ID-600** sensor has three LEDs at the side.

Fig. 197: The **CS-60/ID-600** LEDs.

The LEDs show the following information

Color	Description
green	Power, operating voltage is applied.
yellow	Blinks to show LAN communication.
yellow	Shows an external trigger.

CREDITS

The following third party components are used under the respectively mentioned licenses:

5.1 Boost

http://www.boost.org

License

5.2 Bootstrap

http://www.bootstrap.com

License

5.3 bzip2

http://www.bzip.org/

License

5.4 CommandLine

https://github.com/commandlineparser/commandline

License

5.5 Debian Linux

http://www.https://www.debian.org/legal/licenses/

Apache-2.0

Artistic

BSD

Artistic

GFDL-1.2

GFDL-1.3

GPL

GPL-1

GPL-2

GPL-3

LGPL

Artistic

LGPL-2.1

LGPL-3

5.6 Dynamic Expresso

https://github.com/davideicardi/DynamicExpresso

License

5.7 Eigen

http://eigen.tuxfamily.org

License

5.8 Extended WPF Toolkit

https://github.com/xceedsoftware/wpftoolkit

License

5.9 FontAwesome Fonts

https://fontawesome.com/

License

5.10 FreeImage

http://freeimage.sourceforge.net/

License

5.11 IndependentJPEGGroup

This software is based in part on the work of the Independent JPEG Group.

http://ijg.org/

License

5.12 INI File Parser

https://github.com/rickyah/ini-parser/

License

5.13 IronPython

http://ironpython.net/ https://github.com/IronLanguages/ironpython2.git

License

5.14 Json.NET

https://www.newtonsoft.com/json

License

5.15 KBCsv

https://github.com/kentcb/KBCsv.git

License

5.16 libjpeg-turbo

https://github.com/libjpeg-turbo/libjpeg-turbo/blob/main/LICENSE.md

License

5.17 libpng

http://www.libpng.org/pub/png/libpng.html

License

5.18 libsimdpp

https://github.com/p12tic/libsimdpp

License

5.19 MessagePack-CSharp

https://github.com/neuecc/MessagePack-CSharp

License

5.20 NetworkView

https://www.codeproject.com/Articles/182683/NetworkView-A-WPF-custom-control-for-visualizing-a

License

5.21 NLog

http://nlog-project.org/ https://github.com/NLog/NLog/

License

5.22 OxyPlot

http://www.oxyplot.org/ https://github.com/oxyplot/oxyplot

License

5.23 Microsoft.Diagnostics.Tracing.TraceEvent

https://github.com/Microsoft/perfview

License

5.24 Prism

https://github.com/PrismLibrary/Prism

License

5.25 signalr.min

https://raw.githubusercontent.com/stefanpenner/es6-promise/master/LICENSE

License

5.26 System.Commandline

https://github.com/dotnet/command-line-api

License

5.27 Sphinx RTD Theme

https://sphinx-rtd-theme.readthedocs.io/en/latest/index.html

License

5.28 SSH.NET

https://github.com/sshnet/SSH.NET

License

5.29 The Helper Trinity

https://github.com/kentcb/TheHelperTrinity/blob/master/LICENSE

License

5.30 Threading Building Blocks

https://www.threadingbuildingblocks.org/

License

5.31 VTK

http://www.vtk.org

License

5.32 WpfLocalizeExtension

https://github.com/XAMLMarkupExtensions/WPFLocalizationExtension/

License

5.33 WPF Task Dialog Wrapper

https://github.com/yadyn/WPF-Task-Dialog

License

5.34 zlib

http://www.zlib.net/

License

nVision-i Enduser License Agreement (nVision-i EULA)

You receive and take note of these present terms in conjunction with the offer to make use of one or several components of the software nVision-i (jointly and each component separately hereinafter referred to as the "Software"). **BY DOWNLOADING, INSTALLATION, COPYING OR BY ANY OTHER FORM OF USAGE OF THE SOFTWARE, YOU EXPRESS AND DECLARE YOUR CONSENT TO, AND ACCEPT AS BINDING, THE TERMS AND CONDITIONS OF, AND CONTAINED IN, THIS PRESENT ENDUSER LICENSE AGREEMENT.**

All and any components of the Software nVision-i have originally been available by di-soric Imaging GmbH & Co. KG, Schlingener Strasse 4, DE-86842 Türkheim, Germany (hereinafter: Producer), and are being distributed to you through the distribution chain, i.e. by di-soric GmbH & Co. KG, Steinbeisstraße 6, DE-73660 Urbach, Germany, by one of the di-soric company group members abroad or by one of the international di-soric distribution partners, your respective responsible contract partner (hereinafter: Individual Contract Partner), on the basis of the Individual Contract Partner's individual license terms, as agreed individually between you and the Individual Contract Partner, and, if applicable, together with accompanying media, a product documentation (in particular the User Manual) and hardware such as a vision sensor. These individual license terms of your Individual Contract Partner define the scope of the permitted use of the Software and insofar they form a separate, supplementary set of contractual terms in addition to this present License Agreement legally valid and binding on you as part of separate contractual relationship.

1. Intellectual Property

The Software is, and forms, intellectual property protected by law and under treaty terms, in particular under German copyright, European and international law and treaties. All right, title and interest in and to the Software, including any accompanying material such as the documentation of the Software, are owned by, and rest with **di-soric Imaging GmbH & Co. KG, Schlingener Strasse 4, DE-86842 Tuerkheim, Germany, Telephone: +49 (0) 82 45 774 96 00, imaging@di-soric.com.**
Regarding open source software components, reference is made to the text file, the open source directory and the User Manual, all of which forming part of your installation process; the license terms and conditions cited and contained therein apply in addition to the terms of this present EULA and, within their scope of application, take precedent over this present EULA.

2. License

The Software is licensed, not sold, to you and you are being granted the non- exclusive license to use the Software for its intended purpose and in accordance with the User Manual, i.e. to evaluate the use of di-soric vision sensors in the context of an automated production and for the calibration and control of di-soric vision sensors. The Software is being provided in machine-readable form (object code). Except for the making of one copy of the Software for backup purposes and copies for further applications of the Software for the before mentioned purposes, any other usage of the Software is prohibited. In particular, any alteration, re-engineering, translation or extraction of portions of the Software is forbidden. The same applies to any form of decompilation other than where required to achieve interoperability ((§ 69i German

Copyright Act (UrhG)). You may also transfer the Software for the above named purposes, subject to revocation due to an important reason. Any such duplication or transfer of the Software has to include this present EULA and any and all of the intellectual property notices and markers contained in the Software and has to safeguard the compliance with these present terms and conditions.

3. Warranty, Maintenance of the Software

The range of functions, the system requirements and the technical specifications of the Software are defined in and to be determined by the User Manual of the Software. The Producer offers software updates in accordance with the legal requirements. Any warranty claims are exclusively restricted to, and governed by the contractual stipulations between you and your Individual Contract Partner; the same applies with regard to any kind of software maintenance and support. This present EULA does in particular not include any representations extending beyond its terms and especially with regard to the right to use, and the extent of such right, under the contractual stipulations between you and your Individual Contract Partner.

4. Term and Legal Remedies

This present license is valid for an unlimited period. The Producer is entitled to terminate and cancel this present license with immediate effect upon violation of any of the terms of this present EULA. Upon termination by the Producer, you are under the strict obligation, and have to, cease and desist from any further use of the Software and produce, and hand over, to the Producer any and all copies of the Software or, upon respective request by the Producer, instead permanently delete and destroy any and all installations and copies of the Software and confirm so in writing to the Producer.

5. Export Control; No Military Use

Compliance with any and all export control laws, regulations and guidelines in place, valid and binding on you is part of your own and sole responsibility and you confirm to abide by and obey all and any such rules and laws. Any use for military purposes is excluded. You also confirm that you do not belong to any group or body of persons or companies, which the Software must not be made available to under export control or embargo rules applicable under German law or under the law valid and applicable at the place of business of, or otherwise binding for, your Individual Contract Partner.

6. General Provisions, Language, Forum and Applicable Law

Should any provision of this present EULA be or become legally invalid or inappliable, the validity or applicability of all of its other provisions shall remain unaffected. With regard to any matter of interpretation of any provision contained in this present nVision-i EULA, the English language version shall be decisive for determining its meaning or its scope. The original English text of this present nVision-i EULA is, and shall be upheld as being solely binding.
This present EULA, and any dispute arising in connection with this present EULA, shall exclusively be governed by and construed in accordance with the laws of the Federal Republic of Germany (FRG). If you are a merchant, a legal entity under pub-

lic law or a public special fund entity, or if you do not have a general local domicile with legal venue in the FRG or should you relocate your residence or normal place of abode out of the FRG, or should your residence or normal place of abode be abode at the time the legal action in question is brought is unknown, the exclusive venue for any and all disputes arising out of or in connection with this present EULA is the place of business of the Producer; the Producer, however, reserves the right to institute legal proceedings at the general court of jurisdiction at the place of your main seat of business, in particular including any application other instrument seeking injunctive relief or other interim or provisional remedies.

nVision-i Terms of Use Agreement (nVision-i TUA)

di-soric GmbH & Co. KG, Steinbeisstrasse 6, DE-73660 Urbach, Germany (hereinafter referred to as di-soric) distributes all and all components of the software nVision-i (each such component separately and any such components jointly referred to hereinafter as Software), as a download or as so-called embedded Software together with a di-soric vision sensor, either directly or through one of the international companies of the di-soric group of companies as direct licensors or through other licensees, in particular via one of the worldwide di-soric distribution partners. Depending on your source of supply, the respective supplier of your website download or of the embedded software version, whether or not in your individual case accompanied by media such as product documentation (esp. the nVision-i User Manual), is the distributor responsible to you contractually (hereinafter: your Individual Contract Partner). The following terms represent the specific and primarily valid license conditions agreed between you and your Individual Contract Partner.

The specific scope of use licensed to you according to this present nVision-i TUA is determined additionally, on a secondary level, either, if applicable, by any referenced general terms and conditions of sale of your Individual Contract Partner, if and insofar as these conditions do not contradict or amend the nVision-i EULA and this present nVision-i TUA, or alternatively by the General Terms and Conditions of Sale and Delivery (GTSD) of di-soric in the version currently valid at the time of licensing, as also made available herewith in the version GTSD 07/2017 and as available for review and download on the internet under www.di-soric.com in the currently valid version. Any general terms and conditions of business or of purchase of yours shall not apply.

1. Intellectual Property

The Software forms, and is protected legally and contractually as, intellectual property, in particular on the basis of German copyright law, European and international legal regulations and treaties.

In addition to these present terms, and with overriding effect, the nVision-i Enduser License Agreement (nVision-i ELV) provided to you together with this present nVision-i TUA applies.

In addition to the present terms, and insofar also with overriding effect, regarding OpenSource- software components, as explained in the nVision-i ELV, the OpenSource text file included in connection with the installation, the OpenSource directory, the nVision-i User Manual and in particular, if and insofar as applicable, the license conditions quoted and contained therein apply with priority over this present license agreement.

2. License / Granted Rights of Use

You are hereby granted the license and non-exclusive right to use the Software in accordance with the terms of the nVision-i User Manual, i.e. to evaluate the use of di-soric vision sensors in the context of an automated production process and for the purpose of calibration and control of di-soric vision sensors. The Software is made

available to you in machine-readable form (object code). You are not entitled to claim any access to the source code or to demand to be provided with a copy of otherwise possession of the source code; § 69e of the German Copyright Act (UrhG) remains unaffected. Apart from the creation of a backup copy and of copies for further use of the aforementioned kind, any other use is strictly excluded. In particular, any modification, reverse engineering, translation or extraction of parts of the Software is not permitted. The aforesaid also applies with regard to any form of decompilation, except where necessary in order to facilitate interoperability (§ 69i UrhG).

You are also granted the license and right to transfer the Software to third parties for the aforementioned purposes, subject to revocation for good cause. If you have obtained the Software in the form of embedded software, any further use of this Software must be completely abandoned, when the hardware is transferred together with the embedded Software, and all copies made must be deleted. Any reproduction or transfer of the Software must include this License Agreement and all intellectual property notices and must ensure compliance with these terms and notices. Copyright notices, serial numbers and other features for program identification may not be removed or altered/ changed.

Services other than the present license and granting of rights of use, in particular any installation or configuration services, are not subject of this present nVision-i TUA.

3. Your Responsibility as User

You are solely responsible for setting up a functional hardware and software environment, which is sufficiently dimensioned for the functions of the Software. This responsibility extends to observing the instructions for installation and operation of the Software contained in the nVision-i User Manual and to regularly checking the website of di-soric or of your Individual Contract Partner for new instructions with regard to the operation of the Software or with regard to updates. This responsibility also includes taking reasonable measures of precaution for the event that the Software does not work properly in whole or in part, e.g. a regular data backup, failure diagnosis and continuous checking of the results from data processing.

Any use of controller software requires a thorough examination with regard to the absence of defects and usability as part of the existing hardware and software configuration, before it is being put to use in the ongoing operation. The same also applies with regard to the Software.

It is your duty to safe, install, operate and keep the Software stored in a secure manner in order to prevent unauthorised access and copying. In particular, any copies of the Software and any access data must be kept in a suitably protected environment. Any attempt at or actual circumvention of any and all technical measures for the protection of the Software is strictly prohibited.

4. Warranty and Software Maintenance

The scope of functions, the system requirements and the technical specifications of the Software (jointly hereinafter referred to as the Specifications) are set out in and to be determined in accordance with the nVision-i User Manual, which is being made

available to you for review and download separately and is also provided to you together with the Software in the course of the installation of the Software. Updates of the Software are generally made available as required under the applicable legal regulations.

Any and all warranty claims are exclusively governed by and determined in accordance with the following contractual terms agreed between you and your Individual Contract Partner; the same applies with regard to any Software maintenance:

Your Individual Contract Partner warrants that the functionality of the Software essentially corresponds to the Specifications; the applicable warranty period begins with the download of the Software or otherwise at the time of its delivery. Any warranty claims become statute- barred after expiry of such warranty period as agreed upon individually, under the respective valid GTSD or, respectively, under any applicable general terms and conditions of sale of your Individual Contract Partner. If a software maintenance agreement is in place, the applicable period for remedying any defect is to be determined according to the periods agreed upon therein. Details set out or specified in the Specifications or in the Software documentation do not constitute, and are not meant to mean, guarantees in the legal sense, unless there is an express provision attached to it to the effect of strict liability without fault.

In order to fulfil a warranty claim, your Individual Contract Partner will, at its free option, remedy a defect in the Software by repair, by procuring replacement software or by upgrading or releasing a new version of the Software. A deviation from the Specifications is considered a material defect, if such defect is proven and can be reproduced. In the event of a defect of title, your Individual Contract Partner will provide you, at his free option, with either a legally unobjectionable possibility of using the Software or with a version of the Software modified in such a way that the Software no longer infringes any third-party rights. Your Individual Contract Partner is entitled to two attempts of rectification of a defect. If he does not succeed in rectifying a defect within a reasonable period of time, you are entitled to a reduction of the agreed remuneration ("price reduction"). Instead of reducing the price, you are also entitled to withdraw from the contract, if the defect is substantial.

You are not entitled to any warranty claims,
☐   if you do not use the Software as intended or in an improper manner, or
☐   if you have modified or altered the Software without the prior written consent of your Individual Contracting Partner, or
☐   if problems or faults occurred because the Software was used with programs that are not compatible with the Software,
unless you prove that the respective defect is the effect of a fault of the Software.

If you are entitled to a contractual or statutory warranty claim for compensation or reimbursement of futile expenses, such claim is governed by and determined in accordance with the following regulations on liability.

5. Liability

If di-soric is your Individual Contract Partner, the liability regulations contained in the respective valid GTSD apply. In all other respects, the liability of your Individual Contract Partner is, unless other exclusions and limitations of liability have been agreed with you individually or in general terms and conditions of sale of your Individual Contract Partner, to be assessed and determined as follows:

Your Individual Contract Partner is liable for culpable injury to life, limb or health; furtheron, for wilful damage or gross negligence on the part of its owner, its representatives or executives, for fraudulently concealed defects or for liability under a guarantee in the legal sense or under the German Product Liability Act (Produkthaftungsgesetz (ProdHG)).

Your Individual Contract Partner is also liable for damages in the event of a wilful or negligent breach of a material duty, i.e. of an essential contractual duty required to be performed in order to execute the contract and thus a duty you may reasonably rely on to be fulfilled, with this liability being limited, though, to the predictable damage typical of this type of contract, unless it is a case of wilful damage or gross negligence or of or a damage to life, limb or health.

In the event of a wilful or negligent breach of a duty to provide information or of other secondary obligations, this present clause 5 and the preceding provisions apply mutatis mutandis excluding any further liability. In particular, your Individual Contract Partner is not liable for damages resulting from unsuitable or improper use of the Software or from use other than the intended use of the Software.

Any further liability for damages is excluded. The distribution of the procedural burden of proof in a legal action remains unaffected by the above provisions on liability. The same applies regarding any right of recourse of the producer (so-called Herstellerregress pursuant to Sec. 478 German Civil Code (BGB)). Any of the preceding exemptions from or limitations of liability extend to the personal liability of the employees, representatives and auxiliary personnel of your Individual Contract Partner.

6. Duration and Remedies

This present license is valid for an indefinite period. Your Individual Contract Partner has the right to terminate this license with immediate effect in the event of a violation of these present license terms. In the event of termination by your Individual Contract Partner, you must cease immediately all further use of the Software and return any and all copies of the Software to your Individual Contract Partner or, instead, upon request by your Individual Contract Partner permanently delete all installations and copies and confirm to have done so in writing to your Individual Contract Partner.

7. Export Regulations; No Use for Military Purposes

Compliance with all export regulations applicable to you is solely your responsibility and you hereby undertake to observe any such regulations. Any use of the Software for military purposes is excluded. You also warrant that you are not and that you do not belong to, or form part of, a group of persons or companies to which the Software

may not be made available under German export or embargo regulations or those regulations applicable to your Individual Contract Partner.

8. Final Provisions, Language, Forum and Applicable Law

Should one or more provisions of this present nVision-i TUA be or become invalid, the validity of the remaining provisions remains unaffected. With regard to any matter of interpretation of any provision contained in this present nVision-i TUA, the English language version shall be decisive for determining its meaning or its scope. The original English text of this present nVision-i TUA is, and shall be upheld as being solely binding.

The legal relationship between you and your Individual Contract Partner and all disputes arising out of or in connection with this present license agreement are subject to, and are to be determined in accordance with, the laws the contract which your Individual Contract Partner is also subjected to; if there is doubt as to the applicable law, the law of the Federal Republic of Germany (FRG) applies. If you are a merchant, a legal entity under public law or a special fund under public law or if you have no general place of jurisdiction in the country of the headquarters of your Individual Contract Partner or if you transfer your domicile or usual place of residence from this country after conclusion of the contract or if your domicile or usual place of residence is unknown at the time of filing of a law suit, the exclusive place of jurisdiction for all disputes arising out of or in connection with this present nVision-i TUA is the headquarters of your Individual Contract Partner; your Individual Contract Partner is free to file law suits at its headquarters and may in particular apply for instruments and seek interim legal protection.

General Terms of Sale and Delivery (GTSD) 07/2017 of
di-soric GmbH & Co. KG, DE-73660 Urbach, Germany

1 General Provisions

1.1 Our deliveries and other contractual performances shall solemnly and exclusively be governed by these General Terms of Sale and Delivery (GTSD) in their latest version. General terms and conditions of Purchaser, whether diverging from, opposing to or supplementing these terms and conditions, shall not apply unless such being agreed upon individually, explicitly and in writing, even if we carry out orders knowing of such terms and conditions.

1.2 In the course of an on-going business relationship, our GTSD shall also apply to our future supplies and deliveries irrespective of such being explicitly agreed upon. Our GTSD shall also apply to any precontractual negotiations.

1.3 All intellectual property such as titles, other property rights, rights of exploitation and copyrights concerning cost estimates, drawings or other documents (for example in the context of individual client customized solutions) shall be reserved in our name. Such documents may only be made available to third parties with our express, prior and written consent and shall be returned to us, should the respective order in question not be placed with us. Any software provided shall be licenced to Purchaser only on a non-exclusive basis and only for use in its original and un-altered condition on the agreed hardware using the agreed parameters and solemnly in accordance with the conditions of licence offered.

1.4 Purchaser agrees to the digital storage and processing of his own details and the details of the purchase contract in accordance with the provisions of the German Federal Data Protection Act (Bundesdatenschutzgesetz (BDSG)).

2 Offer

2.1 Our offers shall solemnly be made subject to confirmation and shall only be construed as an invitation to treat.

2.2 References made directly or indirectly in our offers, to technical details, such as measures, weights, performance/performance features or physical or material quality, and any similar application related specifications shall only constitute mere guidelines within the usual limits in the trade, unless where explicitly stipulated otherwise in writing in the offer.

2.3 Technical alterations to the goods to be delivered or alterations to their respective design engineering shall be deemed accepted, except where the alteration is unreasonable or reduces the fitness for use.

2.4 We are entitled to deliver or perform according to the respective current state of the art, even if it changes after the conclusion of the contract.

3 Conclusion of Contract, Call-Off Orders, Samples

3.1 Purchaser's offers shall be valid and binding for a period of at least fourteen days, calculated from the day of receipt.

3.2 A binding and valid contract requires either our written confirmation of the respective order or our respective delivery or other contractual performance, the first of these alternatives to take place forming the relevant event. The order confirmation determines the scope of the contract; in the absence of such, our offer shall be decisive.

3.3 Call-off orders, i.e. orders for a particular overall quantity of goods to be delivered in partial quantities upon call-off within a specific period of time, require Purchaser to call such quantity off in one or more single quantities within the agreed time, otherwise within twelve months from the conclusion of the contract. Should Purchaser not call off the overall quantity within the call-off time period, we shall no longer be under an obligation of performance in advance; instead, we shall be entitled to deliver the remaining quantity subject to immediate payment. This provision shall not limit any possible rights to withdraw from the contract or any possible claims for damages.

3.4 In Germany, sample deliveries shall be deemed a sale on approval, unless explicitly stipulated otherwise. Deliveries of samples for export purposes shall be available as a sale for a special price. Sample equipment delivered in Germany shall be returned to us undamaged and with the original packaging within two weeks after the date of dispatch and at Purchaser's expense, unless Purchaser pays the invoice attached to the sample within this period of time. Samples sent abroad cannot be returned. Title to and ownership of samples shall rest with us until full payment of the invoice has been effected and we have received this payment.

4 Prices

4.1 Our prices shall cover delivery Ex Works (EXW Incoterms 2010) including the packaging of the equipment, unless explicitly stipulated otherwise. The transport packaging, the costs of transport, the costs of transport insurance, customs duties and unloading shall be invoiced separately on a lump-sum-basis. The statutory Value Added Tax on the due day shall be invoiced additionally.

4.2 Deliveries made without specific pricing agreement shall be invoiced according to the list price valid and officially available at the day of delivery.

4.3 The minimum value of each order shall be 50.00 € per German order as well as per foreign order. Smaller orders shall be processed at a separate handling fee of 25.00 €.

4.4 Should the time period between the conclusion of the contract and the date of delivery exceed four months, both Parties shall be entitled to adjust the contractual prices in accordance with an actual change of wages or of costs of material since the conclusion of the contract by at least 10 percent; such change shall be excluded, though, in cases of call-off orders under Clause 3.3, should fixed pricing be guaranteed.

4.5 Partial deliveries, including such deliveries in the context of call-off orders under Clause 3.3, shall be invoiced separately.

5 Payment, Late Payment, Exclusion of Set-Off

5.1 Our invoices shall be payable, subject to Clause 3.4, within thirty days, from the date of invoice, strictly net and always free point of payment as indicated.

5.2 All and every payments shall be made in EURO.

5.3 Means of payment other than cash or bank transfer of monies shall only be accepted if so stipulated in advance and, even if so stipulated, only as conditional payment. Provision of cheques or bills of exchange shall only constitute payment, if and when the last bill of exchange has been honoured. Acceptance of a bill of exchange shall only be deemed to constitute a prolongation of payment, if and when explicitly agreed upon in writing; Purchaser shall bear the interest as well as any costs and expenses, including bank charges.

5.4 Rebates or cash discounts shall only be deducted, if and when explicitly agreed upon in writing. Furtheron, any deduction of rebate shall only be admissible if we receive all payments of the order within the respective rebate time period.

5.5 If payment is received after the due date, we may charge a default interest of 8 percentage points above the base lending rate pursuant to Sec. 247 of the German Civil Code (BGB) (so-called Basiszinssatz under German Law), without prejudice to our claims for

actual damages exceeding the aforementioned amount.

5.6 Purchaser shall only set off claims against us which have been adjudicated by a judgment not subject to an appeal or for which the time to appeal has expired, which are not in dispute or have been acknowledged by us or are otherwise due and certain.

5.7 Should Purchaser delay payment for more than four weeks, should payment by cheque or bill of exchange be protested or should execution be imposed on Purchaser's assets, we may give notice to make further deliveries to Purchaser, whether owed under separate orders or under call-off orders, conditional upon advance payment.

Upon a significant deterioration of Purchaser's financial situation or other events seriously affecting his creditworthiness, all our claims become due and payable with immediate effect, even if we have received letters of credit. This shall apply in particular, if

- Purchaser is late with the payment of considerable dues for more than four weeks,
- the value of the securities provided for this particular contract decreases considerably,
- the financial situation of the personally liable shareholder deteriorates significantly or is seriously at risk or
- the personally liable shareholder dies or changes.

In such event, we are entitled to make further delivery conditional upon payment of all outstanding accounts and advance payment or the provision of securities. Should Purchaser refuse to do so, we shall be entitled to cancel the contract.

5.8 Purchaser may only exercise a right of retention, should his counter-claim be based on the same (single) order in question or, in a case of a call-off order, the same (single) call-off in question. Furthermore, such counter-claim shall require to fulfil the requirements as stipulated in Clause 5.6.

6 Time of Delivery, Impossibility of Performance

6.1 Any agreed delivery times (that is both delivery times and delivery dates) shall only be binding, if Purchaser has given any and all information, clearances, approvals of plans and such alike and has provided any and all construction drawings agreed upon, materials to be provided by him and testing equipment (hereinafter: assistance tasks) necessary for the production and delivery of the ordered goods, in a way that puts us in a position to achieve the respective goal (hereinafter: assistance goal); furtheron, that we are in receipt of any advance payments agreed upon; otherwise, the delivery time shall be postponed or prolonged accordingly until even the last of the beforesaid requirements has been fulfilled, unless we are responsible for such delay.

The delivery time starts upon receipt of the confirmation of order.

Agreements about delivery dates shall require execution in writing in order to be valid and binding. A delivery time shall be met if, as agreed, either the goods to be delivered have left our place of business or we have given notice of readiness for dispatch in due time.

6.2 Correct and punctual deliveries by our own suppliers shall be reserved (so-called Selbstbelieferung under German law).

6.3 Agreed delivery times are reasonably postponed or prolonged in the event of industrial action, in particular strike and lock-out and in the event of extraordinary circumstances, which are beyond our control and impede on our contractual performance such as, but not limited to,

(i) machinery breakage, other damage to machines or machine parts,

(ii) acts of God, in particular civil war, mobilizations, blockades, acts of war, rebellion, sabotage, hurricanes, tornado storms, flood/low water, earthquakes, seismic/tsunami waves or other acts of nature, fire, explosion,

(iii) consequences of an energy crisis with impact to the national economy, shortage of raw materials and supplies, fuel, auxiliary material or energy,

(iv) seizure, import or export bans, embargoes or other administrative measures affecting us or our suppliers.

The same applies if similar restraining events affect a sub-supplier. Even if we are in default, we are not responsible for the aforementioned event. We will inform Supplier about the occurrence of such event without delay, its end and the expected new delivery date.

Should any of the before mentioned events render our commenced commitment to the order unreasonable, we shall be entitled to cancel the contract.

If such event persists for longer than six months, the contract shall be adjusted in good faith. Only if such an adjustment is economically unreasonable for one of the Parties, such Party can cancel the contract. If we intend to cancel the contact, we will inform Purchaser without undue delay, even if an extension of the delivery time had already been agreed.

6.4 Purchaser shall be responsible for the compliance with and the application of all relevant foreign trade and payments regulations (import licences, foreign exchange transfer etc.) and any other applicable law valid outside Germany, save where expressly stipulated otherwise.

6.5 Should we be responsible for a delay of delivery, Purchaser may cancel the contract after a reasonable extension of time to be granted to us has elapsed; should the delivery become impossible, granting of an extension of time shall not be required.

Any claims for damages on the basis of a delay with delivery (including compensation for consequential loss) as well as any claims for reimbursement of expenses pursuant to Sec. 284 of the German Civil Code (BGB) shall exclusively be governed by Cipher 9 (Liability).

6.6 The limitation of liability for delay with delivery under Ciphers 6.5 shall not apply to mercantile contract of sale with an absolutely fixed delivery date pursuant to Sec. 376 German Commercial Code (HGB) (so-called Fixhandelskauf under German law).

7 Consignment and Passing of Risk

7.1 Partial delivery shall be allowed in reasonable quantities.

7.2 We shall deliver „Ex Works" (EXW Incoterms 2010), save where expressly stipulated otherwise. Should dispatch be part of our duties, we may choose the dispatch type sequence, the mode of dispatch and the carrier, save where expressly stipulated otherwise. Ordered modes of dispatch shall not be subject of any warranty. Only upon Purchaser's express request and at his expense dispatch shall be insured against damages due to theft, breakage, fire, water or other insurable risks.

7.3 The risk of accidental loss of or accidental damage to the goods shall pass over to Purchaser at the latest with the picking up of the goods by Purchaser or otherwise with dispatch to the carrier, irrespective of whether it is a partial or complete delivery or whether we have accepted other duties, such as the bearing of costs of shipping or the delivery to Purchaser.

7.4 The risk of accidental loss of or accidental damage to the goods shall also pass over to Purchaser in the event of a default in taking delivery or after ten days from the notice to Purchaser of readiness for dispatch. We shall insure the goods upon written order of Purchaser to that effect at his own expense.

7.5 Should Purchaser be responsible for delayed dispatch, we may charge for any actual costs of storage starting ten weekdays after the default in dispatch or after having given notice of readiness for dispatch; storage in our own facilities shall be charged for at a rate of 0.5 percent of the net value of the goods stored per month or part thereof. Alternatively, we may, after having granted reasonable respite, make other use of the goods and deliver to Purchaser goods of the same kind within duly and reasonably prolonged period of time.

7.6 Without prejudice to any warranty claims, Purchaser shall accept delivery, even if the goods delivered should show minor defects.

8 Responsibility for Defects, Custom-Made Products

8.1 Our goods fulfil all requirements under the technical rules and regulations applicable in Germany. The beforesaid shall also apply regarding any foreign technical rules and regulations, which are not applicable in Germany, if explicitly agreed so in writing.

8.2 With regard to deliveries of custom-made products produced according to specifications of an order that deviates from our product catalogue, Purchaser shall accept any consequential quantitative differences to the order customary in the trade up to a limit of 5 % above or below the quantity ordered.

8.3 Purchaser shall inspect any goods delivered immediately upon receipt for identity, quality or quantity deviations; notice of any obvious defects shall be given in writing without delay. Notice of hidden defects shall be given without delay after discovery. Should notice not be given in due time, the goods shall be deemed accepted by Purchaser.

8.4 Goods ordered shall only be or deemed to be defective, if and insofar as they show more than just minor deviations from the contractual quality or fitness for usage, according to the test of reasonableness, which result from circumstances antecedent to the passing of the risk, as there are in particular faulty design, materials or workmanship.

8.5 In particular, we shall not be liable for any consequences of Purchaser`s failure to observe our operating instructions, security instructions or data sheet, the instructions regarding the assembly, connection and putting into operation or the technical data; we shall neither be responsible for faulty servicing or cleaning, inappropriate or unprofessional use, natural wear and tear, influences of not permitted chemical, electro-chemical, electrical or physical nature (i.e. fluids) nor for modifications or repair work by Purchaser or third parties, which had not been authorized by us in advance.

8.6 We shall remedy defects as follows:

Defects appearing within twenty-four months (current products of the brand di-soric) or within twelve months (products of other brands and discontinued products of the brand di-soric) after the passing of risk shall be remedied at our choice either by repair or by delivery of a replacement (supplementary performance); in the event of intent, fraudulent non-disclosure or violation of a guarantee, the statutory warranty period shall apply. We may refuse supplementary performance for as long as Purchaser is in default with payment of such part payment equalling the part of the delivery free from defects. Replaced parts become our property.

Purchaser shall give us the reasonably required time and opportunity necessary for remedying the defect. Such shall include an analysis of the cause of the defect, if necessary on site. Should a notice of a defect prove incorrect, because we are not liable for the cause of the defect, Purchaser shall be charged for the relevant servicing, in particular any tests and on-site services.

Only in urgent cases of danger to the operating safety, if disproportionately high damages need to be avoided or if we should be in delay with remedying a defect, Purchaser may, if he gives us advance notice of such circumstances and such intention without delay, remedy the defect himself or have it remedied by a third person.

Should remedying of the defect prove impossible or should have failed twice or should remedying of the defect or delivery of a replacement not have been accomplished or in a case of a fixed-date-transaction according to Sec. 376 HGB (German Commercial Code) occurred only with delay due to our fault, Purchaser may choose between a reasonable reduction of the contractual price or the cancellation of the contract; the latter being excluded in cases of only minor deviation from the contract.

The right to reduce the contract price or cancel the contract shall be excluded, once the right to ask for a supplementary performance is statute-barred.

9 Liability for Damages

9.1 We shall accept liability for damages or expenses (hereinafter: damages) as follows:

We shall accept liability for any wilful or negligent damage to life, limb or health; furtheron, for wilful damage or gross negligence by the owners of our company, our representatives and executive employees; further for fraudulent non-disclosure of defects, on the basis of a guarantee and of the German Product Liability Act (Produkthaftungsgesetz).

Furtheron, we shall accept liability for damages in the event of a wilful or negligent breach of a material duty, i.e. an essential contractual duty required to be performed in order to execute the contract and thus a duty Purchaser may reasonably rely on to be fulfilled, with this liability being limited to the predictable damage typical of this type of contract, unless it is a case of wilful damage or gross negligence or a damage to life, limb or health.

In the event of a wilful or negligent breach of a duty to provide information or of other secondary obligations, Cipher 8 and the preceding provisions shall apply mutatis mutandis excluding any further liability.

Any further liability for damages shall be excluded.

9.2 The distribution of the burden of proof in a legal action shall not be affected by the preceding provisions. The same shall apply regarding any right of recourse of the producer (so-called Herstellerregress pursuant to Sec. 478 BGB).

9.3 Any of the preceding exemptions or limitations of liability shall extend to the personal liability of our employees, representatives and auxiliary personnel.

10 Retention of Title

10.1 Ownership of the goods delivered, that is both title in and right of disposal of, shall remain with us (hereinafter: retention of title) until full payment of all open accounts within the framework of our business relationship with Purchaser, including any claims for interest or reimbursement of costs, has been made. Neither entering of particular claims into an open or running account nor balancing of credit and debit sides and having the result confirmed by us shall affect the retention of title.

10.2 Purchaser shall be allowed to sell such goods under reservation in the regular course of business; pledging or assignment as security shall not be permitted, though. Should his contingent right (so-called Anwartschaftsrecht under German law) be pledged or assigned as security, Purchaser shall point out the retention of title by us to the secured third party and shall inform us without delay of such pledging or assignation as security. Purchaser shall do his best to secure our rights in the course of any resale on credit; he shall in particular pass on the retention of title.

10.3 Purchaser herewith already assigns to us his claims for the purchase price from the resale of the goods under reservation up to the amount of our agreed purchase price; we herewith accept this assignment. Should the conditions of a resale exclude such assignment, such resale of the goods under reservation shall require our prior, express and written consent. The same shall apply, if Purchaser's claim were to be entered into an open or run-

ning account. Irrespective of a resale taking place unlawfully or with our consent, Purchaser herewith already assigns to us his claim for any credit balance in the amount of the invoice; we herewith accept this assignment.

10.4 Irrespective of the above assignment and our right of direct collection, Purchaser shall remain entitled to collect the debt under a resale in his own name, as long as he duly fulfils his obligations towards us and does not become insolvent. Upon a significant deterioration of Purchaser`s financial situation – especially in connection with a formal application to start insolvency proceedings – his collection right shall cease, though.

Upon our request, Purchaser shall give us all information necessary for collection of the debt and shall further inform the debtor of the assignment. We may at any time, even if Purchaser is entitled to collection, request from Purchaser a signed declaration of assignation.

10.5 Any eventual processing or further fabrication of the goods under reservation shall be performed in our name and for our benefit, without resulting in any obligation for us, though. In the event of any further fabrication, adjunction, mixture or combination of the goods under reservation with other goods not belonging us, such proportion of the resulting joint ownership in the new product(s) shall vest in us, which is the equivalent to the ratio of the invoiced value of the goods under reservation to the value of the other goods at the time of the further fabrication, adjunction, mixture or combination.

The Parties to this contract agree that in the event, Purchaser acquires sole ownership of such new product(s), Purchaser passes such proportion of this property right onto us as is the equivalent to the ratio of the invoiced value of the goods under reservation to the value of the other goods at the time of the further fabrication, adjunction, mixture or combination; further that Purchaser stores such new product(s) for us free of charge.

In the event of a resale of the goods under reservation, after or without prior further fabrication, adjunction, mixture or combination, the above anticipatory assignment shall only survive in the amount of our invoice regarding the goods under reservation, which are being sold on together with other goods.

10.6 Purchaser shall insure the goods under reservation against all typical risks, in particular against theft, fire and water damage, until he acquires full ownership; Further on, Purchaser shall, on demand, provide us with a confirmation by an insurance company to that effect. Purchaser shall treat the goods under reservation with due care.

10.7 We shall, at Purchaser's request, release securities, if and insofar as the realistic achievable value of the securities provided by Purchaser exceeds the sum of the claims to be secured by more than 10 percent. We shall be free to choose the security to be released.

10.8 Purchaser shall secure and safeguard our title, right of disposal and property rights in accordance with the law and the statutory provisions of the country of destination of the goods under reservation and shall cooperate for that purpose in any way necessary. Should binding local law deny the validity of our retention of title, Purchaser shall, on our request, provide us with other security of equally lasting value.

10.9 The retention of title shall not affect the passing of risk pursuant to Cipher 7.

11 Provision of material and documents

Should Purchaser provide parts or material for the production or for other usage in the context of the processing of the order, we shall not inspect such provided material upon receipt for hidden defects, unless otherwise expressly agreed upon. Should such provided material prove unfit for any agreed processing or further fabrication, we shall inform Purchaser. It shall be Purchaser's responsibility to provide a solution. Possible additional costs shall be borne by Purchaser. Delivery times shall be deemed to be adequately prolonged or postponed.

12 Guaranties

12.1 References to technical standards or other sources of the acknowledged rules of technology shall solemnly serve to describe the relevant goods, but shall not constitute a guarantee. A binding guarantee requires an express stipulation to that effect or an express representation to that effect in advertising or sale material. Under such guaranty we shall be liable as follows:

  a. Guaranties shall only refer to the absence of defects of the material.

  b. Liability under a guaranty shall require as a precondition the professional installation and operation in accordance with the operating instructions, security instructions or data sheet as well as an appropriate use of the goods.

  c. Liability under a guaranty shall be limited to the replacement of the defective goods free of charge; any additional costs and expenses or any damages shall not be borne or covered for by us.

  d. The statutory period of limitation shall start with the passing of the risk.

13 Industrial Property Rights and Copyright

13.1 Cipher 1.3 applies. Purchaser shall acknowledge and observe our industrial property rights, in particular but not limited to all titles to and ownership of domains, names, trade marks and other rights and brands and including title to and ownership of know-how. Each and any use of such shall only and exclusively be made in connection with our products and shall be limited to the use customary in the particular trade and in compliance with the applicable protective laws.

13.2 Regarding deliveries of customized products ordered with specifications deviating from our product catalogue, Purchaser shall solemnly be responsible and liable for the protection of any industrial property rights of third parties. Should a third party assert claims against us based on an infringement of industrial property rights (including claims for compensation for legal costs) with respect to such an order, Purchaser shall indemnify us from such claim including on a full solicitor and client costs basis upon our request and shall assist us in such matter to the best of his ability.

14 Severability Clause

Should any provision of these GTSD or any other provision within the framework of other contracts between the parties be or become invalid, illegal, void or unenforceable, the validity of all other provisions and contracts shall remain unaffected. The Parties shall agree to replace such invalid, illegal, void or unenforceable provision by such valid provision reflecting the originally intended economic aim as closely as possible.

15 Place of Performance, Jurisdiction and Venue, Applicable Law

15.1 Contractual place of performance regarding all and any deliveries and payments shall be our main seat of business at Urbach, Germany.

15.2 Venue for any and all disputes arising directly or indirectly out of or in connection with this contractual relationship shall be our main seat of business at Urbach, Germany. We shall be entitled, though, to also initiate legal proceedings against Purchaser at the courts having jurisdiction over Purchaser under the laws applicable at his place of business.

15.3 The entire agreement, its terms and all legal relationships between the Parties shall be solemnly and exclusively governed by and construed in accordance with the law of the Federal Republic of Germany. The United Nations Convention On Contracts For The International Sale Of Goods (CISG) and the German Law on Conflicts of Law shall not be applicable.

15.4 Any and all modifications or amendments of this contract shall only be effective, if confirmed by us in writing; this shall also apply to this requirement of written form itself.

SOLUTIONS. CLEVER. PRACTICAL.

di-soric Headquarters

Germany: di-soric GmbH & Co. KG | Steinbeisstrasse 6 | 73660 Urbach
Phone +49 71 81 98 79-0 | Fax +49 71 81 98 79-179 | info@di-soric.com

di-soric Subsidiaries

Austria: di-soric GmbH & Co. KG | Phone +43 7228 72 366 | info.at@di-soric.com
China: di-soric Industrial Automation (Suzhou) Co. Ltd. | Phone +86 512 6260 9518 | info@di-soric.cn
France: di-soric SAS | Phone +33 476 61 65 90 | info.fr@di-soric.com
Singapore: di-soric Pte. Ltd. | Phone +65 6694 7866 | info.sg@di-soric.com
The Netherlands: di-soric B. V. | Phone +31 413 33 13 91 | info.nl@di-soric.com

For further information visit www.di-soric.com/international   **www.di-soric.com**