

nVISION-i
BEDIENUNGS-ANLEITUNG

di-soric

nVision-i Bedienungsanleitung

Release 26.1

di-soric GmbH + Co. KG

24.06.2026

Inhaltsverzeichnis

1 Einleitung **3**
- 1.1 Programmausgaben . . . 3
- 1.2 Anforderungen . . . 4
- 1.3 Lizenzierung . . . 4
  - 1.3.1 nVision-i . . . 5
- 1.4 Installation . . . 6
- 1.5 Konfiguration . . . 9
- 1.6 Firmware Update . . . 12
- 1.7 Reset . . . 13

2 nVison-i Benutzen **15**
- 2.1 Der Hauptbildschirm von **nVision-i** . . . 15
- 2.2 Jobs und Werkzeuge . . . 18
- 2.3 Eingangsdaten . . . 18
  - 2.3.1 Verknüpfen von Digitaleingängen . . . 19
  - 2.3.2 Verknüpfen von Feldbus Eingängen . . . 20
  - 2.3.3 Verknüpfen von TCP/IP Kommando Eingängen . . . 25
- 2.4 Ausgangsdaten . . . 27
  - 2.4.1 Digitale Ausgänge produzieren . . . 27
  - 2.4.2 Feldbus Ausgänge produzieren . . . 30
  - 2.4.3 TCP/IP Variablen Ausgänge produzieren . . . 34

3 Machine Vision Methoden **37**
- 3.1 Mustererkennung . . . 37
- 3.2 Korrektur der Schattierung . . . 38
- 3.3 Korrektur der geometrischen Verzerrung . . . 39
- 3.4 Blob Analyse . . . 42
  - 3.4.1 Fläche . . . 43
  - 3.4.2 Winkel . . . 44
  - 3.4.3 Schwerpunkt . . . 44
  - 3.4.4 Oberseite . . . 45
  - 3.4.5 Rechte Seite . . . 46
  - 3.4.6 Unterseite . . . 47

3.4.7 Linke Seite . . . 48
3.4.8 Geometrische Mitte . . . 49
3.4.9 Höhe . . . 50
3.4.10 Breite . . . 51
3.4.11 Umfang . . . 52
3.4.12 Fläche der Löcher . . . 53
3.4.13 Konvexe Fläche . . . 54
3.4.14 Konvexer Umfang . . . 55
3.4.15 Faserlänge . . . 56
3.4.16 Faserbreite . . . 56
3.4.17 Hauptachsenlänge . . . 56
3.4.18 Nebenachsenlänge . . . 57
3.4.19 Seitenverhältnis . . . 57
3.4.20 Zirkularität . . . 57
3.4.21 Kompaktheit . . . 57
3.4.22 Konvexität . . . 57
3.4.23 Anzahl der Löcher . . . 57
3.4.24 Perforation . . . 58
3.4.25 Rauheit . . . 58
3.4.26 Rundheit . . . 58
3.4.27 Sphärizität . . . 58
3.5 Strichcode-Dekodierung . . . 58
3.5.1 Linear . . . 59
3.5.2 Zwei-dimensional . . . 59
3.5.3 Dekoder Merkmale . . . 60
3.5.4 Direkt-markierte Codes . . . 61
3.5.5 Verifikation . . . 61
3.5.6 GS1 Application Identifier . . . 62
3.6 Optische Zeichenerkennung (OCR) . . . 63
3.6.1 `Document` . . . 64
3.6.2 `DotPrint` . . . 64
3.6.3 `HandWritten` . . . 64
3.6.4 `Industrial` . . . 65
3.6.5 `OCR-A` . . . 65
3.6.6 `OCR-B` . . . 65
3.6.7 `Pharma` . . . 66
3.6.8 `SEMI` . . . 66
3.6.9 `Universal` . . . 67

4 Referenz **69**
4.1 Grafische Oberfläche . . . 69
4.1.1 Ribbon . . . 69
4.1.2 Statuszeile . . . 73
4.1.3 Job Testen . . . 73
4.2 Korrektur-Werkzeuge . . . 74
4.2.1 Schattierung . . . 74
4.2.2 Verzerrungs-Korrektur . . . 77
4.3 Werkzeuge . . . 80

4.3.1 Akquisition - Bild . . . 80
4.3.2 Akquisition - Digitale Eingänge . . . 82
4.3.3 Akquisition - Feldbus Eingänge . . . 84
4.3.4 Akquisition - TCP/IP Kommando Eingänge . . . 90
4.3.5 Flächenpixel Erkennen . . . 93
4.3.6 Flächen Zählen . . . 95
4.3.7 Zeichen Zählen . . . 97
4.3.8 Kantenpixel Erkennen . . . 105
4.3.9 Kanten Zählen . . . 107
4.3.10 Formen Zählen . . . 108
4.3.11 Barcodes Zählen . . . 111
4.3.12 Barcode Erkennen . . . 114
4.3.13 Text Erkennen . . . 119
4.3.14 Fläche Lokalisieren . . . 128
4.3.15 Barcode Lokalisieren . . . 131
4.3.16 Kante Lokalisieren . . . 135
4.3.17 Form Lokalisieren . . . 138
4.3.18 Text Lokalisieren . . . 140
4.3.19 OCV . . . 149
4.3.20 Winkel Messen . . . 158
4.3.21 Helligkeit Erkennen . . . 161
4.3.22 Kreis Messen . . . 163
4.3.23 Kontrast Erkennen . . . 165
4.3.24 Distanz Messen . . . 167
4.3.25 Punkt zu Linie Distanz Messen . . . 169
4.3.26 Punkt zu Punkt Distanz Messen . . . 170
4.3.27 Logik - Resultate . . . 171
4.3.28 Logik - Digitale Ausgänge . . . 173
4.3.29 Logik - Feldbus Ausgänge . . . 175
4.3.30 Logik - TCP/IP Kommando Ausgänge . . . 179
4.3.31 Logik - FTP . . . 180
4.3.32 Smart Tool . . . 182
4.4 Grafische Programmierung . . . 182
4.4.1 Grafische Programmierung . . . 182
4.4.2 Grafischer Editor . . . 185
4.5 Knoten . . . 191
4.5.1 Accumulate . . . 191
4.5.2 Acos . . . 191
4.5.3 Add . . . 192
4.5.4 Advanced Result . . . 193
4.5.5 All Ok . . . 193
4.5.6 And . . . 194
4.5.7 Angle . . . 194
4.5.8 Angle Between Lines . . . 195
4.5.9 Asin . . . 195
4.5.10 Atan . . . 196
4.5.11 BinarySelector . . . 196
4.5.12 Boolean . . . 197

4.5.13 Brush . . . 197
4.5.14 Byte . . . 198
4.5.15 Center of Points . . . 199
4.5.16 Circle . . . 199
4.5.17 Circle Display . . . 200
4.5.18 Circle with Circle . . . 201
4.5.19 Comment . . . 201
4.5.20 Compare . . . 201
4.5.21 Compare . . . 202
4.5.22 Concatenate . . . 203
4.5.23 Concatenate . . . 204
4.5.24 Contains . . . 205
4.5.25 Cos . . . 205
4.5.26 Degrees -> Radians . . . 206
4.5.27 Distinct . . . 206
4.5.28 Divide . . . 207
4.5.29 Double . . . 208
4.5.30 Except . . . 208
4.5.31 FallbackValue . . . 209
4.5.32 First . . . 209
4.5.33 Float . . . 210
4.5.34 Format Number . . . 210
4.5.35 Image Upload . . . 211
4.5.36 In . . . 212
4.5.37 Intersect . . . 212
4.5.38 Int16 . . . 213
4.5.39 Int32 . . . 213
4.5.40 Int64 . . . 214
4.5.41 IsNull . . . 214
4.5.42 Item . . . 215
4.5.43 Join . . . 215
4.5.44 Last . . . 216
4.5.45 Line . . . 216
4.5.46 Line . . . 217
4.5.47 Line Display . . . 218
4.5.48 Line with Circle . . . 218
4.5.49 Line with Line . . . 219
4.5.50 Link . . . 219
4.5.51 Make List . . . 220
4.5.52 Multiply . . . 220
4.5.53 Not . . . 221
4.5.54 Or . . . 222
4.5.55 Out . . . 223
4.5.56 Pad Left . . . 223
4.5.57 Pad Right . . . 224
4.5.58 Parse GS1 AI . . . 225
4.5.59 PLC . . . 225
4.5.60 Point . . . 226

4.5.61 Point Display . . . 226
4.5.62 Point to Circle . . . 227
4.5.63 Point to Line . . . 228
4.5.64 Point to Point . . . 228
4.5.65 Pose Result . . . 228
4.5.66 Primary Result . . . 229
4.5.67 Radians -> Degrees . . . 230
4.5.68 Read Boolean . . . 230
4.5.69 Read Boolean . . . 231
4.5.70 Read Boolean . . . 232
4.5.71 Read Byte . . . 234
4.5.72 Read Byte . . . 234
4.5.73 Read Byte . . . 235
4.5.74 Read Float . . . 236
4.5.75 Read Float . . . 237
4.5.76 Read Float . . . 239
4.5.77 Read Int32 . . . 240
4.5.78 Read Int32 . . . 241
4.5.79 Read Int32 . . . 242
4.5.80 Read String . . . 243
4.5.81 Read String . . . 244
4.5.82 Read String . . . 246
4.5.83 Reverse . . . 248
4.5.84 Rectangle Display . . . 248
4.5.85 Replace . . . 249
4.5.86 Round . . . 250
4.5.87 Secondary Result . . . 251
4.5.88 Select GS1 AI . . . 252
4.5.89 Server . . . 253
4.5.90 Sin . . . 253
4.5.91 Skip . . . 254
4.5.92 Sort . . . 254
4.5.93 Split . . . 255
4.5.94 String . . . 256
4.5.95 StateReader . . . 256
4.5.96 StateWriter . . . 257
4.5.97 Subtract . . . 257
4.5.98 Substring . . . 258
4.5.99 Take . . . 259
4.5.100 Tan . . . 260
4.5.101 TCP/IP Output . . . 260
4.5.102 TCP/IP Variables . . . 261
4.5.103 TCP/IP Variables . . . 263
4.5.104 Text Display . . . 264
4.5.105 Text Upload . . . 265
4.5.106 Tool . . . 266
4.5.107 Tool . . . 266
4.5.108 Transform . . . 267

4.5.109 Trim . . . 267
4.5.110 Trim Left . . . 268
4.5.111 Trim Right . . . 268
4.5.112 Truncate . . . 269
4.5.113 UI Circle . . . 269
4.5.114 UI Line . . . 270
4.5.115 UI Point . . . 270
4.5.116 UI Rectangle . . . 270
4.5.117 Union . . . 271
4.5.118 Within Range . . . 271
4.5.119 Write Boolean . . . 272
4.5.120 Write Boolean . . . 273
4.5.121 Write Byte . . . 273
4.5.122 Write Byte . . . 274
4.5.123 Write Float . . . 275
4.5.124 Write Float . . . 276
4.5.125 Write Int32 . . . 277
4.5.126 Write Int32 . . . 277
4.5.127 Write String . . . 278
4.5.128 Write String . . . 279
4.5.129 Xor . . . 280
4.6 Protokolle . . . 281
4.6.1 UDP . . . 281
4.6.2 TCP . . . 282
4.6.3 SSH . . . 282
4.6.4 FTP/SFTP . . . 282
4.6.5 Telnet . . . 283
4.6.6 TCP/IP Kommando-Server . . . 283
4.6.7 TCP/IP-Streams . . . 291
4.6.8 PROFINET . . . 292
4.6.9 EtherNet/IP . . . 295
4.6.10 Standard Status- und Kontroll Signale für PROFINET oder EtherNet/IP 298
4.6.11 HTML Webserver . . . 302
4.6.12 ReST API . . . 305
4.6.13 Job-Wechsel mit den digitalen Eingängen . . . 309
4.7 Hardware . . . 309
4.7.1 Kabel Pinbelegung . . . 309
4.7.2 LED . . . 311

5 Credits 313
5.1 Boost . . . 313
5.2 Bootstrap . . . 313
5.3 bzip2 . . . 313
5.4 CommandLine . . . 313
5.5 Debian Linux . . . 314
5.6 Dynamic Expresso . . . 314
5.7 Eigen . . . 314
5.8 Extended WPF Toolkit . . . 314

5.9 FontAwesome Fonts . . . 315
5.10 FreeImage . . . 315
5.11 IndependentJPEGGroup . . . 315
5.12 INI File Parser . . . 315
5.13 IronPython . . . 315
5.14 Json.NET . . . 315
5.15 KBCsv . . . 315
5.16 libjpeg-turbo . . . 316
5.17 libpng . . . 316
5.18 libsimdpp . . . 316
5.19 Linux kernel . . . 316
5.20 MessagePack-CSharp . . . 316
5.21 NetworkView . . . 316
5.22 NLog . . . 316
5.23 OxyPlot . . . 316
5.24 Microsoft.Diagnostics.Tracing.TraceEvent . . . 317
5.25 Prism . . . 317
5.26 signalr.min . . . 317
5.27 System.Commandline . . . 317
5.28 Sphinx RTD Theme . . . 317
5.29 SSH.NET . . . 317
5.30 The Helper Trinity . . . 317
5.31 Threading Building Blocks . . . 318
5.32 VTK . . . 318
5.33 WpfLocalizeExtension . . . 318
5.34 WPF Task Dialog Wrapper . . . 318
5.35 zlib . . . 318

nVision-i Benutzerhandbuch
Release 26.1.3553

Copyright © 2024 di-soric GmbH & Co. KG - Alle Rechte vorbehalten.

Der Inhalt dieses Dokuments ist geschützt und dient ausschließlich dazu, Kunden zur Bedienung und/oder Wartung von Geräten der di-soric GmbH & Co. KG anzuleiten.

Die darüber hinaus gehende Verwendung, die Vervielfältigung oder die Speicherung, auch auszugsweise, ist nur mit vorheriger in schriftlicher Form vorliegender Genehmigung seitens der di-soric GmbH & Co. KG zulässig.

Haftungsausschluss:
Für Druckfehler und Irrtümer, die bei der Erstellung des Dokumentes unterlaufen sind, ist jede Haftung ausgeschlossen. Die Informationen und Angaben in diesem Dokument können jederzeit geändert werden.

di-soric GmbH & Co. KG
Steinbeisstraße 6
73660 Urbach

Deutschland/Europäische Union

http://www.di-soric.com

KAPITEL 1

Einleitung

nVision-i ist die Entwicklungs-Software von di-soric für Vision Sensoren.

Derzeit gibt es zwei Produktlinien:

- der **ID-600 / ID-620 / ID-520** ist ein stationärer Codeleser,
- der **CS-60 / CS-62 / CS-52** ist ein Allzweck Vision Sensor.

nVision-i bietet wertvolle und leistungsfähige Werkzeuge für Bildverarbeitung und Inspektion. Es wird verwendet, um Lösungen auf einem di-soric Vision Sensor zu konfigurieren.

nVision-i hilft bei einer Vielzahl von Vision Sensor basierten Aufgaben der Industrie:

- Positionsprüfung, um Handling-Systeme zu leiten oder ein Werkzeug in die richtige Position zu fahren.
- Identifikation eines Teils basierend auf Markierungen, seiner Form oder anderen visuellen Aspekten.
- Prüfung eines Teils auf korrekte Produktion oder korrekten Zusammenbau.
- Messen der Teiledimension.
- Inspektion eines Teils auf Defekte.

1.1 Programmausgaben

nVision-i beseteht aus zwei Hauptkomponenten: der **nVision-i Designer** ist zuständig für die Applikations-Entwicklung und die **nVision-i Runtime** ist der Laufzeit Teil.

Der **nVision-i Designer** stellt ein graphisches Benutzerinterface und ein graphisches Programmiersystem bereit.

Die **nVision-i Runtime** stellt die Bildverarbeitungsfunktionalität auf modulare Weise bereit und ermöglicht die Ausführung graphischer Programme. Die **nVision-i Runtime** kommt in zwei Versionen:

- in der Ausführung für den **CS-6x/ID-6x0** Vision Sensor
- in der Ausführung für den PC als **CS-6x/ID-6x0** Emulator

Mit dem **nVision-i Designer** erstellen Sie die Applikationen. Der **nVision-i Designer** kommuniziert mit einem Vision Sensor (oder einem Emulator auf einem PC), auf dem die Applikation ausgeführt wird.

Der Vision Sensor hat Funktionen für Bildverarbeitung, Bildanalyse, Blobanalyse, Identifikation, Vermessung und Muster-Vergleich.

1.2 Anforderungen

Der **nVision-i Designer** läuft auf Windows 10 oder höher. Der **nVision-i Designer** ist eine 64 Bit Applikation.

Die **nVision-i Runtime** läuft sowohl auf dem Vision Sensor als auch auf einem PC im Emulator Modus.

System

1 GHz (Giga-Hertz) oder schneller, zwei oder mehr Kerne, 64 bit Prozessor oder SoC (system on a chip)

Speicher

RAM 4 GB (Giga-Bytes) oder mehr

Festplatte

64 GB (Giga-Bytes) oder mehr verfügbarer Plattenspeicher

Anzeige

HD (high-definition, 720p) Anzeige, 9” Monitor oder größer, 8 bit pro Farbkanal

Betriebssystem

Microsoft Windows 10 oder 11

1.3 Lizenzierung

Der **nVision-i Designer** arbeitet in Verbindung mit einem Vision Sensor. Sowohl der **nVision-i Designer** als auch der Vision Sensor verwenden lizenzierte Softwarekomponenten. Die Lizenz von **nVision-i** wird kostenlos zur Verfügung gestellt. Lizenzschlüssel sind eindeutig und jeweils nur einmalig verwendbar.

1.3.1 nVision-i

Falls noch keine frühere Version installiert und lizenziert wurde, wird beim ersten Start von nVision-i die Lizenz aktiviert.

Wenn die Lizenzierung erfolgreich war, kann nVision-i gestartet werden. Im anderen Fall wird eine Fehlermeldung angezeigt.

Falls Sie keinen Internetzugang haben oder die Firewall-Einstellungen eine Verbindung verhindern kann die Software telefonisch aktiviert werden. In diesen Fällen empfehlen wir Ihnen, sich an unseren Kundenservice wenden - am besten schicken Sie einen Screenshot des Lizenzaktivierungsdialogs an den Kundenservice. Der Kundenservice wird Ihnen Anweisungen und Daten für die manuelle Lizenzaktivierung geben.

Nach erfolgreicher Aktivierung kann nVision-i gestartet werden.

Möglicherweise erhalten Sie beim Start von nVision-i eine Meldung der Windows Firewall.

Sie müssen den Zugriff über die Firewall zulassen, damit nVision-i mit den **CS-6x/ID-6x0** Sensoren kommunizieren kann.

CS-6x/ID-6x0 Werkzeuge

Es stehen mehrere Werkzeug-Sätze zur Verfügung. Nicht alle Werkzeuge sind auf allen Sensor-Modellen verfügbar.

Der Werkzeugsatz **Basis** enthält die Lokalisierungswerkzeuge **Fläche lokalisieren**, **Kante lokalisieren** und **Form lokalisieren**, die Erkennungswerkzeuge **Helligkeit erkennen**, **Kontrast erkennen** und **Flächenpixel erkennen** und die Zählwerkzeuge **Flächen zählen**, **Kanten zählen** und **Formen zählen**.

Optionale Werkzeugsätze sind die **Messen** und **ID** Sätze und die **ID-Pro** Erweiterung.

Der Satz **Messen** enthält die Messwerkzeuge **Winkel messen**, **Kreis messen**, **Distanz messen**, **Punkt zu Punkt messen** und **Punkt zu Linie messen**.

Der Satz **ID** enthält die Barcode-Werkzeuge **Barcode lokalisieren**, **Barcode erkennen** und **Barcodes zählen**.

Die **ID-Pro** Erweiterung bietet erweiterte und verbesserte Erkennung von **DPM** (Direct Part Marking) Barcodes sowie **Verifizierzung** für 2D Codes.

Lizenzverwaltung

Die aktuelle Lizenz eines **CS-6x/ID-6x0** Vision Sensor kann mit der Software **nVision-i** angezeigt werden.

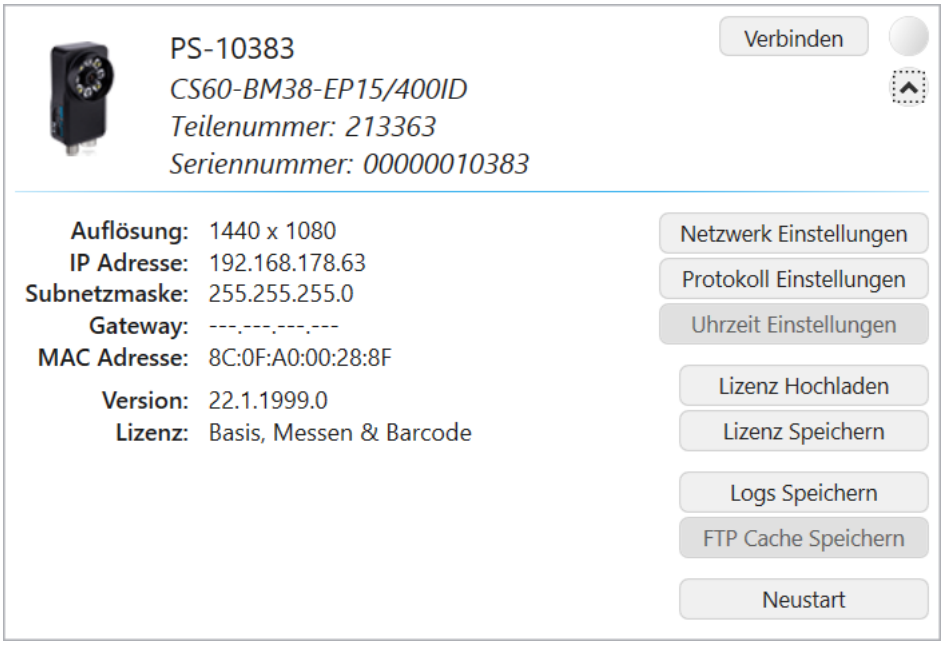


Abb. 1: Erweiterte Informationen zum **CS-6x/ID-6x0** Vision Sensor.

Mit der Taste **Lizenz Hochladen** kann eine Lizenzdatei auf den Sensor hochgeladen werden.

Mit der Taste **Lizenz Speichern** kann eine Lizenzdatei vom Sensor heruntergeladen werden und auf dem PC gespeichert werden (hilfreich bei der Bestellung einer Upgrade Lizenz).

Eine Lizenz ist an die Seriennummer eines **CS-6x/ID-6x0** Vision Sensor gebunden.

Hier ist ein Beispiel für den Inhalt einer Lizenzdatei:

```
<?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>
```

Alle drei Werte sind verschlüsselt. Der **LicenseKey** enthält die Freischaltung der einzelnen Module und Werkzeuge, der **HardwareId** enthält die verschlüsselte Seriennummer und der **LicenseActivationKey** enthält den Freischaltcode des Lizenzservers. Alle drei Schlüssel müssen zusammenpassen, damit die Lizenz gültig ist. Durch die Abhängigkeit der Lizenz von der Seriennummer kann eine Lizenz nicht zwischen Sensoren übertragen werden.

1.4 Installation

Das **nVision-i** Installationsprogramm wird in einer 64 Bit version geliefert. Das Installationsprogramm prüft die nötigen Voraussetzungen (VC++ Runtime), installiert sie nach, falls sie fehlen, und installiert auch die **nVision-i** Software selbst.

Auf der Begrüßungsseite des Installationsprogramms lesen Sie die Lizenzbedingungen, bestätigen Sie "Ich stimme den Lizenzbestimmungen zu" und drücken Sie auf **Installieren**, um die Installation zu starten.

Das Installationsprogramm kopiert die nötigen Dateien in das Installationsverzeichnis (`%PROGRAM_FILES%\di-soric\nVision-i\`).

Auf der Bestätigungsseite der erfolgreichen Installation müssen Sie möglicherweise den PC neu starten.

Sie können mehrere Versionen von **nVision-i** nebeneinander installieren. Die Lizenzinformationen bleiben erhalten und gültig für alle installierten Versionen.

Optional können Sie auf der Begrüßungsseite des Installationsprogramms auf **Optionen** klicken, um das Installationsverzeichnis zu wechseln.

Das Installationsprogramm ist mehrsprachig und unterstützt Deutsch, Englisch, Spanisch, Französisch, Italienisch, Portugiesisch, Koreanisch und Chinesisch. Wenn das Installationsprogramm ohne Option ausgeführt wird, versucht es, die für den Benutzer oder das System eingestellte Sprache zu verwenden. Falls es sich um eine nicht unterstützte Sprache handelt, wird Englisch verwendet. Die Sprache kann auch über die Kommandozeilen-Option `-lang xxxx` ausgewählt werden, z.B.

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

Die folgenden Länder und Sprachen werden unterstützt:

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
1046	pt-BR	Portuguese - Brazil
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
2070	pt-PT	Portuguese - Portugal
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

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Id	BCP 47 Code	Beschreibung
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
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

Fortsetzung auf der nächsten Seite

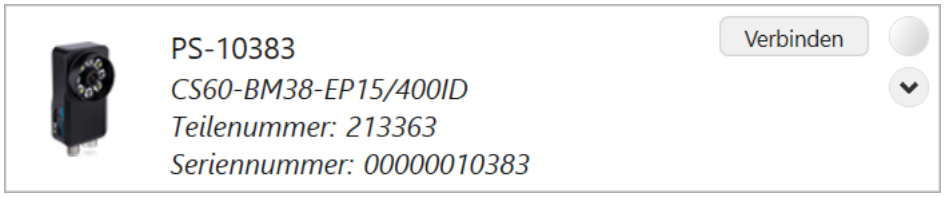
Tab. 1 – Fortsetzung der vorherigen Seite

Id	BCP 47 Code	Beschreibung
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 Konfiguration

CS-6x/ID-6x0 Sensoren werden mit voreingestellter IP-Adresse geliefert: `192.168.3.15`.

Die Netzwerk- und Feldbus-Konfiguration der Sensoren kann von nVision-i aus erfolgen.



Drücken sie die Pfeil-Taste um mehr Informationen zum ausgewählten Gerät anzuzeigen, wie Kamera-Auflösung, Ethernet Konfiguration, Industrieprotokoll-Einstellungen und Lizenzinformation. Wenn ein Sensor verfügbar ist, ist die LED auf der rechten Seite aus.

Klicken Sie auf **Verbinden**, um eine Verbindung zum Gerät herzustellen. Wenn ein Gerät verbunden ist, wechselt die Taste auf **Trennen**. Die LED wechselt auf blau um die Verbindung anzuzeigen.

Wenn ein Sensor mit einer anderen Instanz von nVision-i verbunden ist, leuchtet die LED in gelb, um anzuzeigen, dass keine Verbindung zu diesem Sensor möglich ist.

Oben rechts in der Geräte-Spalte sind zwei Tasten zum Starten des Emulators und zum Auffrischen der Geräteliste.

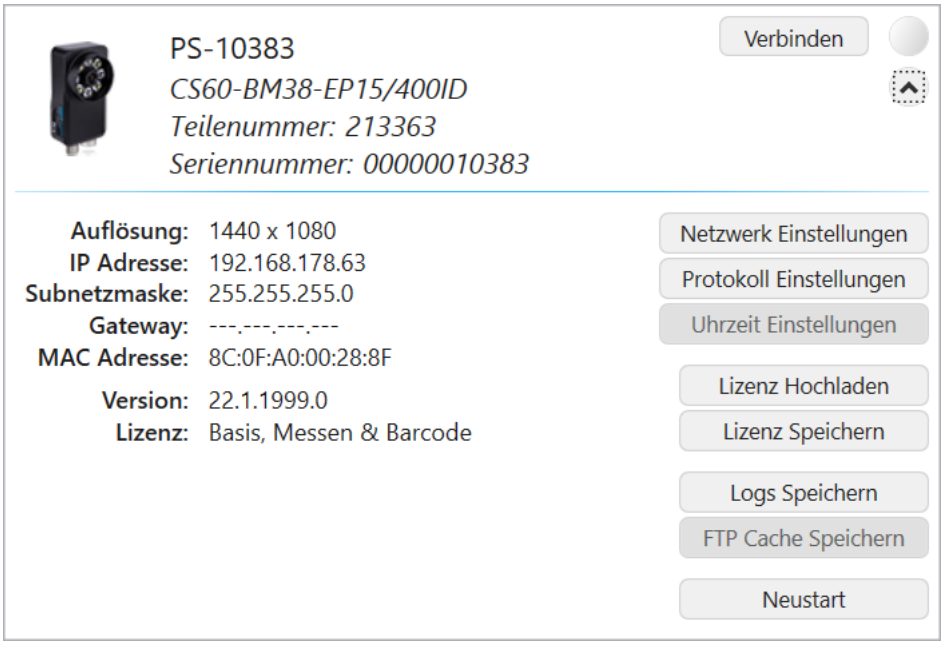


Abb. 2: **CS-6x/ID-6x0**, erweiterte Einstellungen

Klicken Sie auf die **Netzwerk Einstellungen** Taste, um die IP (Internet Protocol) Einstellungen des Geräts zu ändern:

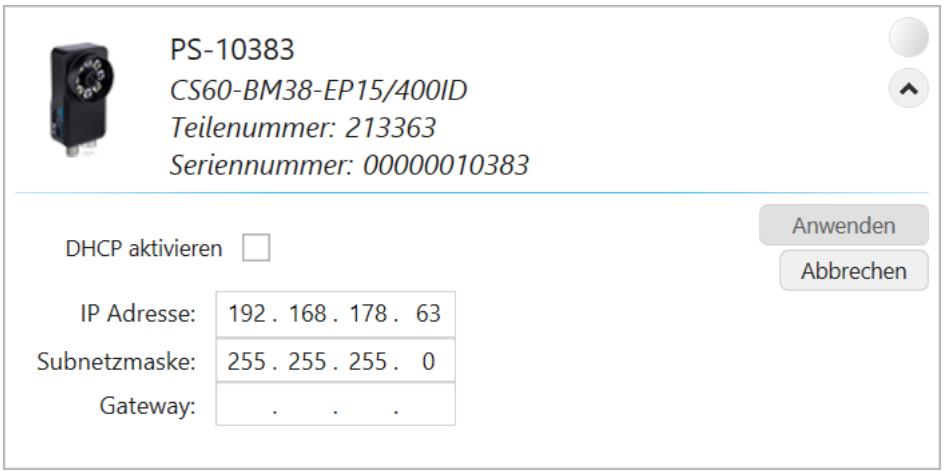


Abb. 3: **CS-6x/ID-6x0**, Netzwerk Einstellungen

DHCP aktivieren schaltet den DHCP Modus ein oder aus (das Gerät kann seine IP Adresse von einem DHCP Server im Netzwerk beziehen). Wenn das Gerät nicht im DHCP Modus ist, können **IP Adresse**, **Subnetzmaske** und **Gateway** geändert werden.

Anwenden wendet die Einstellungen an, mit **Abbrechen** verlassen Sie den Dialog ohne Änderungen.

Klicken Sie auf die Taste **Protokoll Einstellungen** um optionale Kommunikationsprotokolle zu konfigurieren.

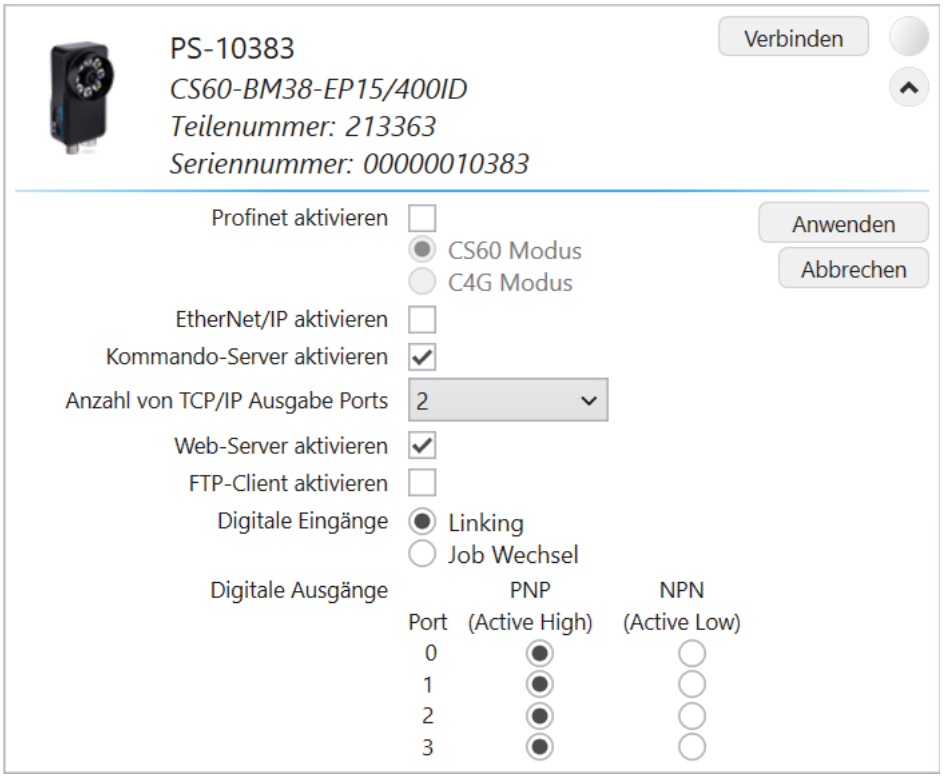


Abb. 4: **CS-6x/ID-6x0**, Protokoll Einstellungen

Profinet aktivieren schaltet das Profinet Subsystem des Sensors ein oder aus.

EtherNet/IP aktivieren schaltet das EtherNet/IP Subsystem des Sensors ein oder aus.

Web-Server aktivieren schaltet den eingebauten **CS-6x/ID-6x0** Web-Server ein oder aus.

FTP-Client aktivieren schaltet die FTP/SFTP Funktionalität ein oder aus. Die **FTP-Cache-Größe** gibt die Anzahl der Dateien (Bilder und Text) an, die maximal zwischengespeichtert werden können, wenn der FTP/SFTP-Server nicht verfügbar ist. Wenn die maximale Anzahl von Dateien erreicht ist, wird Platz geschaffen, indem zuerst die ältesten OK-Dateien und dann die ältesten NOK-Dateien gelöscht werden.

Die digitalen Eingänge können für den **Job Wechsel** oder für **Linking** verwendet werden.

Die digitalen Ausgänge können jeweils im **PNP (Active High)** oder **NPN (Active Low)** Modus betrieben werden.

Anwenden wendet die Einstellungen an, mit **Abbrechen** verlassen Sie den Dialog ohne Änderungen.

Klicken Sie auf die Taste **Uhrzeit Einstellungen** um die Uhr des **CS-6x/ID-6x0** zu stellen.

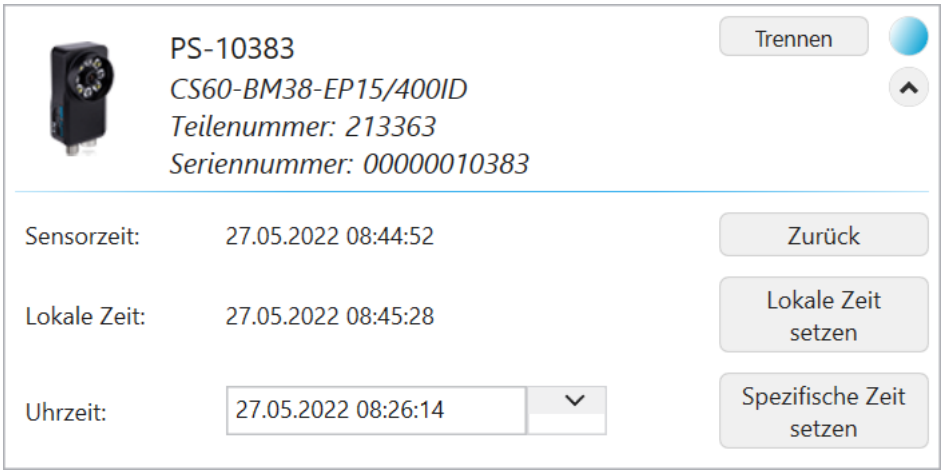


Abb. 5: **CS-6x/ID-6x0**, Uhrzeit Einstellungen

Lokale Zeit setzen stellt den **CS-6x/ID-6x0** auf die PC Zeit.

Spezifische Zeit setzen stellt den **CS-6x/ID-6x0** auf eine manuell eingegebene Zeit.

Mit **Zurück** verlassen Sie den Dialog.

Verwenden Sie die Schaltflächen **Lizenz hochladen** und **Lizenz speichern** um die Lizenzdateien auf **CS-6x/ID-6x0** hochzuladen oder von **CS-6x/ID-6x0** auf den PC herunterzuladen.

Mit der Taste **Logs Speichern** laden Sie die Fehler-Logs vom **CS-6x/ID-6x0** auf den PC herunter.

1.6 Firmware Update

Wenn die Firmware der Sensoren nicht passt, wird beim Verbinden eine Warnung angezeigt. Wenn die Sensor Firmware nicht aktualisiert wird ist eine Verbindung nicht möglich.

Bei einem Klick auf **Ändern auf Version** wird der Firmware Updater gestartet.

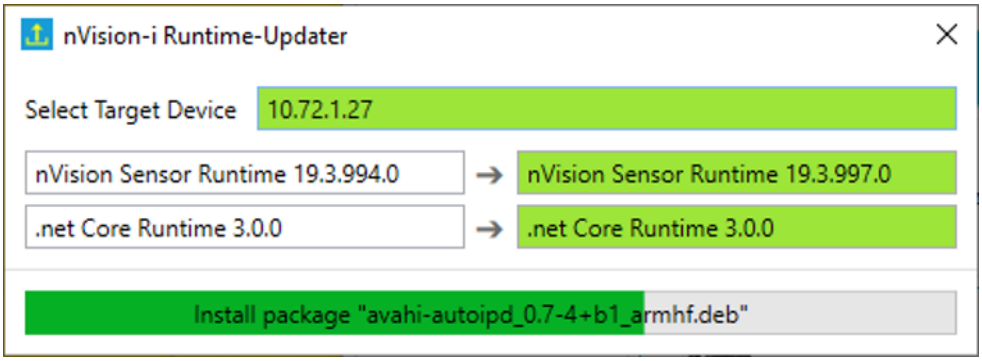


Abb. 6: Firmware aktualisieren.

Nach dem Aktualisieren der Firmware wird der Sensor neu gestartet. Der Sensor verschwindet kurz aus der Geräteliste, startet neu und erscheint wieder nach dem Start.

1.7 Reset

Falls der **CS-6x/ID-6x0** aus irgendeinem Grund nicht reagiert und nicht mehr in der nVision-i-Software gefunden und aufgelistet wird, können Sie die folgenden Schritte ausführen:

- Starten Sie den **CS-6x/ID-6x0** neu, indem Sie ihn von der Stromversorgung trennen, etwa 10 Sekunden warten und danach die Stromversorgung wieder herstellen.

Falls der **CS-6x/ID-6x0** nach einem Neustart über einen Aus-/Einschaltzyklus immer noch nicht sichtbar ist, können Sie versuchen, einen harten Reset durchzuführen.

- Trennen Sie den **CS-6x/ID-6x0** von der Stromversorgung. Die Reset-Taste befindet sich etwa bei 7 Uhr in der Objektivabdeckung. Halten Sie die Reset-Taste (mit einem dünnen Draht oder einer gebogenen Büroklammer) gedrückt, während Sie die Stromversorgung wieder anlegen. Halten Sie die Taste mindestens 30 Sekunden lang gedrückt, bis Sie als Bestätigung einen dreifachen Blitz sehen. Falls DHCP aktiviert ist, kann die Haltezeit auch über 2 Minuten oder noch länger betragen. Nicht bei allen Sensoren gibt es eine Reset-Taste.

Durch Reset werden die IP- und Kommunikationseinstellungen auf die Werkseinstellungen zurückgesetzt. Die Jobs bleiben jedoch erhalten.

- IP Adresse: 192.168.3.15
- Subnetzmaske: 255.255.255.0
- DHCP: aus
- Profinet: aus
- EtherNet/IP: aus
- Kommando-Server: aus
- Web-Server: aus
- FTP-Client: aus
- Digitale Eingänge: Job Wechsel
- Die Boot-Job Einstellung wird entfernt (nicht jedoch der Job selbst).

Alternativ kann der Sensor zurückgesetzt werden durch Klicken auf **Sensor Zurücksetzen**. Geben Sie die MAC Adresse des Sensors ein, der zurückgesetzt werden soll. Ein Zurücksetzen ist nur möglich, wenn der Sensor lokal über das Netzwerk erreichbar ist.

KAPITEL 2

nVison-i Benutzen

Nach dem Start von **nVision-i** wird zunächst der Hauptbildschirm angezeigt.

2.1 Der Hauptbildschirm von nVision-i

Der Hauptbildschirm listet die im Netzwerk gefundenen Geräte sowie die auf den Sensoren verfügbaren Jobs auf.

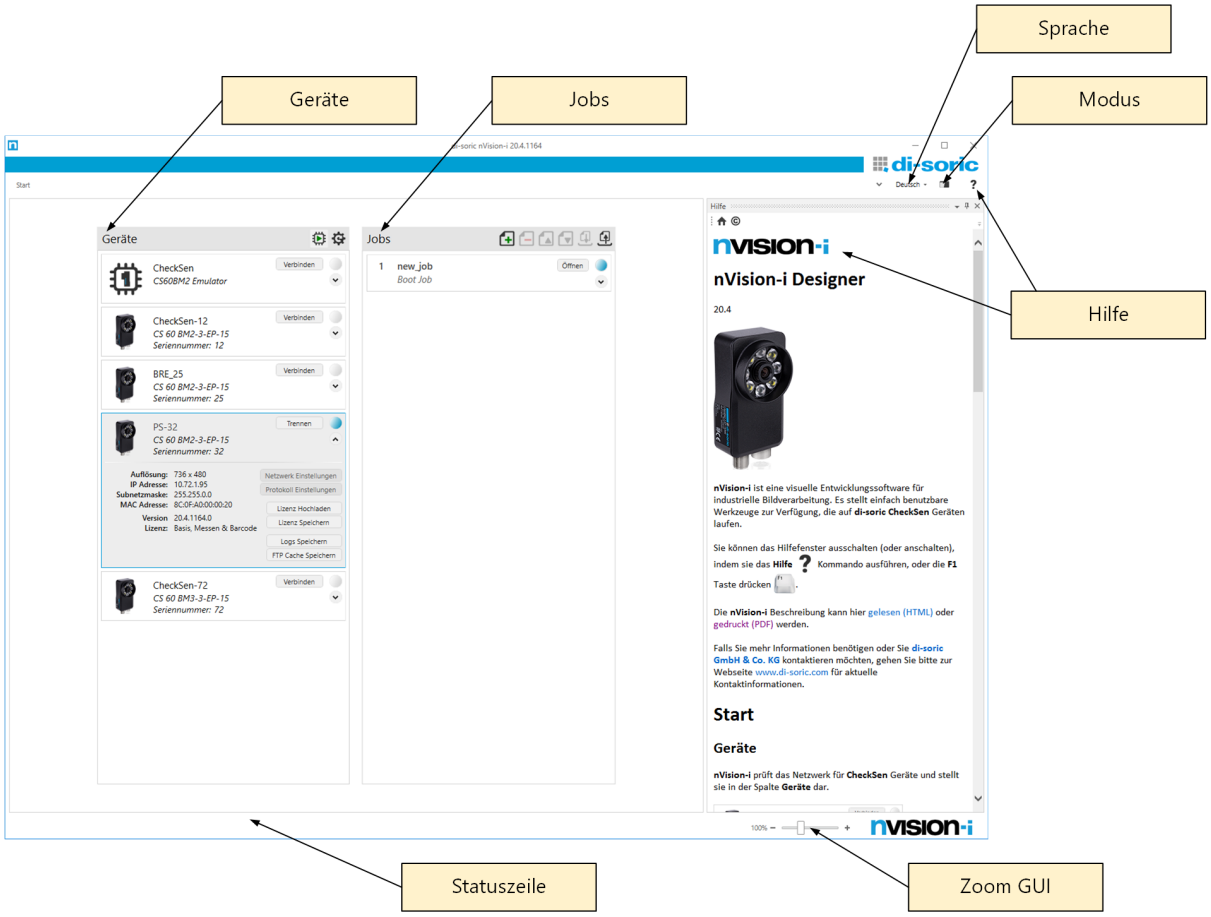


Abb. 1: Der nVision-i Hauptbildschirm.

nVision-i listet die verfügbaren **Geräte** auf, indem *Willkommensnachrichten* an das Netzwerk gesendet werden. Die Vision Sensoren senden Antworten an nVision-i zurück. Diese Antworten werden zum Erstellen der Geräteliste verwendet. Die *Willkommensnachrichten* werden regelmregelmäßig in Abständen von ca. 4 Sekunden gesendet, damit Geräte in der Liste **Geräte** aufgeführt werden, wenn sie verfügbar werden.

Gerät über Eingabe seiner MAC-Adresse zurücksetzen. Dabei werden die Netzwerk- und Protokolleinstellungen zurückgesetzt und die Bootjob-Einstellung entfernt.

\+ fügt ein Gerät unter Eingabe seiner IP-Adresse hinzu.

/ startet und stoppt den Emulator.

aktualisiert die Geräteliste.

Wenn eine Verbindung zu einem Vision Sensor hergestellt wurde, wird die Liste der verfügbaren Jobs in der **Jobs** Liste angezeigt. Jobs können in der Liste nach oben oder unten verschoben werden, was sich auf ihre Job-ID auswirkt, oder sie können auf andere Vision Sensor Geräte verschoben werden. Um Jobs zwischen Sensoren zu verschieben, muss nVision-i Verbindungen zu beiden Vision Sensor Geräten unterhalten.

fügt der Liste einen Job hinzu.

 entfernt einen Auftrag aus der Liste.

 verschiebt einen Job aufwärts auf einen anderen Platz.

 verschiebt einen Job abwärts auf einen anderen Platz.

 speichert einen Job auf die Festplatte.

 lädt einen Job von der Festplatte.

Rechts oben im nVision-i Fenster finden sich noch weitere Kommandos:

nVision-i ist in mehreren Sprachen verfügbar. Sie können die Sprache über das Menü **Sprachauswahl** wechseln.

Mit der Taste **Modusauswahl** können Sie zwischen drei verschiedenen Darstellungen wechseln: *Weiß*, *Hell* und *Dunkel*.

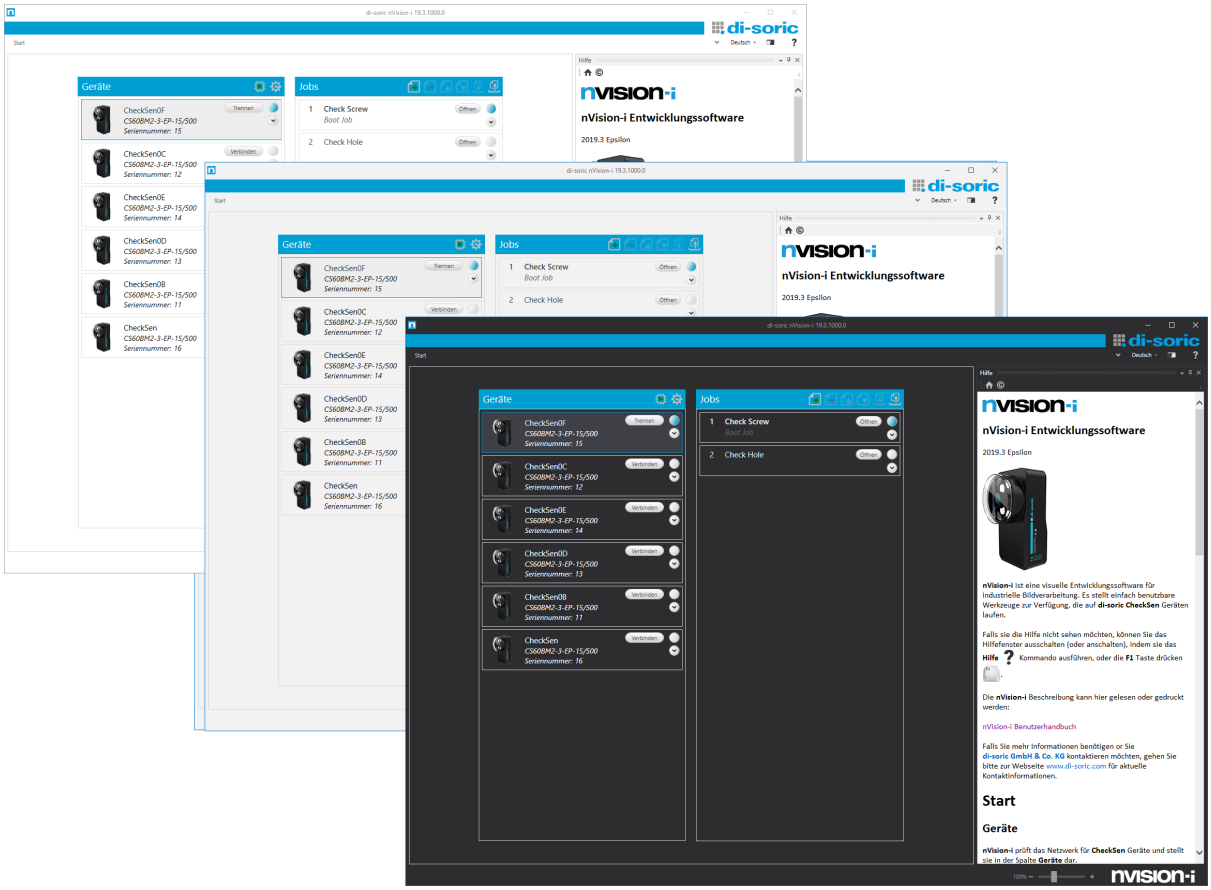


Abb. 2: Darstellung im *Weiß*-, *Hell*- und *Dunkel*-modus.

Mit der Schaltfläche **Hilfe** wird der Hilfebildschirm angezeigt und ausgeblendet.

In der **Statusleiste** können Sie - zusätzlich zu den angezeigten Informationen - die Benutzeroberfläche zoomen, beispielsweise für die Verwendung auf Touch-Geräten.

2.2 Jobs und Werkzeuge

Der Vision Sensor **CS-6x/ID-6x0** wird durch Erstellen von Jobs programmiert. Jobs werden schrittweise aufgebaut, indem ein Werkzeug nach dem anderen platziert wird. Jeder Job beginnt standardmäßig mit dem Werkzeug **Akquisition** und endet mit dem Werkzeug **Logik**. Dazwischen können andere Werkzeuge hinzugefügt werden, die das Verhalten des Jobs bestimmen.

Das Werkzeug **Akquisition** legt fest, wie das Bild und andere Eingänge (digitale Eingangsleitungen, Feldbus-Eingänge, TCP/IP-Variablen) erfasst werden.

Das Werkzeug **Logik** legt fest, wie die Ausgänge ermittelt und an die Außenwelt kommuniziert werden (digitale Ausgangsleitungen, Feldbus-Ausgänge, TCP/IP Variablen).

Neben den Werkzeugen **Akquisition** und **Logik** können die verfügbaren Verarbeitungswerkzeuge in die Gruppen **Lokalisieren**, **Erkennen**, **Zählen** und **Messen** eingeteilt werden.

Die Werkzeuge in der Gruppe **Lokalisieren** sind **Fläche Lokalisieren**, **Kante Lokalisieren**, **Form Lokalisieren** und **Barcode Lokalisieren**. Sie werden verwendet, um einen Referenzort in einer Szene zu finden - eine sogenannte Pose, die aus einer Position und einer Drehung besteht -, welche von jedem anderen Werkzeug in der Verarbeitungspipeline verwendet werden kann.

Die Werkzeuge in der Gruppe **Erkennen** erkennen Helligkeit und Kontrast sowie Flächen, Kanten und Barcodes.

Die Werkzeuge in der Gruppe **Zählen** zählen Flächen, Kanten und Formen.

Die Werkzeuge in der Gruppe **Messen** messen Winkel, Kreise und Abstände.

Nicht alle Werkzeuge sind auf jedem Sensor verfügbar. Der verfügbare Werkzeugsatz wird über Lizenzen freigeschaltet. Das Modell **Basis** enthält die Werkzeuge von **Lokalisieren**, **Detektieren** und **Zählen** - ohne **Barcode Lokalisieren** und **Barcode Erkennen**. Die Erweiterungen **Messen** enthalten die Werkzeuge zum Messen, und die Erweiterung **ID** enthält die Werkzeuge **Barcode Lokalisieren**, **Barcode Erkennen** und **Barcodes zählen**. Zusätzlich gibt es optional die **ID-Pro** Erweiterung für verbessertes Lesen von direkt-markierten Codes sowie Verifikation von 2D Codes.

2.3 Eingangsdaten

Viele Werkzeugparameter können durch Auswahl oder Eingabe von Werten geändert werden. Die meisten dieser Parameter können auch bearbeitet werden durch **Verknüpfen** - d. h. Verknüpfen mit digitalen Eingängen, Feldbus-Eingängen oder über TCP/IP Variablen.

Das Verknüpfen eines Werts erfolgt in zwei Schritten: Zunächst wird der Wert auf den Registerkarten **Digitale Eingänge**, **PROFINET-Eingänge**, **EtherNet/IP-Eingänge** oder **TCP/IP Variablen Eingänge** des Werkzeugs **Akquisition** benannt. Dann kann der benannte Wert mit der Taste als Parameter verwendet werden.

Eingabedaten werden immer dann erfasst, wenn ein Bild erfasst wird. Dies geschieht zu Beginn eines Inspektionszyklus.

2.3.1 Verknüpfen von Digitaleingängen

Der Vision-Sensor **CS-6x/ID-6x0** verfügt über zwei digitale Eingänge, die verknüpft werden können. Die Eingangsknoten werden auf der Seite **Digitale Eingänge** des Werkzeugs **Akquisition** angezeigt. Um sie zum Verlinken zu verwenden, müssen sie benannt werden. Ein Eingang wird benannt durch Hinzufügen eines **Link** Knotens (mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden des Eingabeknoten mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

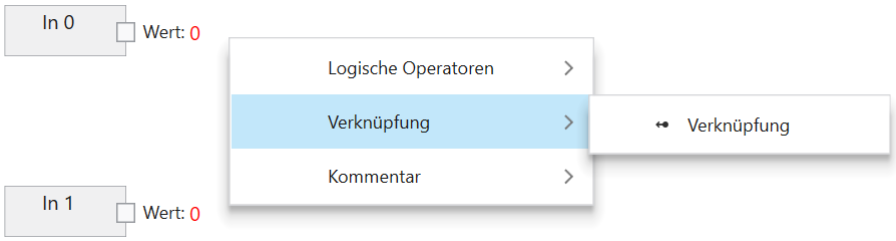


Abb. 3: Hinzufügen eines **Link** Knotens mit einen Rechts-Klick.



Abb. 4: Verbindung des Eingabeknotens mit dem **Link** Knoten.

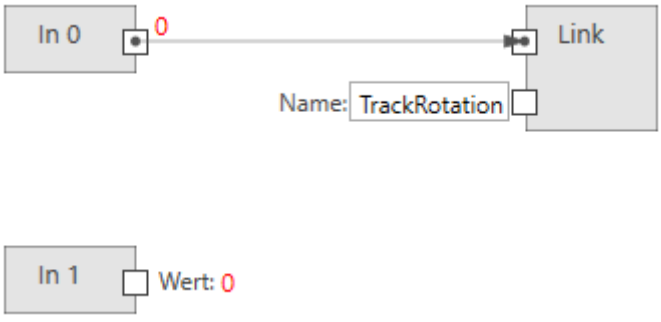


Abb. 5: Namesvergabe des **Link**

Sobald Sie Links benannt haben, können Sie diese in jedem Werkzeug (außer dem Akquisitions-Werkzeug selbst) als Boolesche Werte verwenden. Ein Beispiel Dies kann die Einstellung **Rotation verfolgen** im Werkzeug **Fläche Lokalisieren** sein.

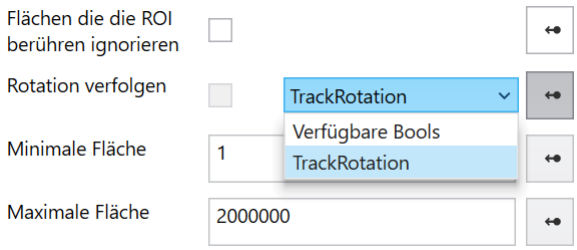


Abb. 6: Beispiel der Verwendung eines Links.

Aus technischen Gründen können digitale Eingänge nicht zum Verknüpfen von Werten in das Werkzeug **Akqisition** verwendet werden.

2.3.2 Verknüpfen von Feldbus Eingängen

Hinweis: Die Screenshots zeigen PROFINET, aber die Verfahren für EtherNet/IP sind genau dieselben, es sei denn, es ist ausdrücklich vermerkt.

Der **SPS** Knoten wird verwendet, um Module in bestimmten Steckplätzen zu definieren. Dieses Setup muss exakt mit dem Setup auf der SPS übereinstimmen, um die Kommunikation zwischen der SPS und dem **CS-6x/ID-6x0** zu ermöglichen. Der SPS Knoten wird sowohl im Akquisition als auch im Logik Werkzeug angezeigt. Innerhalb des SPS Knotens werden die Namen Eingang und Ausgang für die Module wie von der SPS aus gesehen verwendet (Ein SPS Ausgang wird von der SPS zum **CS-6x/ID-6x0** übertragen, wo er als Eingang angesehen wird. Ein SPS-Eingang wird vom **CS-6x/ID-6x0** über zur SPS übertragen, auf dem **CS-6x/ID-6x0** wird dies als Ausgang gesehen.)

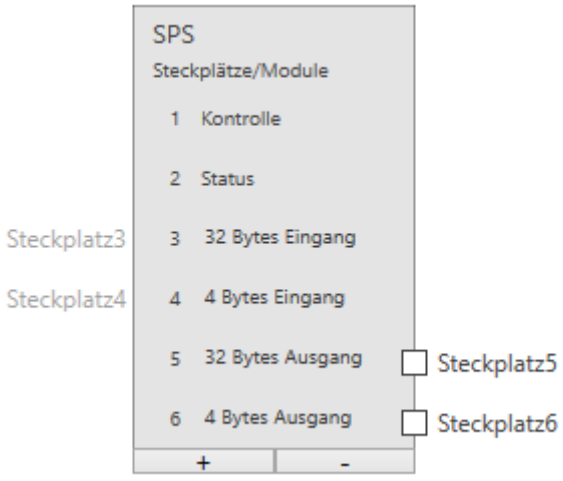


Abb. 7: Der SPS Knoten mit sechs Modulen.

Beiespielhaft sind hier sechs Module ausgewählt (die Module Kontrolle und Status sind nicht abwählbar). Auf der Seite **PROFINET Eingänge** können nur die Module 5 und 6 verbunden werden (aus Sicht der SPS sind das Ausgänge, aus Sicht des **CS-6x/ID-6x0** sind das Eingänge.)

Unter der Überschrift **Steckplätze** werden die Module grafisch angezeigt. Außerdem werden die verwendeten Regionen innerhalb der Module angezeigt.

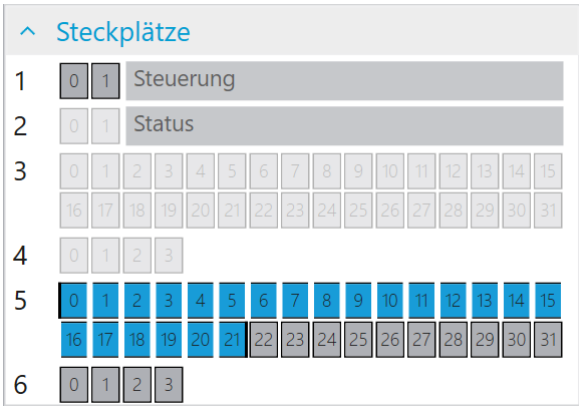


Abb. 8: Ein String mit 20 Zeichen im Modul 5 an Adresse 0.

PROFINET oder EtherNet/IP kann Eingabewerte verschiedener Typen bereitstellen: Logisch (Bool), Numerisch (Byte, Int32, Float) und Text (String). Um sie zum Verlinken zu verwenden, müssen sie definiert und benannt werden. Ein Eingang wird definiert, indem ein Leseknoten des gewünschten Typs hinzugefügt wird (mit einem Rechtsklick auf Canvas und Auswahl von **Read Boolean**, **Read Byte**, **Read Int32**, **Read Float** oder **Read String**) und mit einem bestimmten Modul in einem Steckplatz verbunden wird, der im SPS Knoten definiert ist. Ein Eingang wird benannt durch Hinzufügen eines **Verknüpfung** Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden eines Leseknotens mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

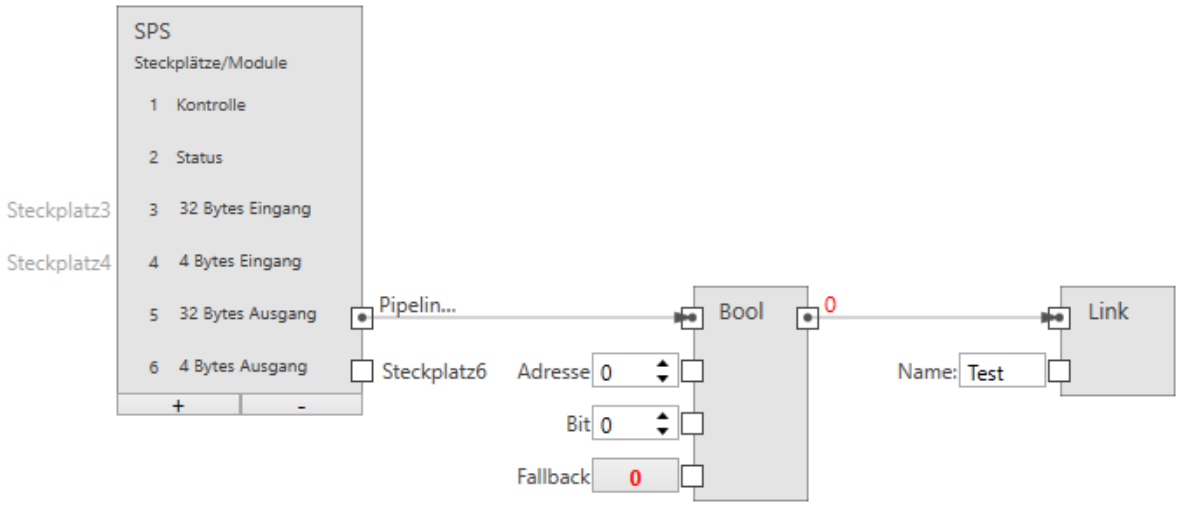


Abb. 9: Ein boolescher Eingang.

Die Größe eines booleschen Eingangs beträgt 1 Bit (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Bit gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Fallback gibt den Booleschen Wert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

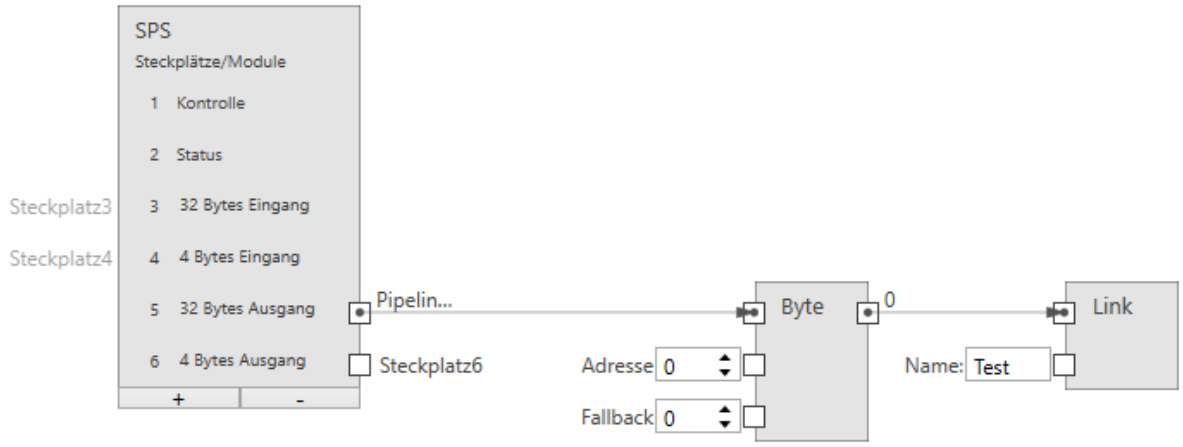


Abb. 10: Ein Byte Eingang.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Fallback gibt den Byte Wert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

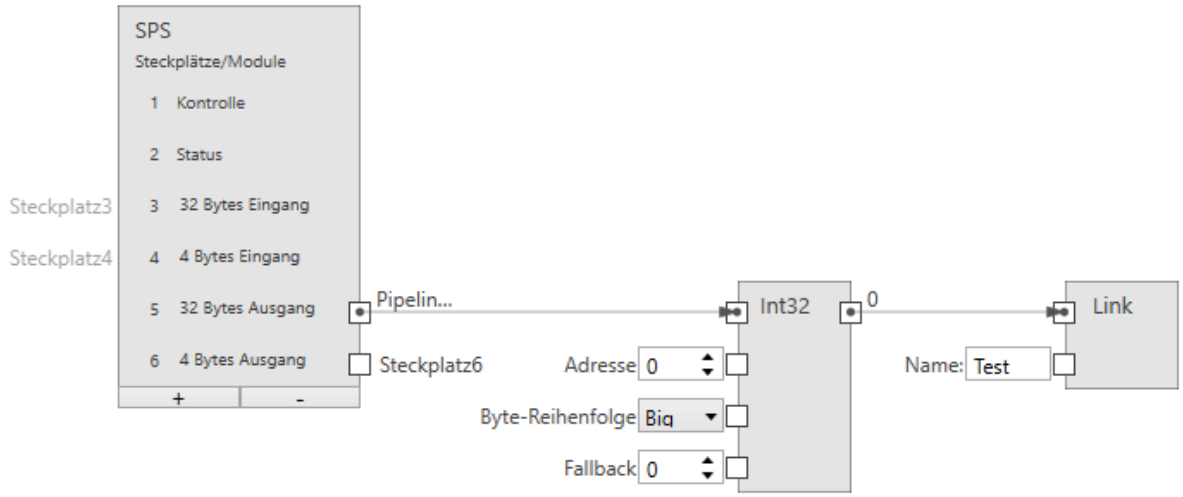


Abb. 11: Ein ganzzahliger Eingang.

Die Größe eines ganzzahligen Eingangs beträgt 4 Byte.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback gibt den Zahlenwert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

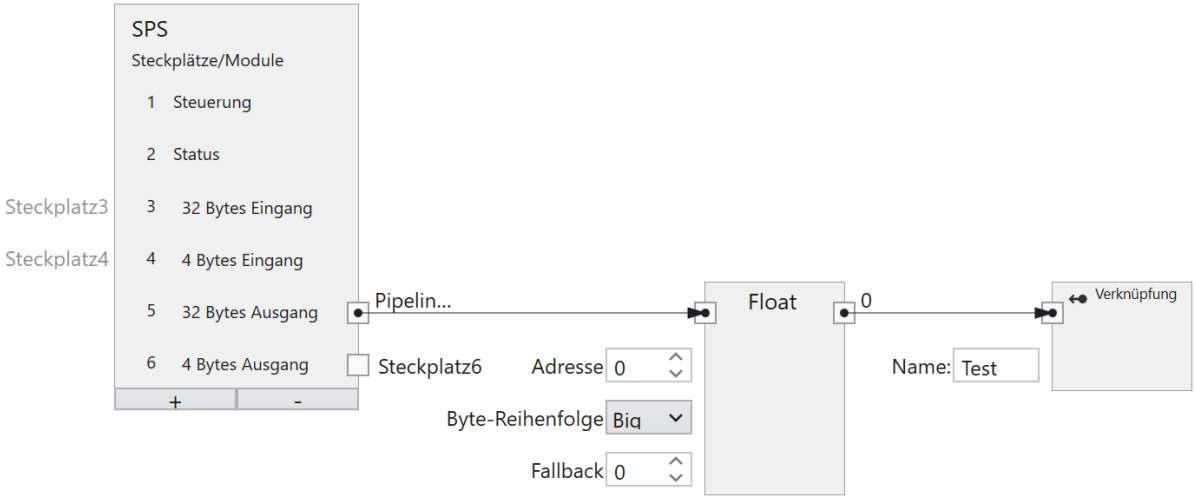


Abb. 12: Ein Fließkomma Eingang.

Die Größe eines Fließkomma Eingangs beträgt 4 Byte.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback gibt den Fließkommawert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

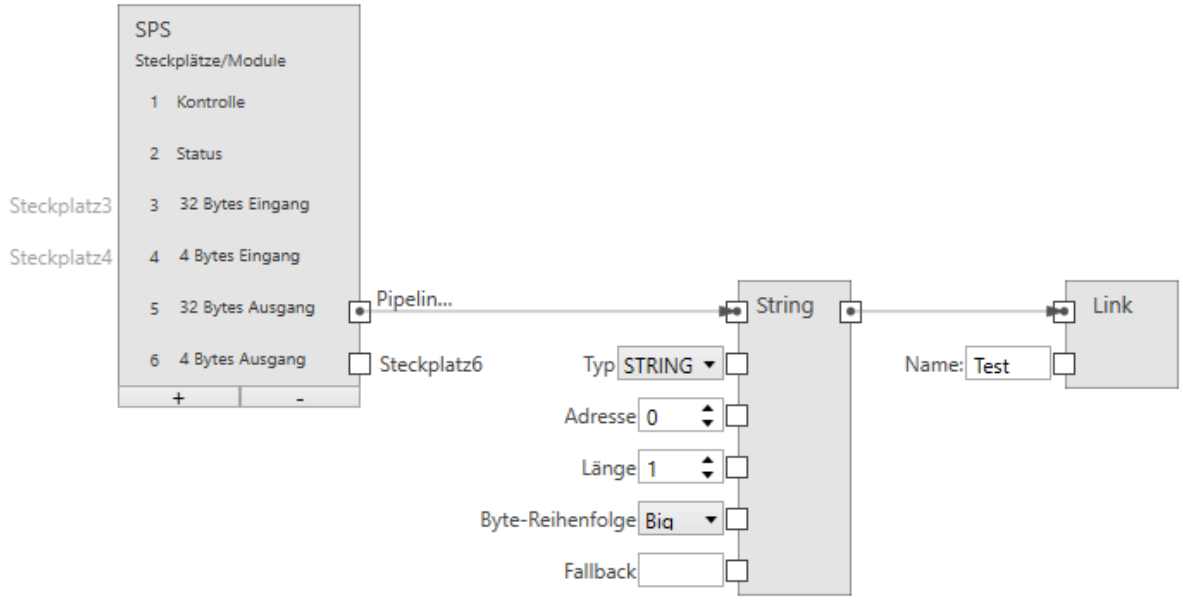


Abb. 13: Ein Text Eingang.

Profinet: Die Größe eines Texts beträgt (2 + n) * s, wobei n die maximale Anzahl von Zeichen ist und s = 1 (für einen Ansi STRING) oder s = 2 (für einen Unicode-WSTRING). Das erste Byte/Wort (Ansi/Unicode) enthält die maximale Anzahl von Zeichen, welche die Zeichenfolge enthalten kann. Das zweite Byte/Wort (Ansi/Unicode) enthält die tatsächliche Anzahl der Zeichen, welche die Zeichenfolge enthält.

EtherNet/IP: Die Größe eines Texts beträgt (4 + n), wobei n die maximale Anzahl von Zeichen ist. Die ersten 4 Byte werden als int32 interpretiert und stellt die tatsächliche Anzahl der Zeichen des strings da.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Typ gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist. Bei EtherNet/IP entfällt dieser Input, da EtherNet/IP nur Ansi Texte unterstützt.

Adresse gibt die Adresse im Modul an.

Länge gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Byte-Reihenfolge Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden. Bei EtherNet/IP entfällt dieser Input, da er nur für Unicode Texte relevant ist.

Fallback gibt den String Wert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

2.3.3 Verknüpfen von TCP/IP Kommando Eingängen

Der **Inputs** Knoten wird verwendet, um Variablen im TCP/IP Namensraum zu benennen.

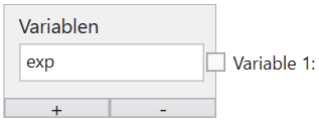


Abb. 14: Der Inputs Knoten mit einer Variablen.

Beispielhaft ist hier eine Variable definiert. Der Name muss mit einem Buchstaben beginnen und kann optional gefolgt sein von einem oder mehreren Buchstaben, Ziffern oder _ (Unterstrich). Dieser Name ist gültig im TCP/IP Namensraum, d.h. bei Verwendung mit den TCP/IP Kommando `set name=value`. Beispielsweise würde das Kommando `set exp=100` den Wert `100` der Variablen `exp` zuweisen.

TCP/IP Kommando Variablen können Eingabewerte verschiedener Typen bereitstellen: Logisch (Bool), Numerisch (Byte, Int32, Float) und Text (String). Um sie zum Verlinken zu verwenden, müssen sie definiert und benannt werden. Ein Eingang wird definiert, indem ein Leseknoten des gewünschten Typs hinzugefügt wird (mit einem Rechtsklick auf Canvas und Auswahl von **Read Boolean**, **Read Byte**, **Read Int32**, **Read Float** oder **Read String**) und mit einem bestimmten Modul in einem Steckplatz verbunden wird, der im Inputs Knoten definiert ist. Ein Eingang wird benannt durch Hinzufügen eines **Verknüpfung** Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden eines Leseknotens mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

Bool Eingang

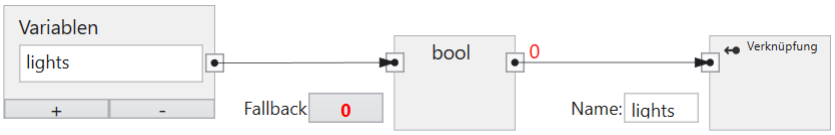


Abb. 15: Ein Bool Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den Booleschen Wert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Byte Eingang

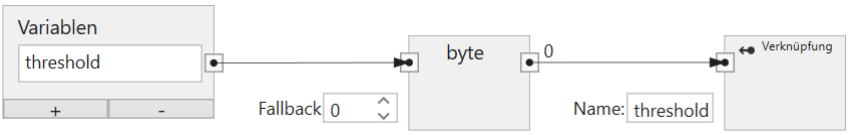


Abb. 16: Ein Byte Eingang.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Fallback gibt den Byte Wert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Ganzzahliger Eingang

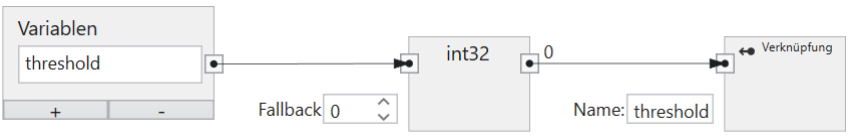


Abb. 17: Ein ganzzahliger Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den Zahlenwert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Fließkomma Eingang

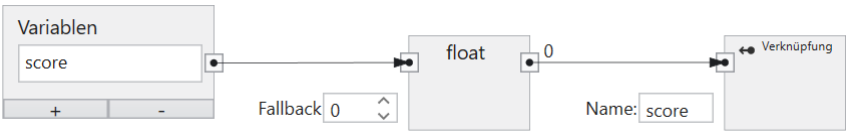


Abb. 18: Ein Fließkomma Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den Fließkommaert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Text Eingang

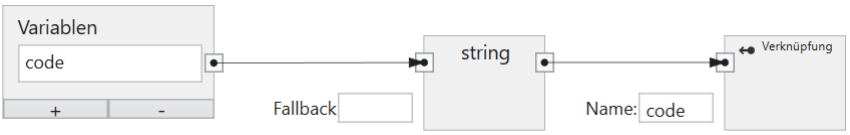


Abb. 19: Ein Text Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den String Wert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

2.4 Ausgangsdaten

Ein Job kann Ausgabedaten erzeugen. Die Ausgabe erfolgt am Ende eines Inspektionszyklus.

Das Werkzeug **Logik** am Ende eines Jobs wird verwendet, um die Ausgaben eines Jobs anzugeben.

2.4.1 Digitale Ausgänge produzieren

Der **CS-6x/ID-6x0** verfügt über vier digitale Ausgänge.

Standardmäßig werden die vier digitalen Ausgänge des **CS-6x/ID-6x0** als Knoten auf der Zeichenfläche angezeigt.

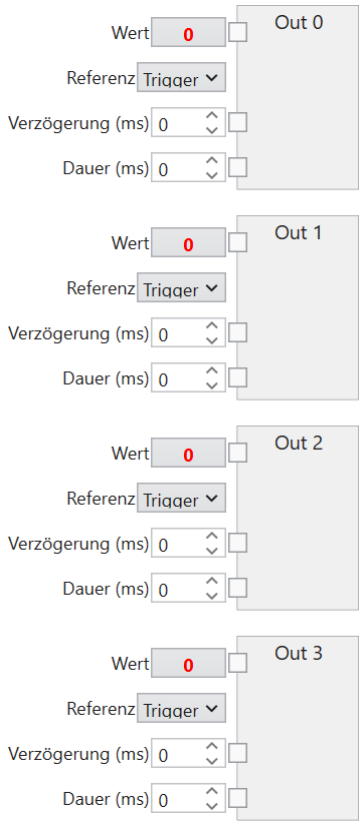


Abb. 20: Die digitalen Ausgangsknoten des **Logic** werkzeugs.

Die **Verzögerung** und die **Länge** eines digitalen Ausgangssignals können angegeben werden (in Millisekunden). Die Verzögerung wird ab dem Start der Bildaufnahme gemessen, eine Länge von 0 bedeutet unendliche Dauer.

Werkzeuge können mit der Maus aus der Werkzeugliste gezogen werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die noch logisch kombiniert werden können, um die gewünschten Ausgaben zu erstellen. Hier ist ein Beispiel:

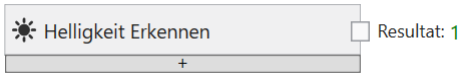


Abb. 21: Ein Werkzeug Knoten.

Das primäre Ergebnis des Werkzeugs ist **True** oder **False** und kann mit jedem der vier Ausgänge verbunden werden (durch Ziehen vom kleinen quadratischen Ausgangs-Pin mit der Bezeichnung **Resultat** des Werkzeugs auf den entsprechenden Pin) - auf den Pin des Ausgangs mit dem Namen **Wert**.



Abb. 22: Ein Werkzeug mit einem Ausgang verbinden.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

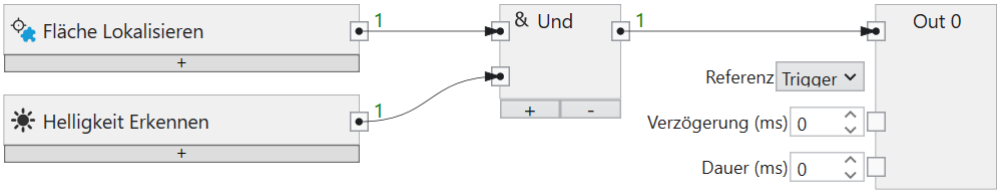


Abb. 23: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Ein Knoten oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

Ein Klick auf die kleine Schaltfläche **+** am unteren Rand eines Werkzeugknotens fügt dem Werkzeugknoten eine zusätzliche Zeile hinzu, in der Sie einen logischen Ausdruck für jedes sekundäre Werkzeugergebnis erstellen können.

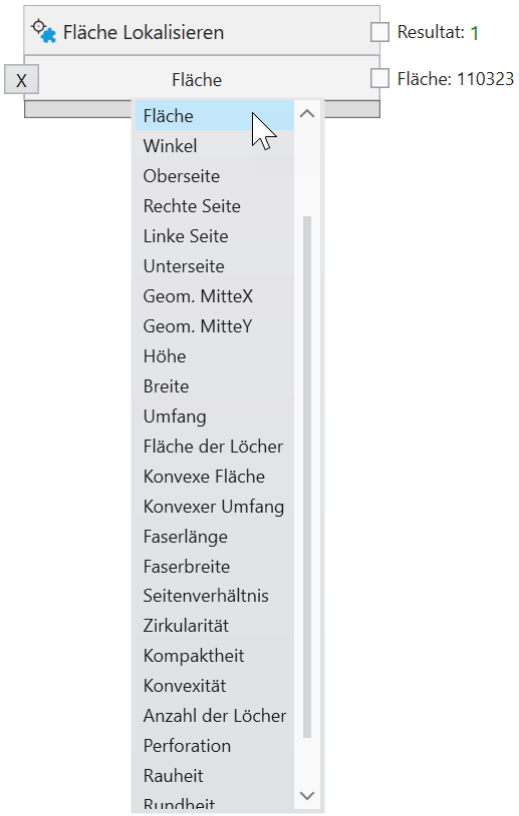


Abb. 24: Zusätzliche logische Ausgänge mit Sekundärresultaten erzeugen.

Geben Sie minimale oder maximale Grenzen ein und erstellen Sie zusätzliche Logikergebnisse aus den Ergebnissen des sekundären numerischen Resultats.

Die Taste **X** löscht eine Zeile, wenn sie nicht mehr benötigt wird.

2.4.2 Feldbus Ausgänge produzieren

Hinweis: Die Screenshots zeigen PROFINET, aber die Verfahren für EtherNet/IP sind genau dieselben, es sei denn, es ist ausdrücklich vermerkt.

Feldbus-Ausgänge können definiert werden, indem Sie im linken Menü **Write Boolean**, **Write Byte**, **Write Int32**, **Write Float** oder **Write String** auswählen oder mit der rechten Maustaste in die Zeichenfläche klicken um eine Auswahl zu treffen.

Werkzeuge können aus der Werkzeugliste gewählt werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die noch logisch kombiniert werden können, um die gewünschten Ausgaben zu erstellen. Hier ist ein Beispiel:

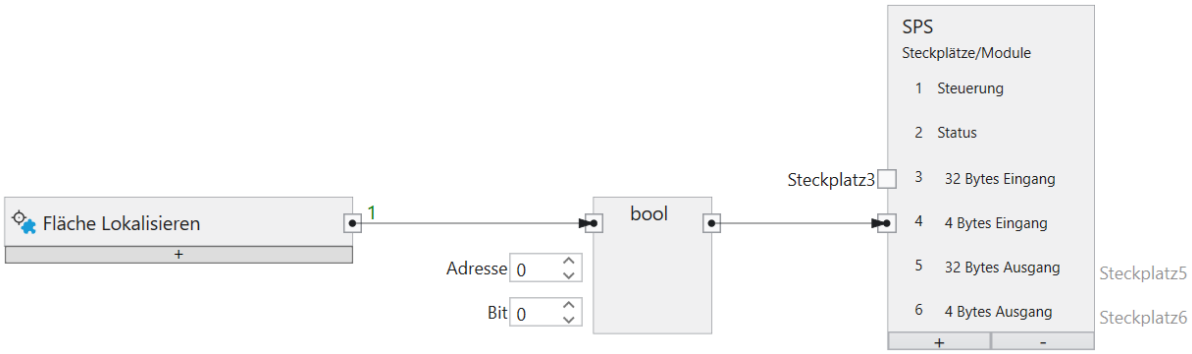


Abb. 25: Ein Werkzeug Knoten für PROFINET.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

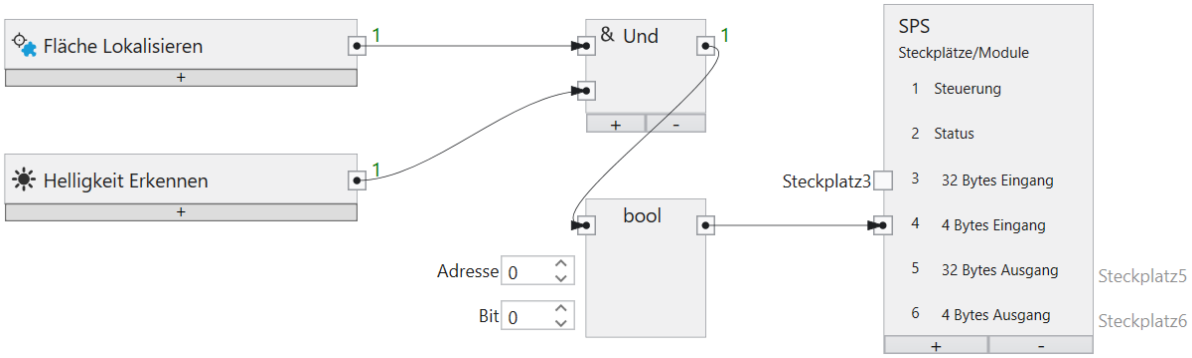


Abb. 26: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Ein Knoten oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

PROFINET oder EtherNet/IP kann neben logischen Werten auch numerische Werte oder Texte übertragen.

Mit den PROFINET oder EtherNet/IP Ausgangsknoten (**Write Boolean**, **Write Byte**, **Write Int32**, **Write Float** oder **Write String**) können Sie bestimmen, an welche Adresse innerhalb eines Moduls die Werte geschrieben werden.



Abb. 27: Ein boolescher Ausgang.

Die Größe eines booleschen Ausgangs beträgt 1 Bit (Sie können jedoch bis zu 8 bit in einem Byte zuweisen).

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Bit gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).



Abb. 28: Ein Byte Ausgang.

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

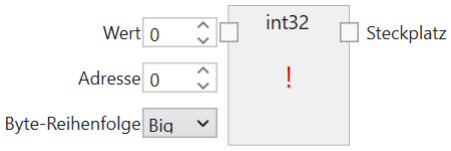


Abb. 29: Ein ganzzahliger Ausgang.

Die Größe eines ganzzahligen Ausgangs beträgt 4 Byte.

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

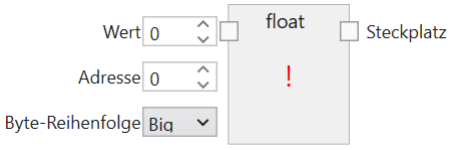


Abb. 30: Ein Fließkomma Ausgang.

Die Größe eines Fließkomma Ausgangs beträgt 4 Byte.

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

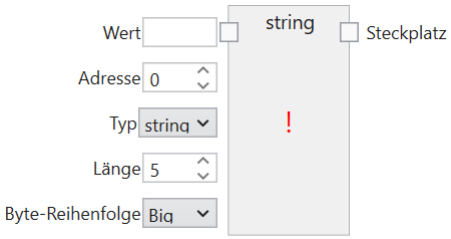


Abb. 31: Ein Text Ausgang.

Profinet: Die Größe eines Texts beträgt (2 + n) * s, wobei n die maximale Anzahl von Zeichen ist und s = 1 (für einen Ansi STRING) oder s = 2 (für einen Unicode-WSTRING). Das erste Byte/Wort (Ansi/Unicode) enthält die maximale Anzahl von Zeichen, welche die Zeichenfolge enthalten kann. Das zweite Byte/Wort (Ansi/Unicode) enthält die tatsächliche Anzahl der Zeichen, welche die Zeichenfolge enthält.

EtherNet/IP: Die Größe eines Texts beträgt (4 + n), wobei n die maximale Anzahl von Zeichen ist. Die ersten 4 Byte werden als int32 interpretiert und stellt die tatsächliche Anzahl der Zeichen des strings da.

Wert ist der Wert, der übertragen werden soll.

Typ gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist. Bei EtherNet/IP entfällt dieser Input, da EtherNet/IP nur Ansi Texte unterstützt.

Adresse gibt die Adresse im Modul an.

Länge gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Byte-Reihenfolge Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden. Bei EtherNet/IP entfällt dieser Input, da er nur für Unicode Texte relevant ist.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

Der **SPS** Knoten wird verwendet, um Module in bestimmten Steckplätzen zu definieren. Dieses Setup muss exakt mit dem PROFINET-Setup oder dem EtherNet/IP-Setup auf der SPS übereinstimmen, um die Kommunikation zwischen der SPS und dem **CS-6x/ID-6x0** zu ermöglichen. Der SPS Knoten wird sowohl im Akquisition als auch im Logik Werkzeug angezeigt. Innerhalb des SPS Knotens werden die Namen Eingang und Ausgang für die Module wie von der SPS aus gesehen verwendet. Ein SPS Ausgang wird von der SPS über PROFINET oder EtherNet/IP zum

CS-6x/ID-6x0 übertragen, wo er als Eingang angesehen wird. Ein SPS-Eingang wird vom **CS-6x/ID-6x0** über PROFINET oder EtherNet/IP zur SPS übertragen, auf dem **CS-6x/ID-6x0** wird dies als Ausgang gesehen.

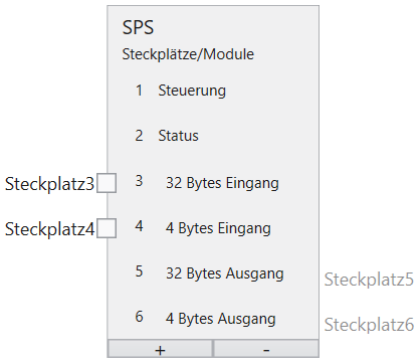


Abb. 32: Der SPS Knoten mit sechs Modulen.

Beiespielhaft sind hier sechs Module ausgewählt.

Unter der Überschrift **Steckplätze** werden die Module grafisch angezeigt. Außerdem werden die verwendeten Regionen innerhalb der Module angezeigt.

2.4.3 TCP/IP Variablen Ausgänge produzieren

Werkzeuge können aus der Werkzeugliste gewählt werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die mit dem TCP/IP Outputs Knoten verbunden werden können. Hier ist ein Beispiel:

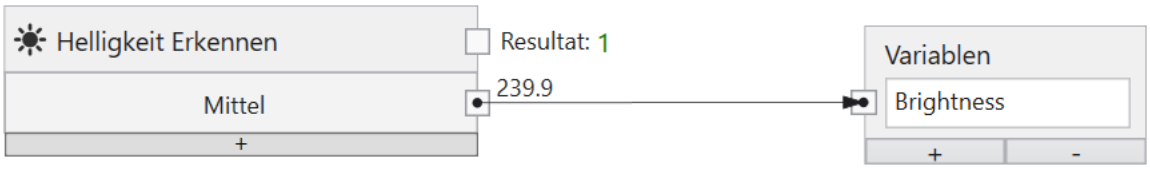


Abb. 33: Ein Werkzeug Knoten für TCP/IP.

TCP/IP-Ausgänge müssen definiert werden, indem Sie sie im Kontextmenü links auswählen oder mit der rechten Maustaste klicken und **Write Bool**, **Write Byte**, **Write Int32**, **Float schreiben** oder **String schreiben** auswählen.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste hinzugefügt werden.

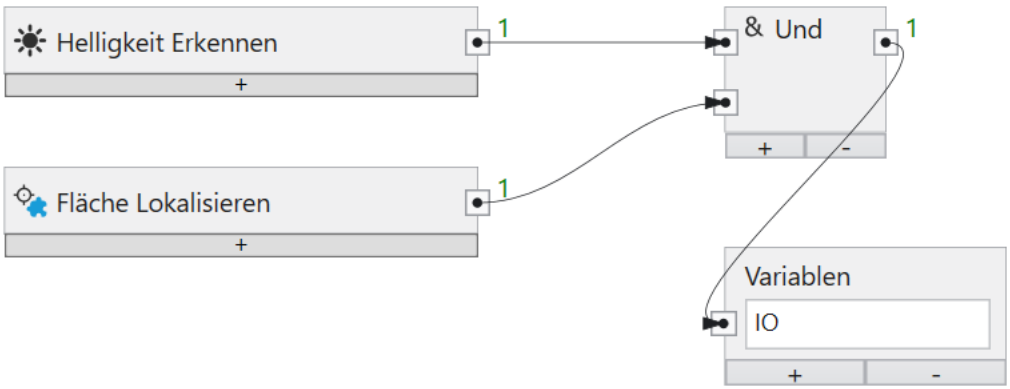


Abb. 34: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Ein Knoten oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

TCP/IP Kommando Variablen können neben logischen Werten auch numerische Werte oder Texte übertragen.

KAPITEL 3

Machine Vision Methoden

3.1 Mustererkennung

nVision-i stellt drei verschiedene Vefahren zur Mustererkennung bereit: Korrelation, Geometrisch und Keypoint. Sie eignen sich für unterschiedliche Szenarien der Erkennung von zweidimensionalen Mustern unter einer senkrechten Kamera-Ansicht. Im ersten Schritt wird eine Vorlage bzw. ein Muster trainiert. Dabei ist darauf zu achten, dass sowohl die Vorlage als auch das Bild so belichtet werden, dass relevante Pixelbereiche nicht gesättigt sind (entweder tief-schwarz oder hell-weiß). Außerdem sollte die Vorlage so gewählt sein, dass genügend Struktur vorkommt, wie Grauwertverläufe, Kanten und Ecken.

Das Korrelations-Verfahren basiert auf Grauwerten. Dabei wird die normalisierte Kreuz-Korrelation zwischen einer Vorlage und einem Bildbereich im Suchbild berechnet. Das Verfahren kann lineare Variationen in der Helligkeit bis zu einem gewissen Grad kompensieren. Es ist rotations-invariant, d.h. die Vorlage kann unter verschiedenen Drehlagen im Suchbild erkannt werden. Das Korrelations-Verfahren ist speicher-intensiv, und birgt das Risiko von Out-Of-Memory Fehlern, die zu Fehlverhalten führen, wenn die Vorlagen bzw. die zulässige Rotation zu groß gewählt werden. Außerdem ist das Verfahren rechenintensiv und sollte nur gewählt werden, wenn andere Verfahren nicht zum gewünschten Erfolg führen.

Das geometrische Verfahren basiert auf Helligkeits-Gradienten. Das Verfahren kann Verdeckungen kompensieren, und kommt mit Störungen im Hintergrund relativ gut zurecht. Das Verfahren ist rotations-invariant, d.h. die Vorlage kann unter verschiedenen Drehlagen im Suchbild erkannt werden. Das geometrische Verfahren ist speicher-intensiv, und birgt das Risiko von Out-Of-Memory Fehlern, die zu Fehlverhalten führen, wenn die Vorlagen bzw. die zulässige Rotation zu groß gewählt werden. Außerdem ist das Verfahren rechenintensiv und sollte nur gewählt werden, wenn andere Verfahren nicht zum gewünschten Erfolg führen.

Das Keypoint-Verfahren detektiert wenige Schlüssel-Punkte in der Vorlage und dem Bildbereich, und benützt die Korrespondenz dieser Punkte zur Lage-Bestimmung. Bei der Wahl der Vorlage ist darauf zu achten, dass die Anzahl der gefundenen Schlüssel-Punkte ausreichend ist, und dass

sie möglichst über die ganze Vorlage verteilt vorkommen. Neben der Rotationsinvarianz ist die Keypoint-Methode auch invariant gegenüber kleinen Größenänderungen.

Die Erkennung der Muster in den Randbereichen des Suchbildes wird durch zwei Effekte erschwert:

- die Vignettierung der Optik sorgt dafür, dass die Helligkeit in den Ecken abfällt.
- die Verzeichnung der Optik sorgt für eine geometrische Verzerrung, die zum Rand hin stärker wird.

Beide Effekte erschweren die Mustererkennung in den Randbereichen des Bildes. Die Leistung der Mustererkennung kann verbessert werden, wenn diese Effekte kompensiert werden.

3.2 Korrektur der Schattierung

Die Shading Korrektur kompensiert den Helligkeitsabfall in einem aufgenommenen Bild zum Rand hin (auch Vignettierung genannt). Die Abschattung zum Rand hin ist ein optischer Effekt, der mit den verwendeten Objektiven variiert, und auch durch die Beleuchtung hervorgerufen bzw. verstärkt wird. Eine Voraussetzung für eine optimale Shading Korrektur ist die orthogonale Betrachtung einer Szene (die Kamera muss unter einem rechten Winkel auf die Szene blicken).

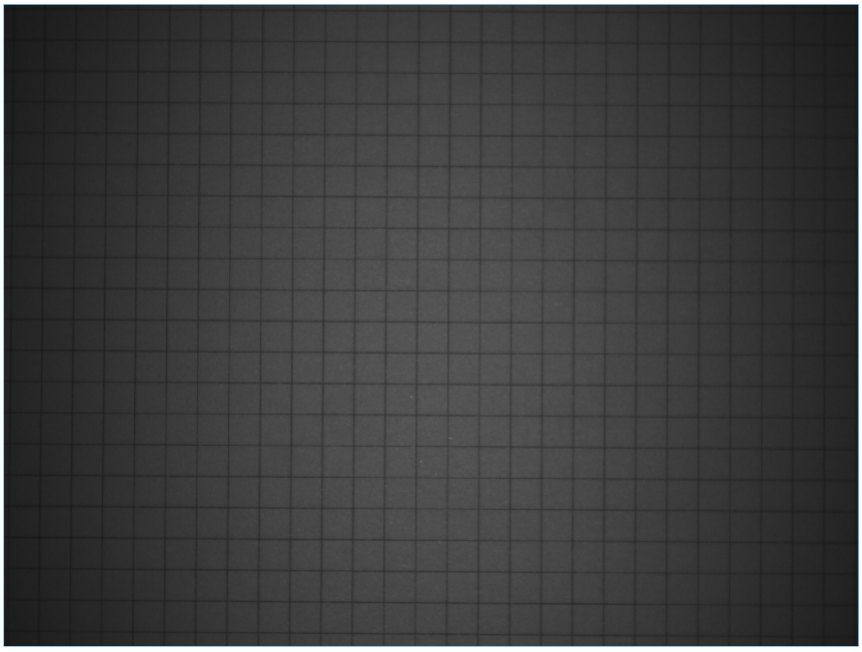


Abb. 1: Schattierung durch Randabschattung.

Die Kalibrierung sollte bei jedem Objektivwechsel bzw. bei einem Wechsel der Beleuchtung neu durchgeführt werden.

Für die Kalibrierung sollte ein möglichst gleichförmig helles Bild aufgenommen, wobei darauf zu achten ist, dass die Intensitäten weder zu dunkel noch zu hell sind. Das kann mit der Histogrammdarstellung bzw. mit den Funktionen **Unterbelichtung hervorheben** und **Überbelichtung hervorheben** aus dem Bereich **Visualisierung** erfolgen.

Wenn das Korrekturbild sowie die Parameter zur Belichtung und Beleuchtung korrekt eingestellt sind, kann mit der Taste **Korrigieren** die Korrektur berechnet werden.

Nach der erfolgten Berechnung kann die Korrektur mit **Korrektur speichern** gespeichert werden, damit sie in Jobs angewandt werden kann, bzw. durch die Taste **Reset** wieder gelöscht werden.

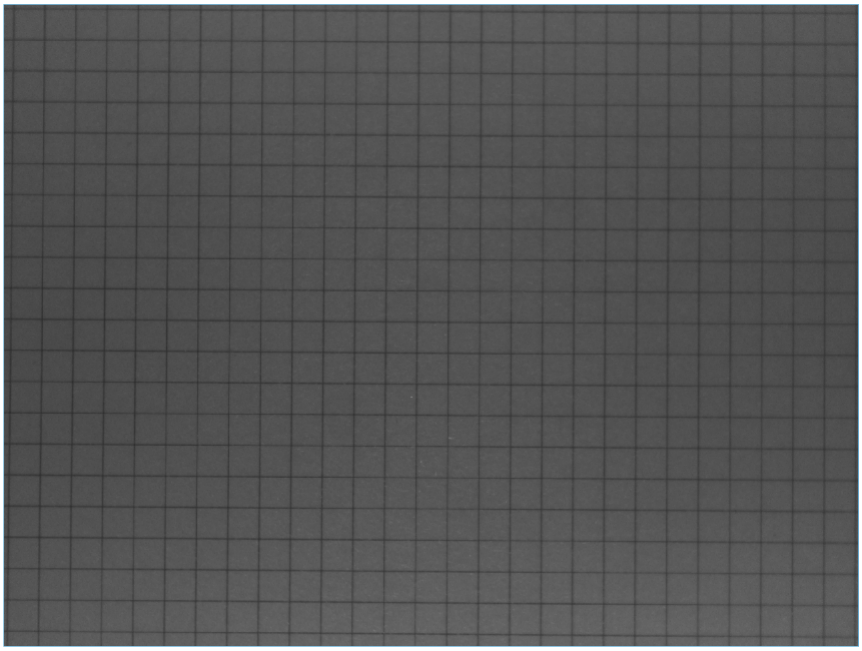


Abb. 2: Korrigierte Randabschattung.

Wenn ein Sensor kalibriert wurde, kann die Shading Korrektur im Akquisition Werkzeug eines Jobs jeweils angewählt oder abgewählt werden. Die Korrektur eines Bildes kostet Zeit, weswegen die Zeit für die Akquisition sich erhöhen wird, falls die Shading Korrektur angewählt wird. Eine Shading Korrektur verbessert die Leistung nachfolgender Werkzeuge, weil Blob-Analyse, Detektion von Kanten und Erkennung von Mustern auch am Bildrand gleichförmige Helligkeitsbedingungen vorfinden.

3.3 Korrektur der geometrischen Verzerrung

Die Verzeichnungs-Korrektur kompensiert optische Verzerrungen, die durch verschiedene Objektive in verschiedenem Grad zum Rand hin verstärkt vorkommen. Eine Voraussetzung für eine optimale Verzerrungs-Korrektur ist die orthogonale Betrachtung einer Szene (die Kamera muss unter einem rechten Winkel auf die Szene blicken). Ein zweiter Effekt der Verzerrungs-Korrektur ist die Bestimmung der intrinsischen und extrinsischen Kamera-Parameter, insoweit sie für eine kalibrierte Messung einer orthogonalen Messebene nötig sind.

Die Kalibrierung muss bei jedem Objektivwechsel bzw. bei einem Wechsel der Mess-Ebene neu durchgeführt werden..

Für die Kalibrierung muss ein Bild eines punktförmigen Kalbrier-Targets möglichst bildfeldfüllend aufgenommen werden, und die Punktabstände eingegeben werden. Solche Targets sind käuflich zu erwerben, bzw. sie können für weniger genaue Anforderungen auch selbst mit einem Dru-

cker ausgedruckt werden. Im Verzeichnis **Docs** der nVision-i Installation liegt eine PDF Datei zum Ausdrucken für verschiedene Bildfeldgrössen.

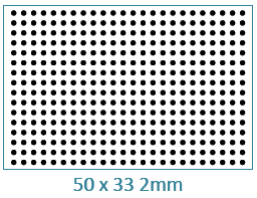


Abb. 3: Eine Vorlage für die Verzeichnungs-Korrektur und die geometrische Kalibrierung.

Das Kalibrierungs-Target muss in der Messebene angebracht werden (nur in dieser Mess-Ebene sind die Messwerte gültig). Bei den verwendeten endozentrischen Objektiven werden weiter entfernte Bildteile kleiner und näher liegende Bildteile größer abgebildet, d.h. die auch die Meßwerte werden entsprechend kleiner bzw. größer. Richtige Meßwerte werden nur in der Messebene ermittelt.

Der **Schwellenwert** für die Detektion des Kalibrations-Targets kann **automatisch** bestimmt oder über einen Schieberegler eingestellt werden. Ziel ist die optimale Bestimmung der Punkte des Kalibriertargets.

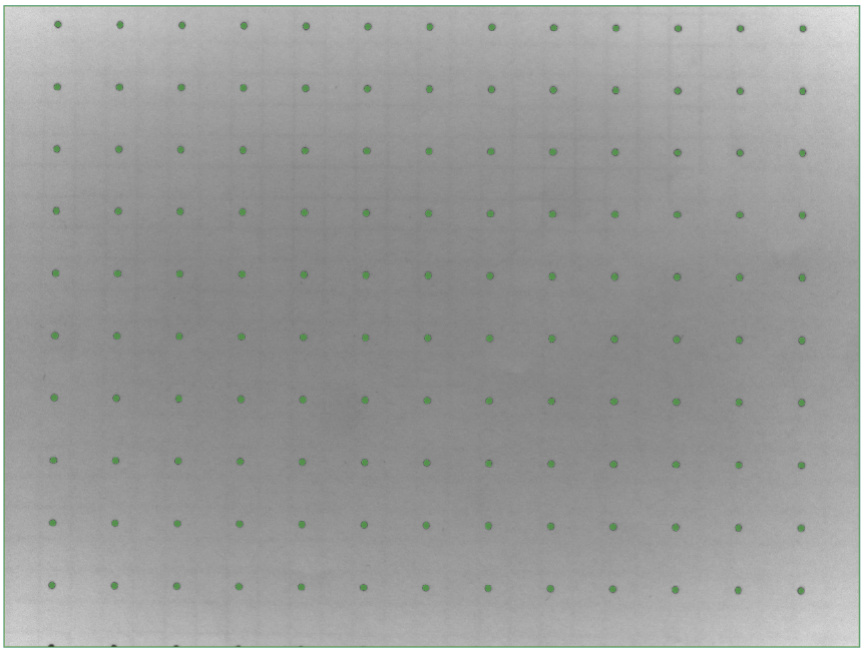


Abb. 4: Bild eines Targets für die geometrische Kalibrierung.

Der **Abstand** ist die Distanz zwischen zwei Punkten des Targets (von Mittelpunkt zu Mittelpunkt). Dieser Wert muss mit dem tatsächlichen Abstand der Punkte auf dem Target übereinstimmen, damit richtig kalibriert werden kann.

Die Region im Bild sollte dazu verwendet werden, die relevanten Punkte des Kalibrierungs-Targets einzugrenzen.

Zur korrekten Einstellung kann das **Raster** angewählt werden, um die Rasterlinien zwischen den Punkten des Targets anzuzeigen.

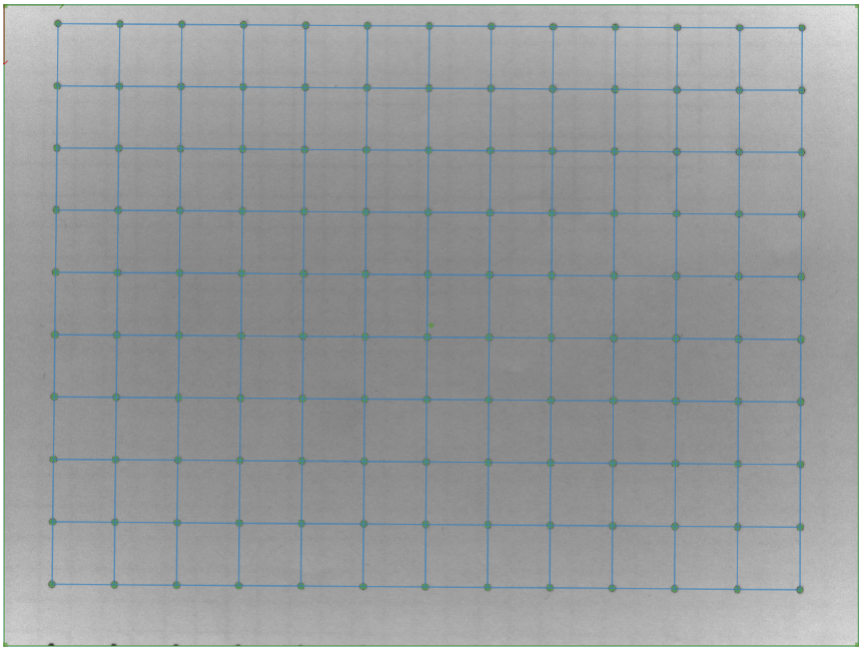


Abb. 5: Target mit Rasterlinien.

Wenn das Korrekturbild sowie die Parameter zur Belichtung und Beleuchtung korrekt eingestellt sind, kann mit der Taste **Korrigieren** die Korrektur berechnet werden.

Die **Kalibrierungsfehler** können angezeigt werden, um die Qualität der Kalibrierung zu beurteilen.

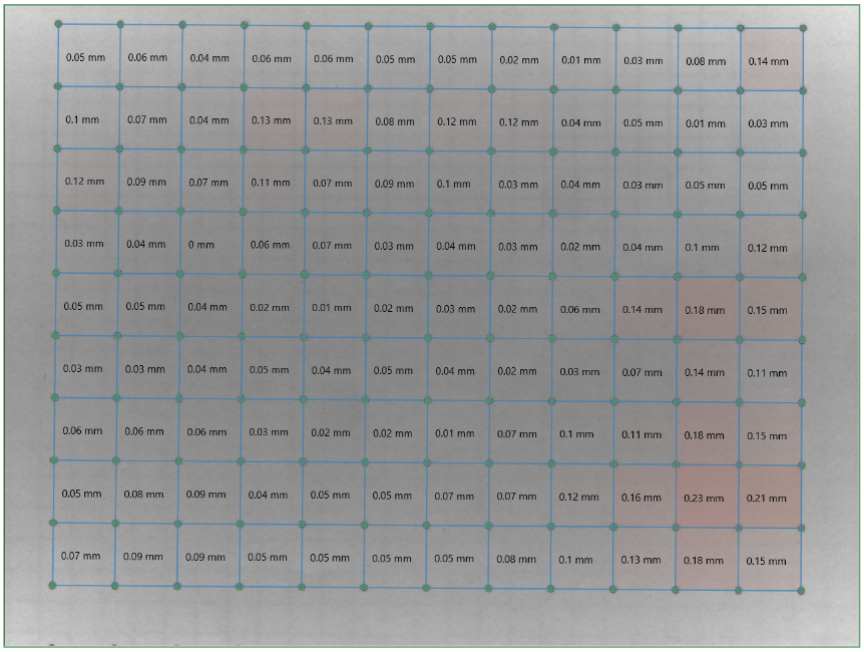


Abb. 6: Target mit Anzeige der Kalibrierungsfehler.

Wenn ein Sensor kalibriert wurde, kann die Entzerrung im Akquisition Werkzeug eines Jobs jeweils angewählt oder abgewählt werden. Die Korrektur eines Bildes kostet Zeit, weswegen die Zeit für die Akqusition sich erhöhen wird, falls die Verzeichnungs-Korrektur angewählt wird. Eine Entzerrung verbessert die Leistung nachfolgender Werkzeuge, weil Blob-Analyse, Detektion von Kanten und Erkennung von Mustern auch am Bildrand gleichförmige Geometrie vorfinden.

3.4 Blob Analyse

Eine Gruppe von räumlich zusammenhängenden Pixeln wird allgemein als Blob bezeichnet.

Bei der Blob-Analyse wird ein Satz von zusammenhängenden Pixeln verwendet, die in der Regel durch die Abbildung realer Objekte erzeugt werden, und aus dieser Pixelmenge werden Werte oder Merkmale wie z. B. ihre Fläche, ihre Position usw. berechnet.

Oft werden BLOBs durch einen Segmentierungsprozess erzeugt, und eine der einfachsten Segmentierungsmethoden ist die binäre Schwellenwertbildung: Jedes Pixel mit einem Wert unterhalb des Schwellenwerts ist Teil des Hintergrunds und jedes andere Pixel ist Teil eines Blob. Zu welchem Blob das Pixel gehört, wird durch eine Konnektivitätsanalyse bestimmt, die manchmal als Labelling oder Analyse der verbundenen Komponenten genannt wird.

Hier ist eine Liste der Merkmale, die berechnet werden können:

Merkmal	Beschreibung
Fläche	Die Fläche als Anzahl von Pixeln.
Winkel	Der Winkel der Fläche.
Schwerpunkt	Der Schwerpunkt.
Oberseite	Die oberste Zeilenposition (einschließlich).
Rechte Seite	Die Position der Spalte ganz rechts (exklusiv).
Unterseite	Die Position der untersten Zeile (exklusiv).
Linke Seite	Die Position der Spalte ganz links (einschließlich).
Geom. Mitte	Der geometrisch Mittelpunkt.
Höhe	Die Höhe (achsparallel).
Breite	Die Breite (achsenparallel).
Umfang	Der Umfang.
Fläche der Löcher	Die Fläche der Löcher als Anzahl von Pixeln.
Konvexe Fläche	Die Fläche der konvexen Hülle.
Konvexer Umfang	Der Umfang der konvexen Hülle.
Faserlänge	Ein Längenmaß für faserartige Objekte.
Faserbreite	Ein Dickenmaß für faserähnliche Objekte.
Hauptachsenlänge	Die Länge der Hauptachse der äquivalenten Ellipse.
Nebenachsenlänge	Die Länge der Nebenachse der äquivalenten Ellipse.
Seitenverhältnis	Verhältnis von Breite zu Höhe.
Zirkularität	Die Zirkularität.
Kompaktheit	Die Kompaktheit.

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Merkmal	Beschreibung
Konvexität	Die Konvexität.
Anzahl der Löcher	Die Anzahl der Löcher in der Region.
Perforation	Die Perforation.
Rauheit	Die Rauheit.
Rundheit	Die Rundheit.
Sphärizität	Die Sphärizität.

Die Merkmale der Blob Analyse können in den Werkzeugen **Fläche Lokalisieren**, **Flächenpixel Erkennen** und **Flächen Zählen** angewandt werden.

3.4.1 Fläche

Die Fläche einer Region ist ein ganzzahliger Wert und gibt die Anzahl der Pixel des Objekts an.

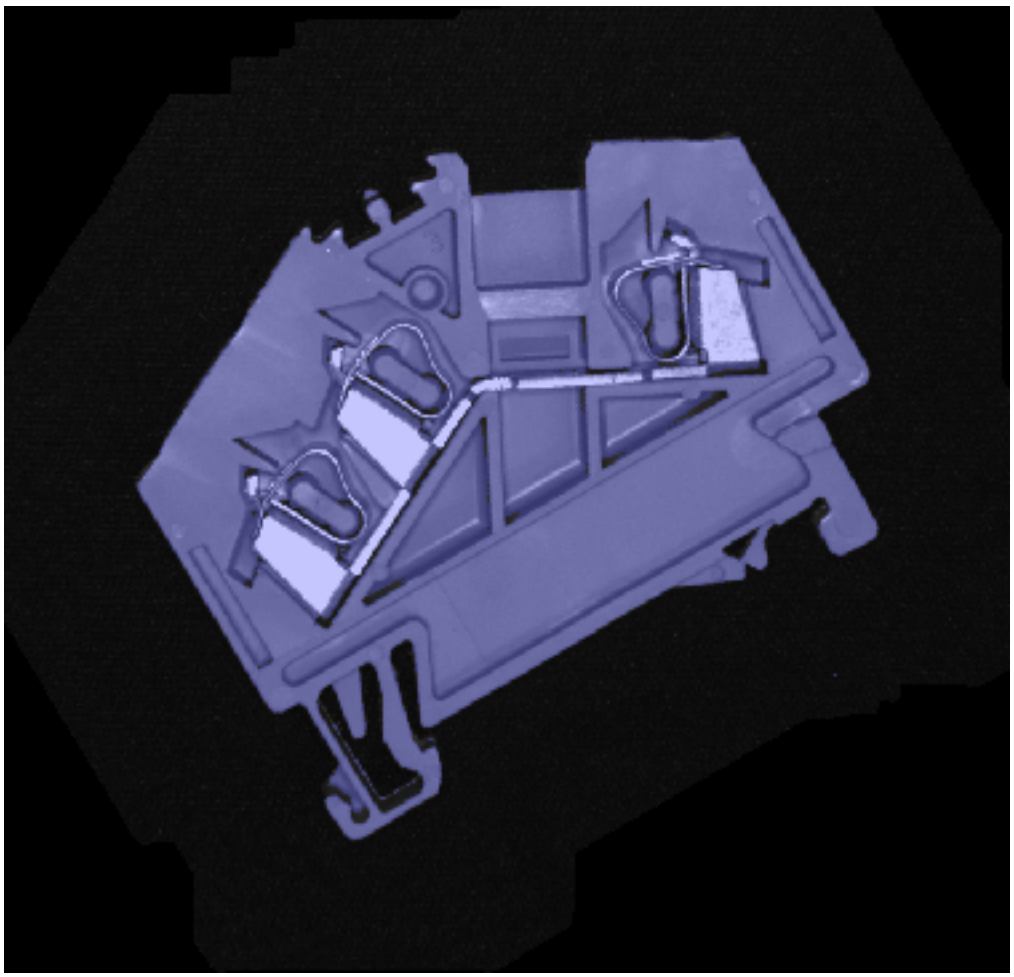


Abb. 7: Die Fläche eines Objekts.

3.4.2 Winkel

Der Winkel einer Fläche ist der Winkel der äquivalenten Ellipse der Fläche.

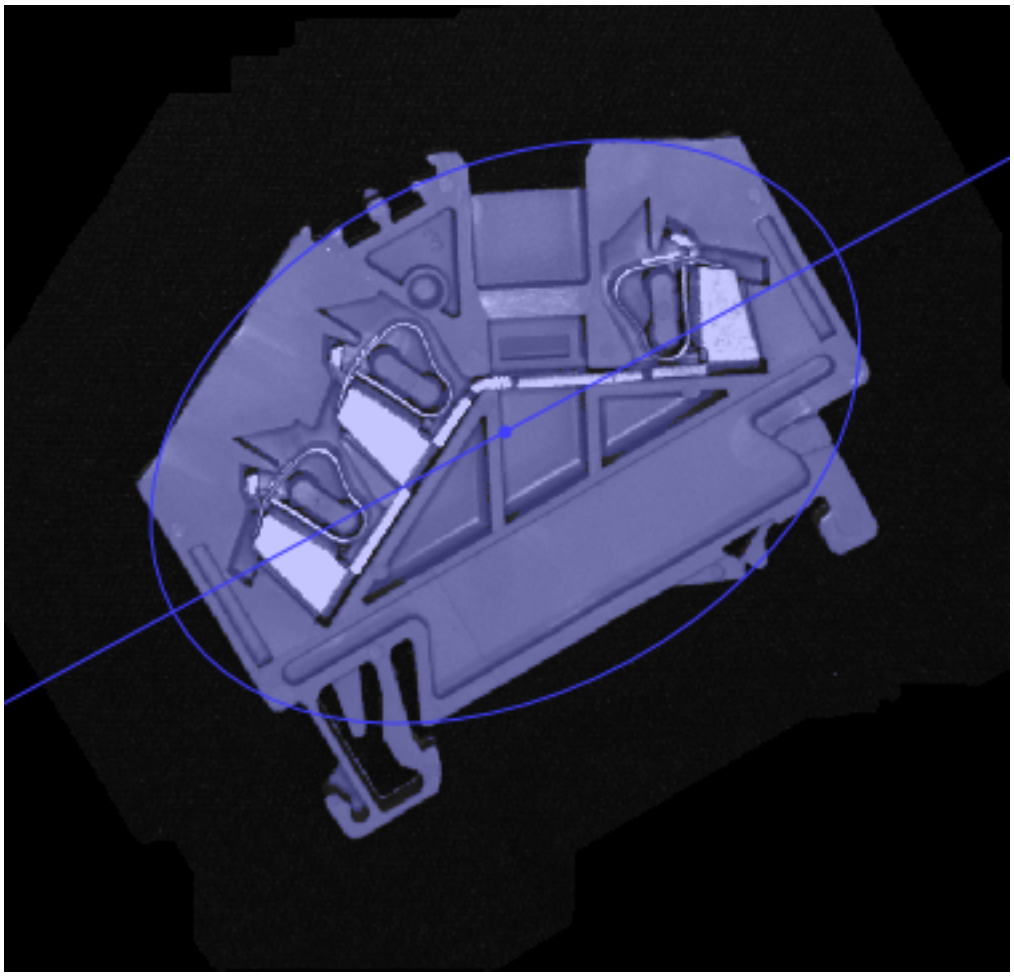


Abb. 8: Der Winkel eines Objekts.

3.4.3 Schwerpunkt

Berechnet den Objektschwerpunkt.

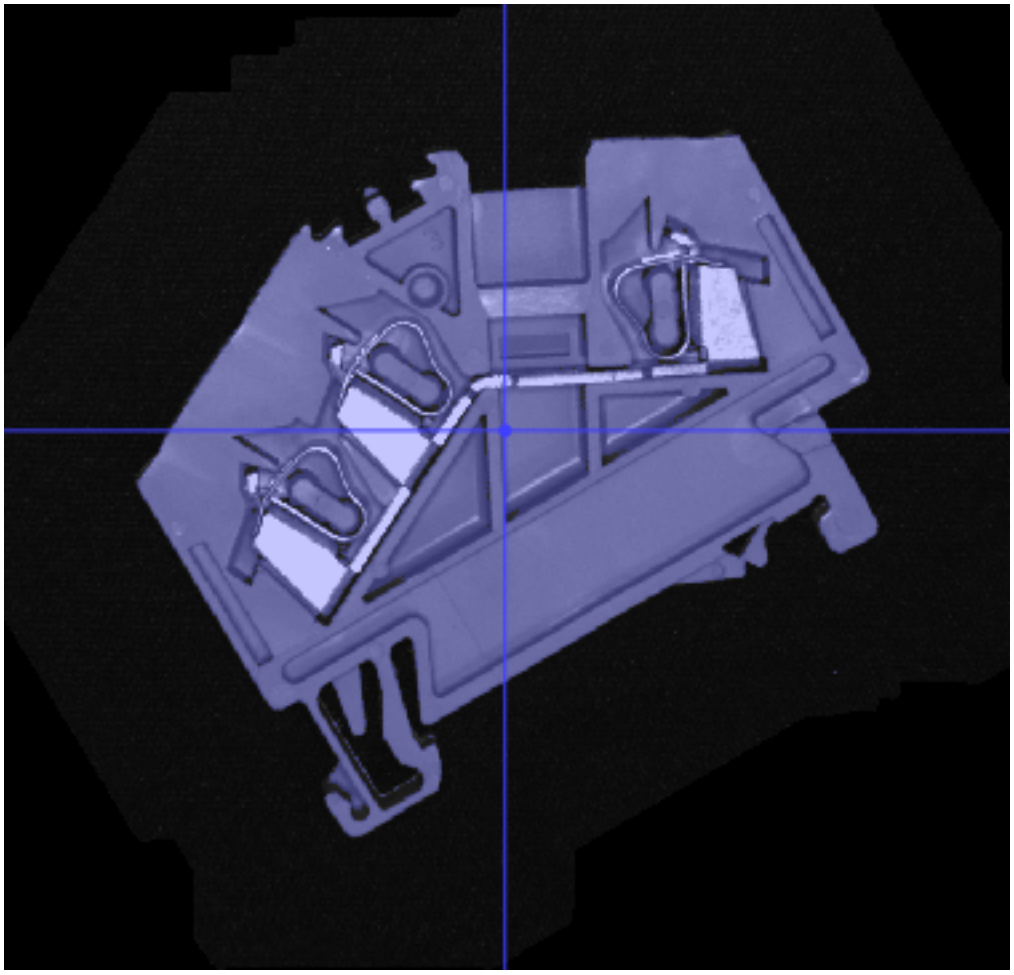


Abb. 9: Der Schwerpunkt eines Objekts.

3.4.4 Oberseite

Dadurch wird die minimale y-Koordinate (einschließlich) der Region berechnet.

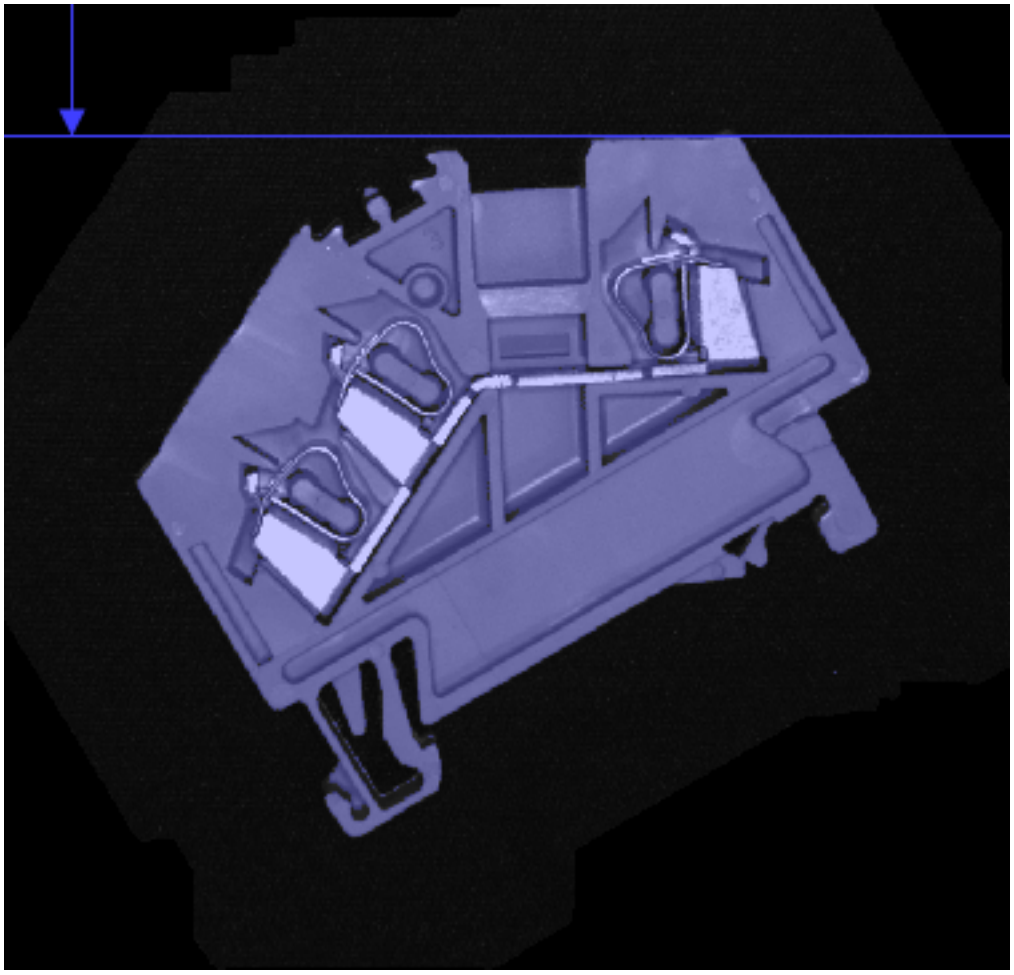


Abb. 10: Die minimale vertikale Koordinate.

3.4.5 Rechte Seite

Dies berechnet die maximale x-Koordinate (exklusiv) der Region.

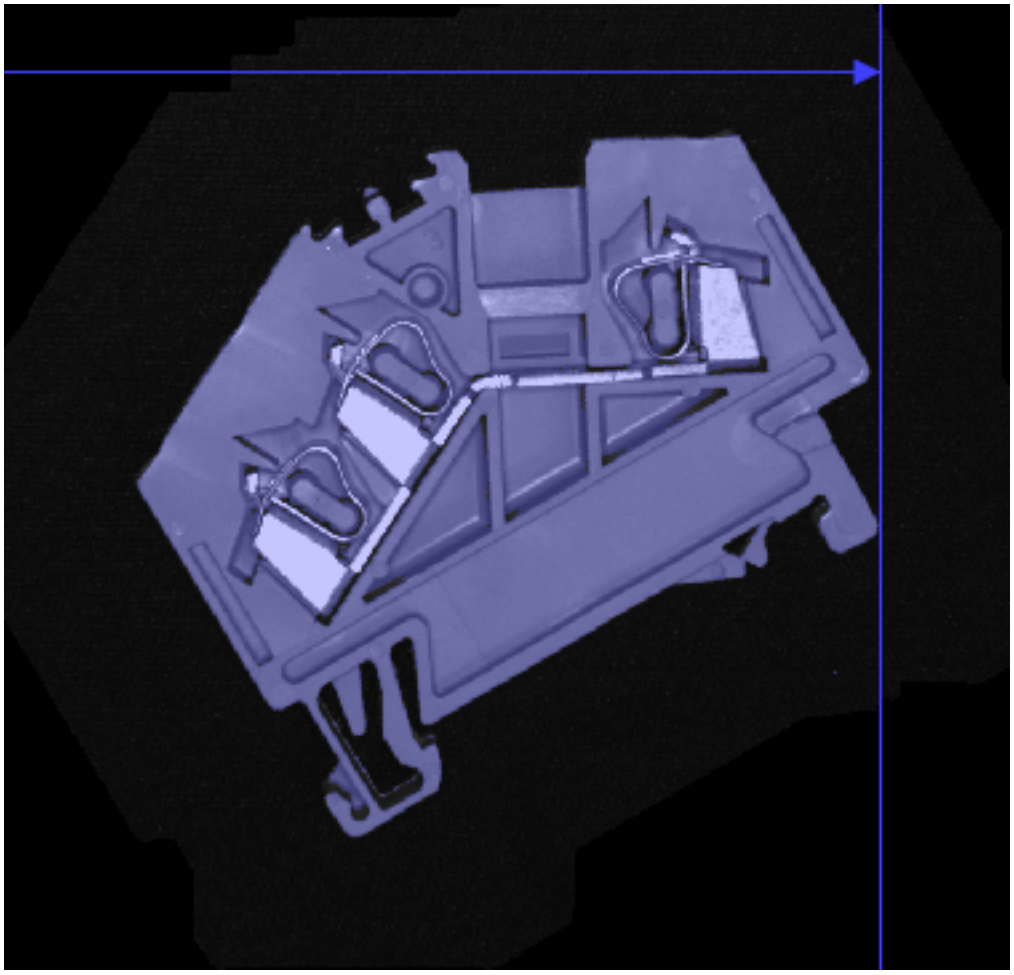


Abb. 11: Die maximale horizontale Koordinate.

3.4.6 Unterseite

Berechnet die maximale y-Koordinate (exklusiv) der Region.

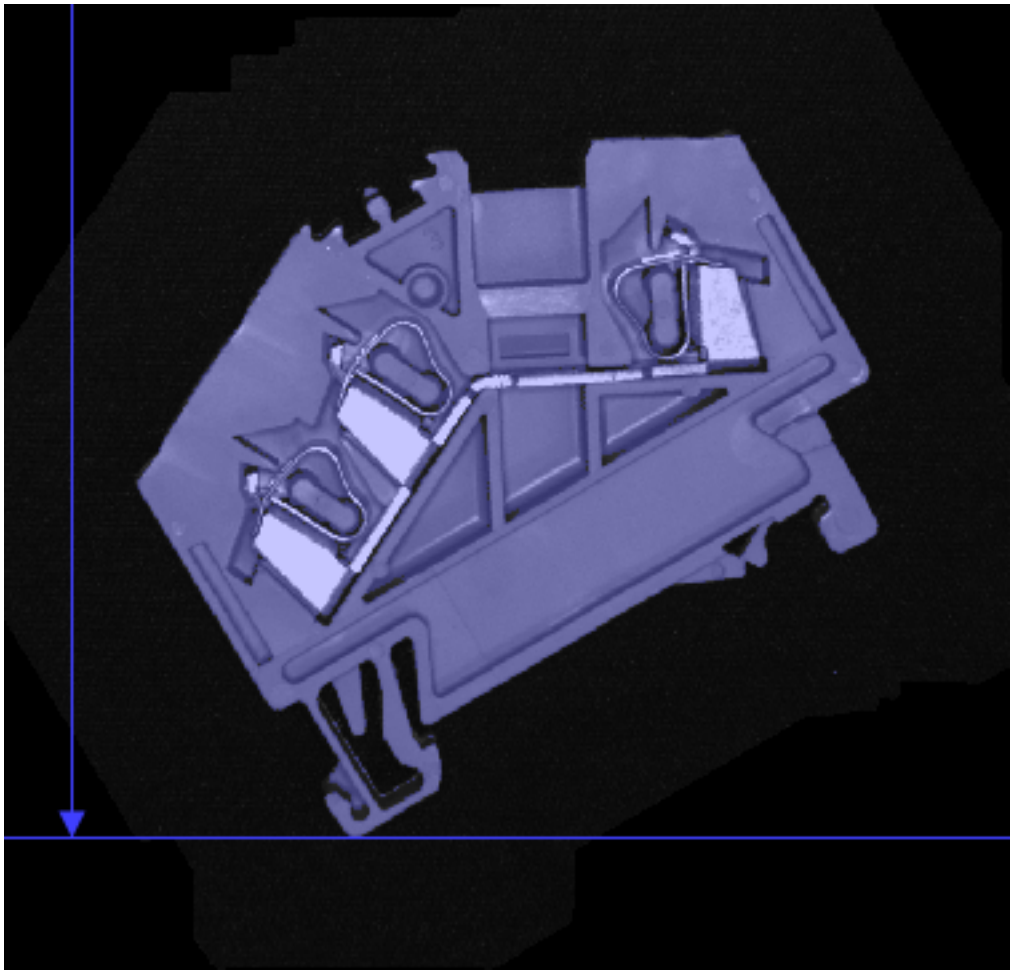


Abb. 12: Die maximale vertikale Koordinate.

3.4.7 Linke Seite

Dies berechnet die minimale x-Koordinate (einschließlich) der Region.

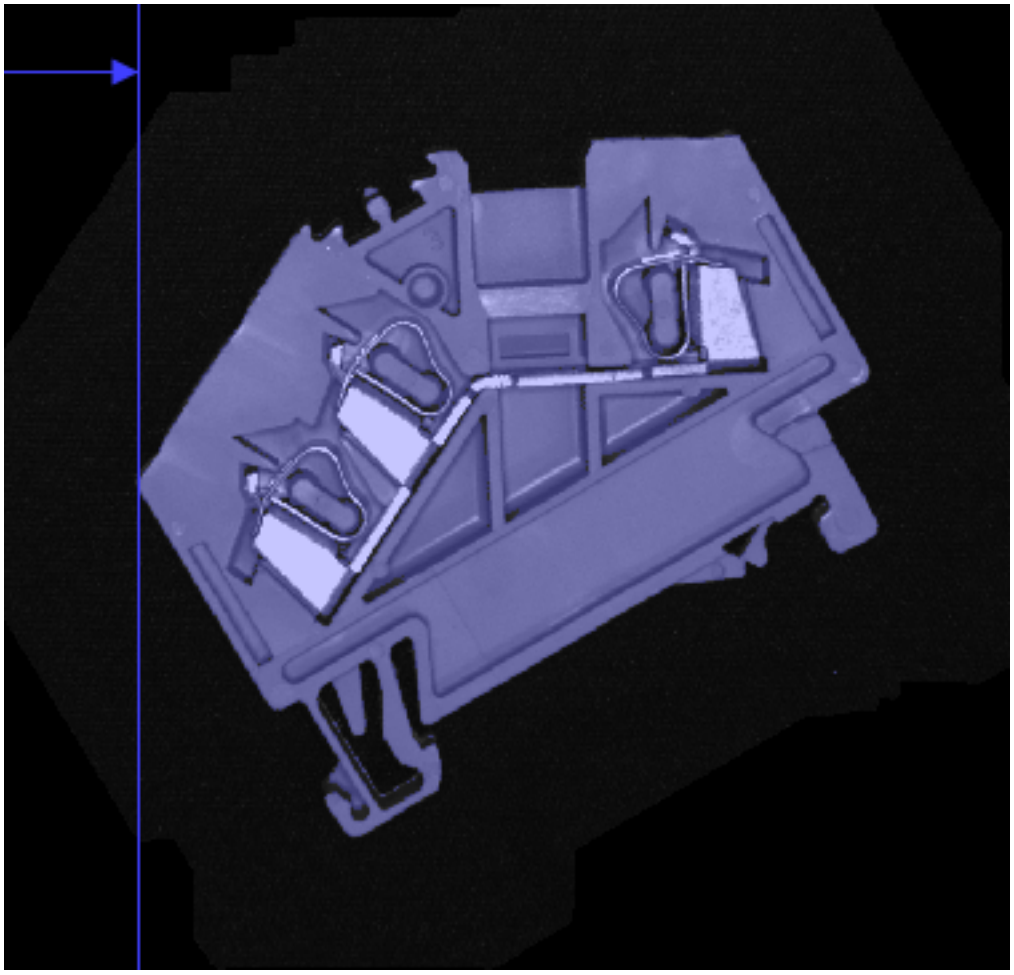


Abb. 13: Die minimale horizontale Koordinate.

3.4.8 Geometrische Mitte

Berechnet die Mitte des umschließenden Rechtecks.

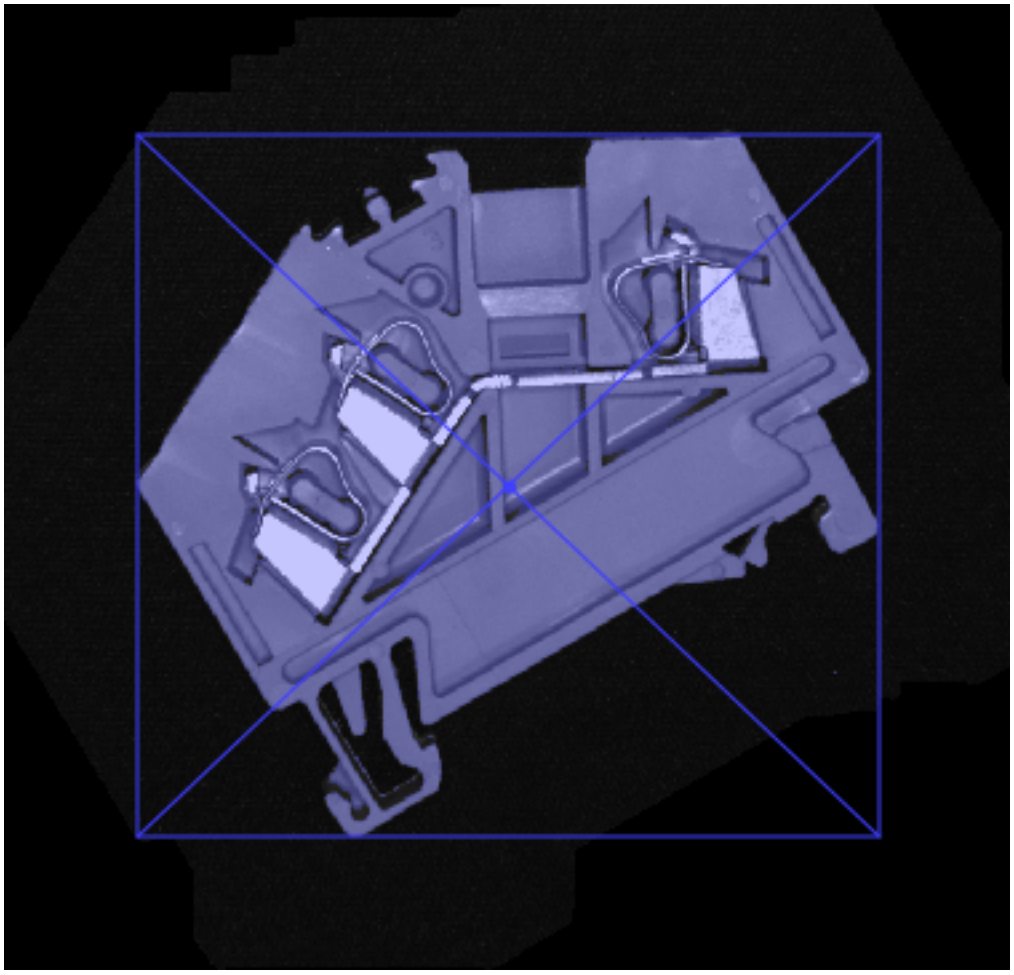


Abb. 14: Die geometrische Mitte.

3.4.9 Höhe

Dadurch wird die achsparallele Höhe des Bereichs berechnet.

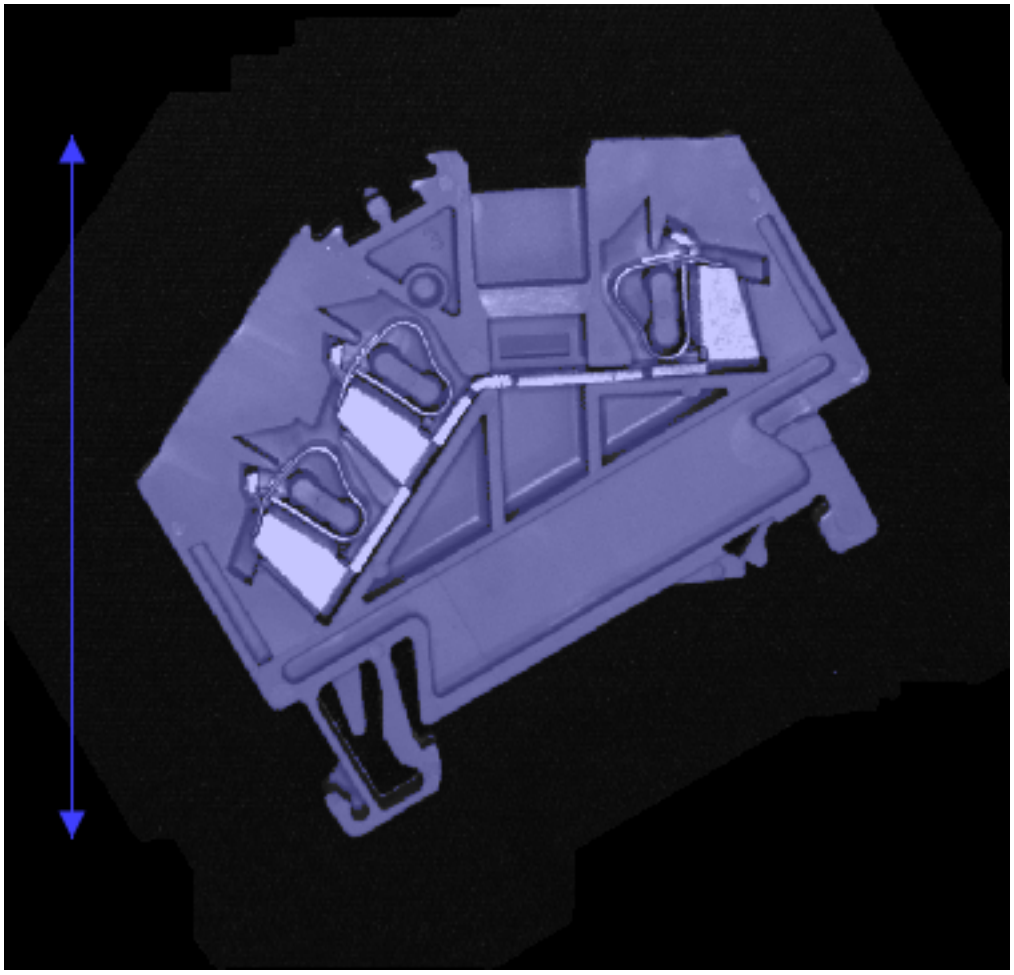


Abb. 15: Die Höhe eines Objekts.

3.4.10 Breite

Dadurch wird die achsparallele Breite des Bereichs berechnet.

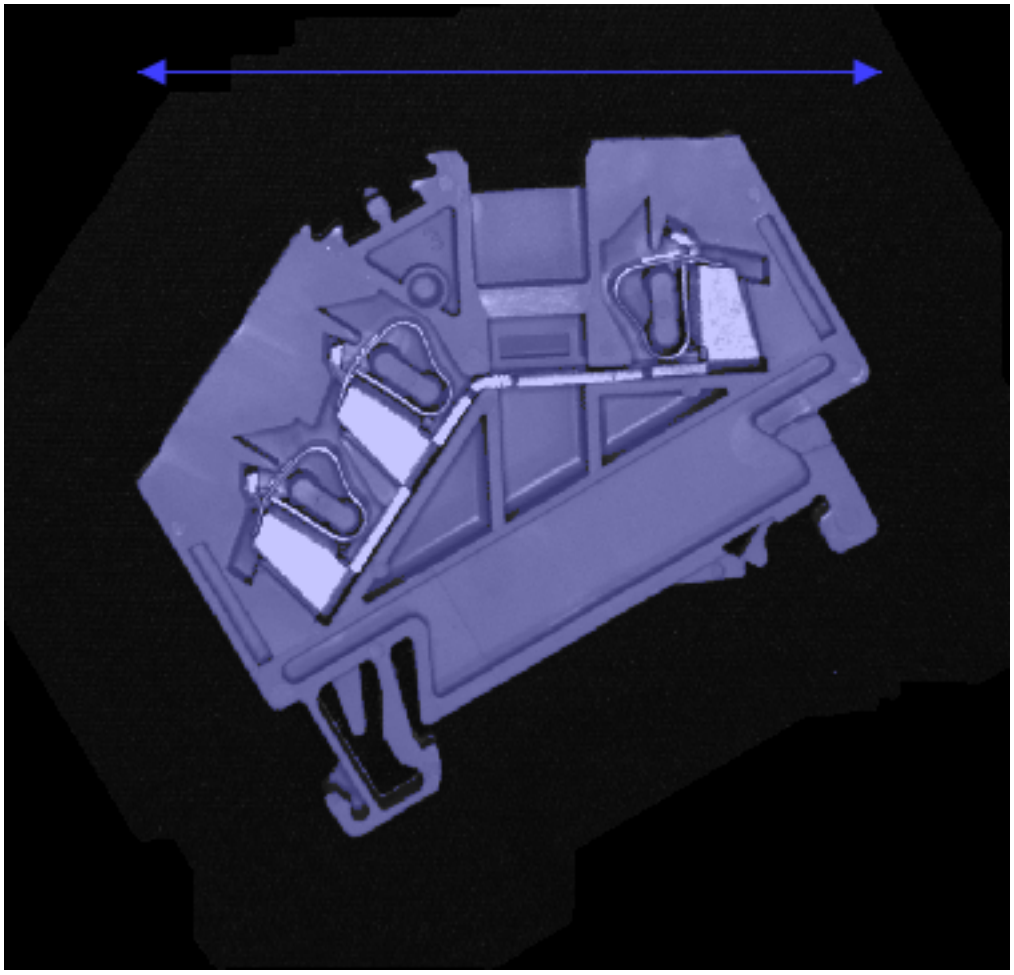


Abb. 16: Die Breite eines Objekts.

3.4.11 Umfang

Der Umfang einer Region gibt die Länge des Objektumrisses an.

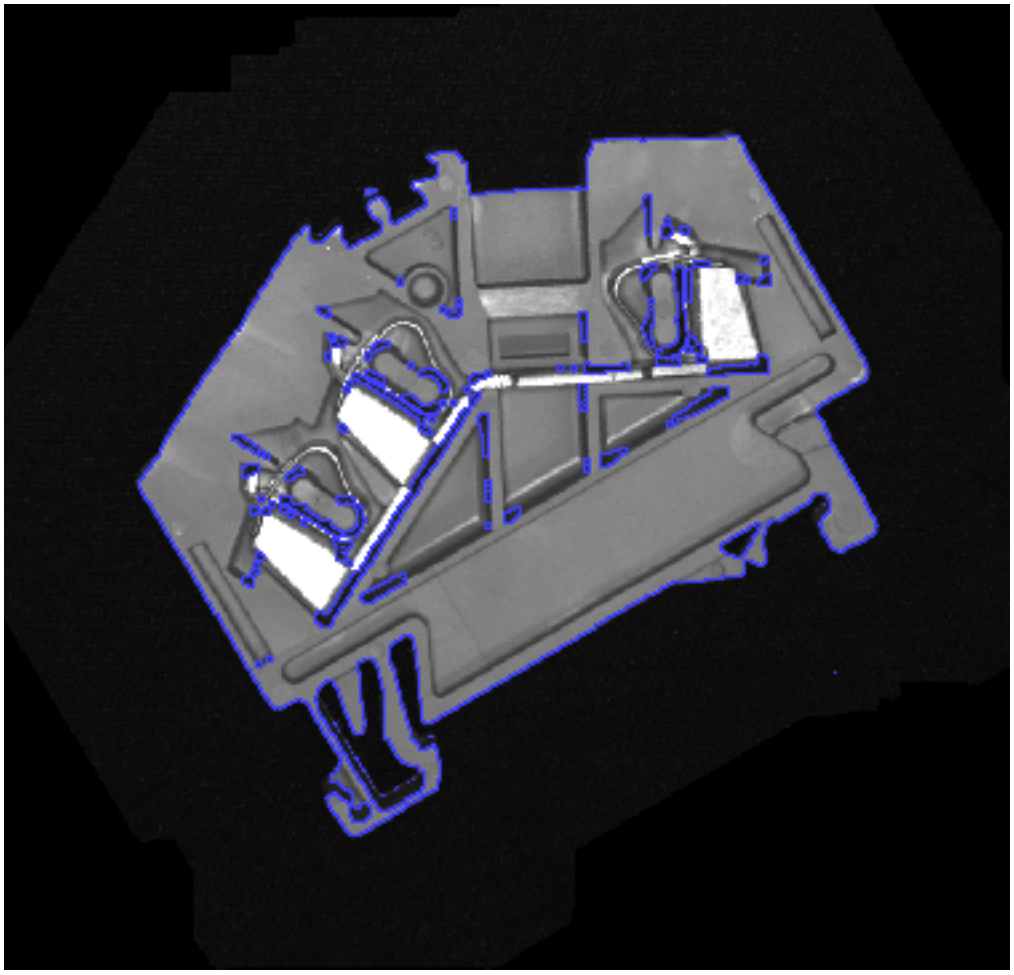


Abb. 17: Der Umfang eines Objekts.

Bei manchen Objekten, je nach Rauschen in der Bildaufnahme, kann sich ein hoher Umfang ergeben, wenn die Umrißlinie zerklüftet ist. Mit dem Parameter **Toleranz** kann dieser Umriß entsprechend geglättet werden, sodass sich ein realistischerer Umfang ergibt.

3.4.12 Fläche der Löcher

Berechnet die Gesamtfläche der Löcher in der Region.

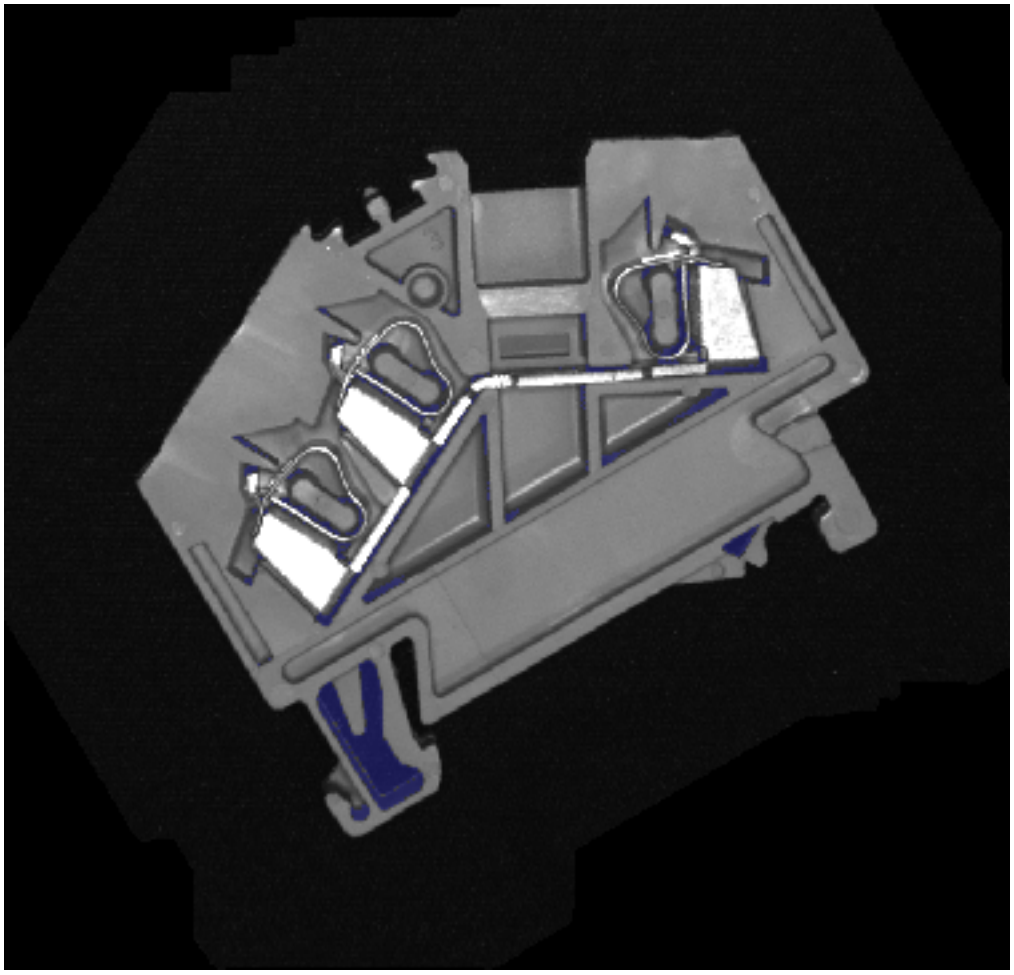


Abb. 18: Die Fläche der Löcher in einem Objekt.

3.4.13 Konvexe Fläche

Berechnet die Fläche, die von der konvexen Hülle eingeschlossen wird.

Die konvexe Fläche ist immer gleich groß oder größer als die Fläche der ursprünglichen Region.

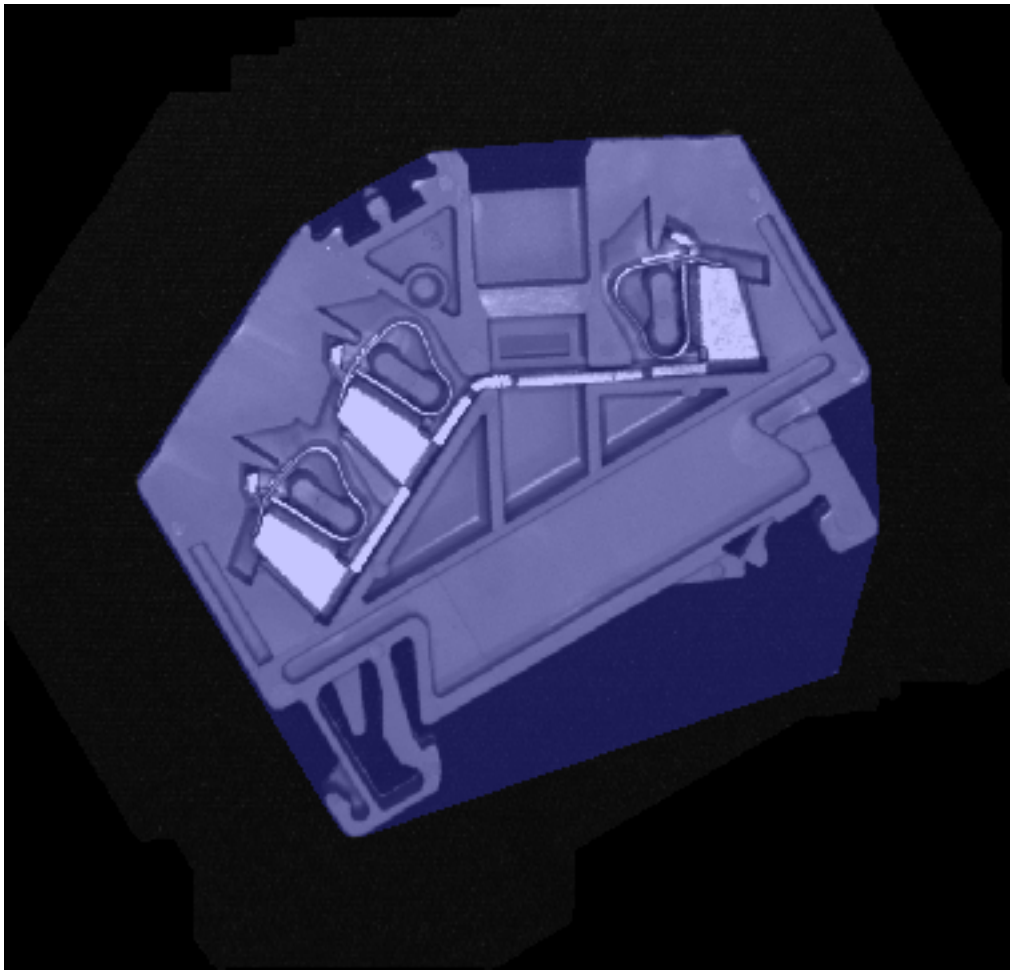


Abb. 19: Die konvexe Fläche eines Objekts.

3.4.14 Konvexer Umfang

Dadurch wird der Umfang des konvexen Hüllenpolygons berechnet.

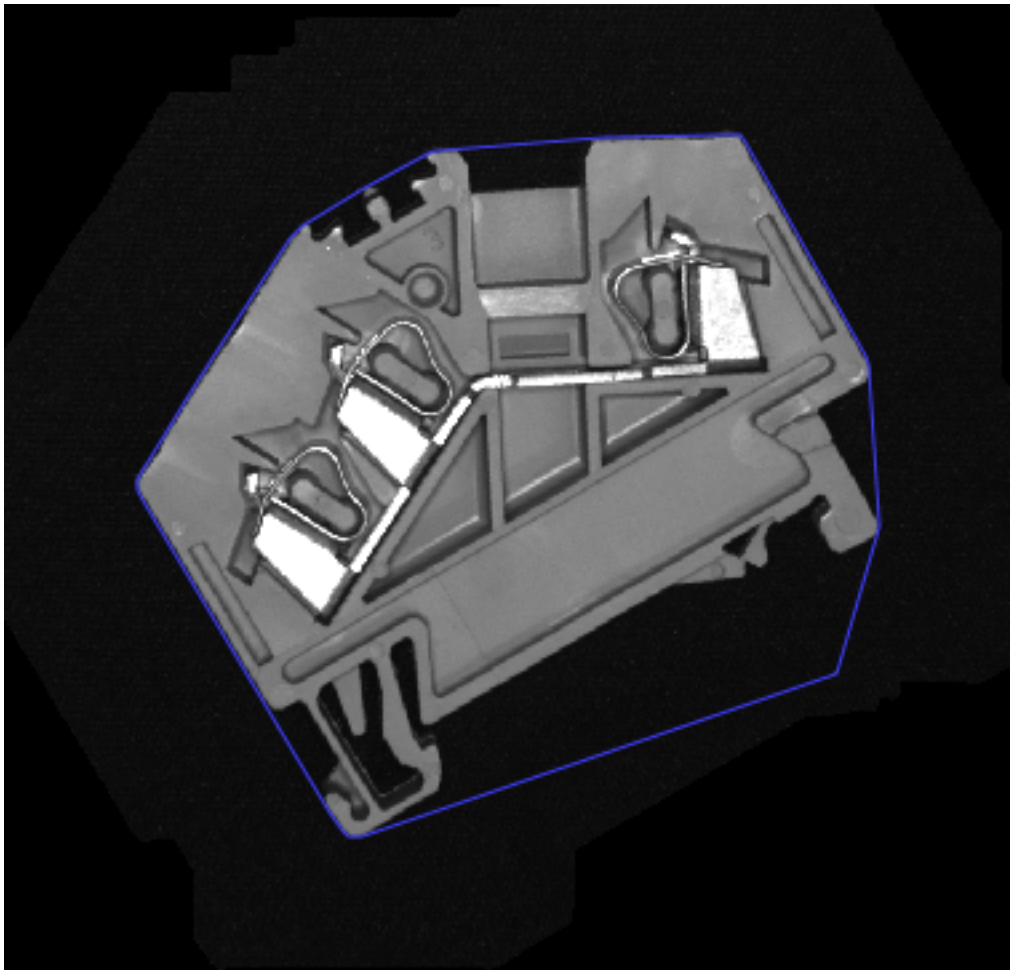


Abb. 20: Der konvexe Umfang eines Objekts.

3.4.15 Faserlänge

Dies berechnet eine Näherung einer Faserlänge. Diese Messung gilt nur für längliche Objekte. Die Faserlänge wird angenähert als der halbe Umfang.

Bei manchen Objekten, je nach Rauschen in der Bildaufnahme, kann sich ein hoher Umfang ergeben, wenn die Umrißlinie zerklüftet ist. Mit dem Parameter **Toleranz** kann dieser Umriß entsprechend geglättet werden, sodass sich ein realistischerer Umfang ergibt.

3.4.16 Faserbreite

Dies berechnet eine Näherung einer Faserbreite. Die Faserbreite ist angenähert als das Doppelte des Maximalwerts der Abstandstransformation.

3.4.17 Hauptachsenlänge

Die Länge der Hauptachse der äquivalenten Ellipse.

3.4.18 Nebenachsenlänge

Die Länge der Nebenachse der äquivalenten Ellipse.

3.4.19 Seitenverhältnis

Dies berechnet das Verhältnis von Breite zu Höhe nach der folgenden Formel:

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

wobei w die Breite und h die Höhe ist.

3.4.20 Zirkularität

Hier wird die Kreisförmigkeit der Region berechnet. Die Zirkularität ist definiert als der Quotient aus des Umfangs eines Kreises mit der gleichen Fläche wie das Objekt und dem Umfang des Objekts. Die Kreisförmigkeit c wird nach der folgenden Formel berechnet:

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

wobei a die Fläche und p der Umfang ist.

3.4.21 Kompaktheit

Hier wird die Kompaktheit der Region berechnet.

Die Kompaktheit c wird nach der folgenden Formel berechnet:

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

wobei a die Fläche und p der Umfang ist.

3.4.22 Konvexität

Dies berechnet die Konvexität der Region.

Die Konvexität c errechnet sich nach dieser Formel:

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

wobei a die Fläche und a_c die konvexe Fläche ist.

Die Konvexität ist 1 für einen Bereich, der bereits konvex ist. Sie ist kleiner als 1 für nicht konvexe Bereiche. Die Konvexität ist skaleninvariant.

3.4.23 Anzahl der Löcher

Dadurch wird die Anzahl der Löcher der Region berechnet.

3.4.24 Perforation

Dadurch wird die Perforation der Region berechnet.

Die Perforation *p* errechnet sich nach dieser Formel:

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

wobei a die Fläche und a_h die Fläche der Löcher ist.

Die Perforation ist Null für einen Bereich, der keine Löcher hat, und nimmt zu je mehr Löcher die Region hat. Die Perforation ist maßstabsinvariant.

3.4.25 Rauheit

Dies berechnet die Rauheit der Region. Die Rauheit ist definiert als der Quotient aus der Umfang des Objekts und der Umfang der konvexen Hülle. Die Rauheit ist berechnet nach folgender Formel:

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

wobei p der Umfang und p_c der Umfang der konvexen Hülle ist.

3.4.26 Rundheit

Dies berechnet die Rundheit der Region. Die Rundheit ist als Quotient von definiert die Fläche des Objekts und die Fläche des minimalen Begrenzungskreises. Die Rundheit ist berechnet nach folgender Formel:

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

wobei a die Fläche und a_c die Fläche des minimalen Begrenzungskreises ist.

3.4.27 Sphärizität

Dadurch wird die Sphärizität der Region berechnet. Die Sphärizität ist definiert als der Quotient von der Durchmesser eines Kreises mit der gleichen Fläche wie das Objekt und der maximale Fert-Durchmesser des Objekts. Die Sphärizität wird nach folgender Formel berechnet:

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

wobei a die Fläche und f_{max} der maximale Feret-Durchmesser ist.

3.5 Strichcode-Dekodierung

Der Strichcode Dekoder unterstützt die wichtigen linearen und zwei-dimensionalen Symbologien.

Folgende Symbologien werden unterstützt:

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 Zwei-dimensional

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

3.5.3 Dekoder Merkmale

Der Dekoder wird in drei verschiedenen Werkzeugen verwendet: **Barcode Lokalisieren**, **Barcode Erkennen** und **Barcodes Zählen**. Die drei Werkzeuge sind in großen Teilen ähnlich, unterscheiden sich aber im Detail:

Barcode Lokalisieren kann einen Code einer einzelnen Symbologie detektieren und liefert sein Lage zum Zweck der Nachführung nachgelagerter Werkzeuge. Alle Parameter der Symbologie können detailliert eingestellt werden.

Barcode Erkennen kann einen Code einer einzelnen Symbologie detektieren. Alle Parameter der Symbologie können detailliert eingestellt werden.

Barcode Zählen kann mehrere Codes verschiedener Symbologien detektieren. Dabei sind nicht alle Parameter detailiiert einstellbar, sondern es werden Standard-Werte verwendet.

Der Dekoder kann die folgenden Merkmale berechnen:

Code

Der im Strichcode kodierte Inhalt.

Symbologie

Die für den jeweiligen Code verwendete Symbologie.

Mitte (px)

Der Mittelpunkt des dekodierten Codes in Pixel-Koordinaten.

Im Logik-Werkzeug werden die Koordinaten einzeln als **MitteX (px)** bzw. **MitteY (px)** angesprochen.

Mitte (mm)

Der Mittelpunkt des dekodierten Codes in mm. Dieser Wert ist nur verfügbar, wenn eine Kalibrierung durchgeführt und angewandt wurde.

Im Logik-Werkzeug werden die Koordinaten einzeln als **MitteX (px)** bzw. **MitteY (px)** angesprochen.

Winkel

Der Winkel des Codes in Bezug auf die Lage der ROI.

Größe

Die Breite und Höhe des Codes in Pixeln.

Im Logik-Werkzeug werden die Größen einzeln als **Breite** und **Höhe** angesprochen.

Zellengröße

Die Zellenanzahl in horizontaler und vertikaler Richtung (nur für 2D Symbologien).

Im Logik-Werzeug werden die Größen einzeln als **Zellenreite** und **Zellenöhe** angesprochen.

Pixel pro Modul

Die Anzahl der Pixel pro Modul (nur für 2D Symbologien).

Quality (UEC)

Ein Qualitätsmaß auf Basis der ungenutzen Bits für Fehlerkorrektur (nur für 2D Symbologien).

3.5.4 Direkt-markierte Codes

Der Dekoder kann einfache direkt-markierte Codes dekodieren. Für besonders schwierige Markierungen gibt es ein optionales ID-Pro Modul mit zusätzlichen Dekodier-Optionen.

Die DPM Einstellungen haben nur bei Data Matrix und bei QR Codes eine Auswirkung.

3.5.5 Verifikation

Das optionale ID-Pro Modul ermöglicht auch die Verifizierung von Codes, angelehnt an die Normen ISO/IEC 15415 bzw. ISO/IEC TR29158.

Verifikation ist nur für Data Matrix und für QR Codes definiert.

Der Dekoder kann die folgenden Merkmale detektieren:

Bewertung	Beschreibung
Gesamtbewertung	Die Gesamtnote ist die schlechteste aller anderen Noten.
Achsen-Ungleichheit	Bewertet die ungleichmäßige Skalierung des Codes.
Kontrast	Bewertet den Unterschied zwischen den dunkelsten und hellsten Modulen.
Musterfehler	Beurteilt Fehler in L-Form, Taktmuster und Ruhezone.
Gitter-Ungleichheit	Bewertet die Abweichung vom Raster.
Modulation	Beurteilt lokale Kontrastabweichungen.
Referenz-Dekodierung	Der Referenz-Dekodierungsalgorithmus: 4 für Erfolg, andernfalls 0.
Reflexions-Abstand	Bewertet, wie gut jedes Element korrekt unterscheidbar ist.
Ungenutzte Fehlerkorrektur	Bewertet nicht verwendete Bits zur Fehlerkorrektur.

Die Grades sind ganze Zahlen von 0 (schlecht) bis 4 (gut). Alternativ werden auch Buchstaben zur Bezeichnung verwendet: A = 4, B = 3, C = 2, D = 1 und F = 0, entsprechend dem amerikanischen Notensystem.

3.5.6 GS1 Application Identifier

Das optionale ID-Pro Modul ermöglicht auch die Verwendung von GS1 Application Identifiern.

In der modernen Logistik und im Supply Chain Management reicht es oft nicht aus, nur zu wissen, welches Produkt vor einem liegt (identifiziert durch die GTIN). Man muss oft wissen: Wann läuft es ab? Welche Charge ist es? Wie schwer ist es?

Hier kommen die GS1 Application Identifier (AI) ins Spiel.

Application Identifier sind numerische Präfixe (in der menschenlesbaren Form meist in Klammern dargestellt, z. B. (10)), die einem Barcode-Scanner oder einem IT-System mitteilen, welche Art von Daten direkt danach folgen und wie diese Daten zu interpretieren sind.

Sie sind der Standard, der eine bloße Ziffernfolge in strukturierte, nutzbare Informationen verwandelt.

Basierend auf den GS1-Spezifikationen lassen sich die Ziele der AIs in drei Hauptbereiche unterteilen:

1. Eindeutige Datendefinition (Semantik)

Ein Scanner liest eine lange Kette von Zeichen. Ohne AIs wüsste das System nicht, wo die Artikelnummer aufhört und wo das Verfallsdatum beginnt.

Beispiel: Der AI (17) sagt dem System: “Die nächsten 6 Ziffern sind ein Verfallsdatum im Format JJMMTT.”

Zweck: Fehlerfreie Interpretation von Daten über Unternehmens- und Ländergrenzen hinweg.

2. Verschlüsselung mehrerer Informationen (Verkettung)

Dank der AIs können mehrere unterschiedliche Informationen in einem einzigen Barcode (z. B. GS1-128 oder GS1 DataMatrix) untergebracht werden.

Zweck: Platzersparnis auf dem Etikett und Effizienzsteigerung, da nur einmal gescannt werden muss, um Artikelnummer, Charge und Datum zu erfassen.

3. Rückverfolgbarkeit und Sicherheit

In den Spezifikationen von 2025 liegt ein massiver Fokus auf Traceability (Rückverfolgbarkeit). AIs sind das technische Rückgrat dafür.

Zweck: Durch die Kombination von Produkt-ID (01) mit einer Chargennummer (10) oder einer Seriennummer (21) kann jedes einzelne Produkt im Falle eines Rückrufs präzise identifiziert werden.

Die GS1 General Specifications legen strenge Regeln für die Datenstruktur fest:

Einige AIs haben eine feste Länge (z. B. hat das Datum bei AI 17 immer genau 6 Stellen). Das System weiß automatisch, wann das Feld zu Ende ist.

Andere AIs haben eine variable Länge (z. B. die Chargennummer bei AI 10, die bis zu 20 Zeichen haben kann). Hier wird ein spezielles Trennzeichen (FNC1) benötigt, um das Ende des Datenfeldes zu markieren, wenn noch weitere Daten folgen.

Numerisch vs. Alphanumerisch: Die Spezifikation definiert genau, ob nach einem AI nur Zahlen oder auch Buchstaben folgen dürfen.

Obwohl es eine Vielzahl verschiedene AIs gibt, decken diese wenigen Beispiele einen Großteil der täglichen Anwendungen ab:

AI	Bedeutung	Datenformat	Beispiel-Anwendung
00	NVE / SSCC (Nummer der Versandeinheit)	18 Ziffern	Identifikation einer ganzen Palette im Transport
01	GTIN (Global Trade Item Number)	14 Ziffern	Identifikation der Produktart (z.B. Dose Cola)
10	Chargennummer / Losnummer	Variabel (bis 20)	Rückverfolgbarkeit bei Produktionsfehlern
17	Verfallsdatum	6 Ziffern (JJMMTT)	Lebensmittelsicherheit, Lagersteuerung (FEFO)
21	Seriennummer	Variabel (bis 20)	Identifikation eines einzelnen Geräts (z.B. Smartphone)
310x	Nettogewicht (in kg)	6 Ziffern	Gewichtsbasierte Abrechnung bei Fleisch/Käse

Eioe vollständige Spezifikation der GS1 Application Identifier gibt es hier: https://www.gs1.org/standards/barcodes-epcrfid-id-keys/gs1-general-specifications

Die Implementierung erfolgt in spezifischen Knoten, die in der Logik verwendet werden können. Die Knoten trennen einen Text in eine Liste von Application Identifiern, die entsprechend weiterverarbeitet werden können.

3.6 Optische Zeichenerkennung (OCR)

Optische Zeichenerkennung ist der Fachbegriff für die Identifizierung von Zeichen, d. h. das Lesen von Text.

Nach der Segmentierung der Symbole in Regionen, die die Zeichen darstellen, wird ein Klassifikator eingesetzt, um die Interpretation eines jeden Zeichens zu erkennen. Es wird eine Reihe von Schriftarten bereitgestellt, die auf einer großen Menge von Trainingsdaten aus verschiedenen Anwendungsbereichen beruhen. Diese Schriftarten ermöglichen das Lesen von Text in Dokumenten, auf pharmazeutischen oder industriellen Produkten, Punktdrucken und sogar handgeschriebenen Zahlen.

Unser OCR-Leser geht von einem annähernd horizontalen Text aus. Wenn der Text nicht waagerecht ausgerichtet ist, sollte die ROI so gedreht werden, dass der Text innerhalb der ROI in Bezug auf die Ausrichtung der ROI horizontal ist.

Es gibt mehrere Gruppen von Schriften. Die Mitglieder der einzelnen Gruppen unterscheiden sich, da sie unterschiedliche Symbolsätze enthalten. Der Inhalt einer Schriftart wird durch ihren Namen beschrieben. Für die Namen der vortrainierten Schriften gilt die folgende Nomenklatur:

Der Name beginnt mit dem Gruppennamen, z. B. **Document** oder **DotPrint**, gefolgt von Indikatoren für den Symbolsatz die in der OCR-Schriftart enthalten sind. Die Bedeutung der Indikatoren ist die folgende:

- **0-9**: Die OCR-Schriftart enthält die Ziffern 0 bis 9.

- **A-Z**: Die OCR-Schriftart enthält die Großbuchstaben A bis Z.
- **+** : Die OCR-Schriftart enthält Sonderzeichen. Die Liste der Sonderzeichen variiert leicht zwischen den einzelnen OCR-Schriften. Sie ist im Folgenden für jede OCR-Schriftart einzeln aufgeführt.

Wenn der Name der OCR-Schriftart keines der oben genannten Kennzeichen enthält, enthält die OCR-Schriftart normalerweise die Ziffern 0 bis 9, die Großbuchstaben A bis Z, die Kleinbuchstaben Kleinbuchstaben a bis z und Sonderzeichen. Einige OCR-Schriften enthalten keine Kleinbuchstaben (z. B., DotPrint). Dies wird in der Beschreibung der jeweiligen Schriftart ausdrücklich erwähnt.

3.6.1 `Document`

Die Schriftart `Document` kann verwendet werden, um Zeichen zu lesen, die in Schriftarten wie Arial, Courier oder Times New Roman gedruckt wurden. Dies sind typische Schriftarten für den Druck von Dokumenten oder Briefen.

Beachten Sie, dass die Zeichen I und l der Schriftart Arial nicht unterschieden werden können. Das bedeutet, dass ein `l` fälschlicherweise für ein `I` gehalten werden kann und umgekehrt.

Verfügbare Sonderzeichen: - = + < > . # $ % & ( ) @ * e £ ¥

3.6.2 `DotPrint`

Die OCR-Schriftart DotPrint kann verwendet werden, um mit Punktdruckern gedruckte Zeichen zu lesen. Sie enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . * :

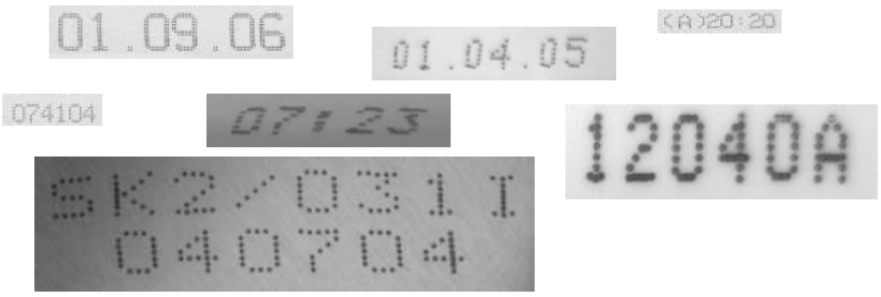


Abb. 21: Beispiele für gepunktete Schrift.

3.6.3 `HandWritten`

Die Schriftart HandWritten_0-9 kann zum Lesen handgeschriebener Zahlen verwendet werden.

Sie enthält die Ziffern 0-9.

Verfügbare Sonderzeichen: keine

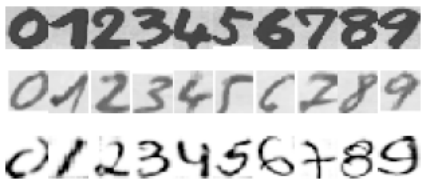


Abb. 22: Beispiele für handgeschriebene Ziffern.

3.6.4 Industrial

Die Schriftart Industrial kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B oder anderen serifenlosen Schriftarten gedruckt wurden. Diese Schriftarten werden normalerweise zum Drucken von z. B. Etiketten verwendet.

Verfügbare Sonderzeichen: - / + . $ % * e £ ¥

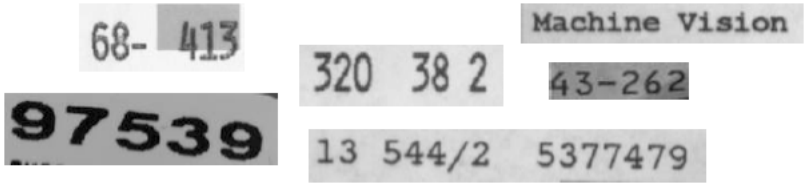


Abb. 23: Beispiele für industrielle Drucke.

3.6.5 OCR-A

Die Schrift OCR-A kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-A gedruckt sind.

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



Abb. 24: Ausgewählte Zeichen der OCR-A Schrift.

3.6.6 OCR-B

Die Schrift OCR-B kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-B gedruckt sind.

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



Abb. 25: Ausgewählte Zeichen der OCR-B Schrift.

3.6.7 Pharma

Die Schriftart Pharma kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B und anderen Schriftarten gedruckt sind, die typischerweise in der pharmazeutischen Industrie verwendet werden.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . ( ) :

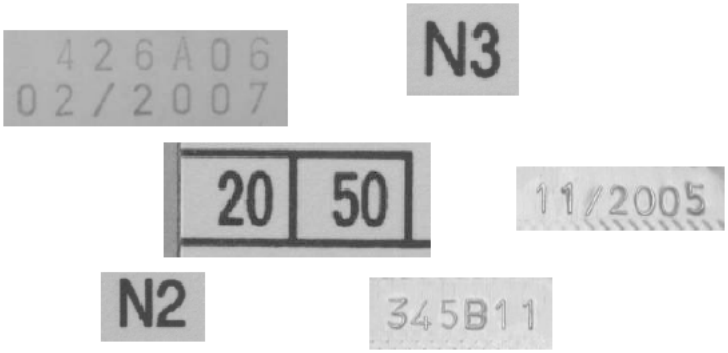


Abb. 26: Beispiele für pharmazeutische Drucke.

3.6.8 SEMI

Die Schriftart SEMI kann verwendet werden, um Zeichen zu lesen, die in der Schriftart SEMI gedruckt sind, die aus Zeichen besteht, die die so gestaltet sind, dass sie sich leicht voneinander unterscheiden lassen. Sie hat einen begrenzten Satz von Zeichen.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - .



Abb. 27: Zeichensatz der SEMI Schriftart.

3.6.9 Universal

Die OCR-Schriftart Universal kann zum Lesen einer Vielzahl unterschiedlicher Zeichen verwendet werden. Das Training dieser CNN-basierten Schriftart basiert auf den Zeichensätzen 'Document', 'DotPrint', 'SEMI' und 'Industrial'.

Verfügbare Sonderzeichen: - / = + : < > . # $ % & ( ) @ * e £ ¥

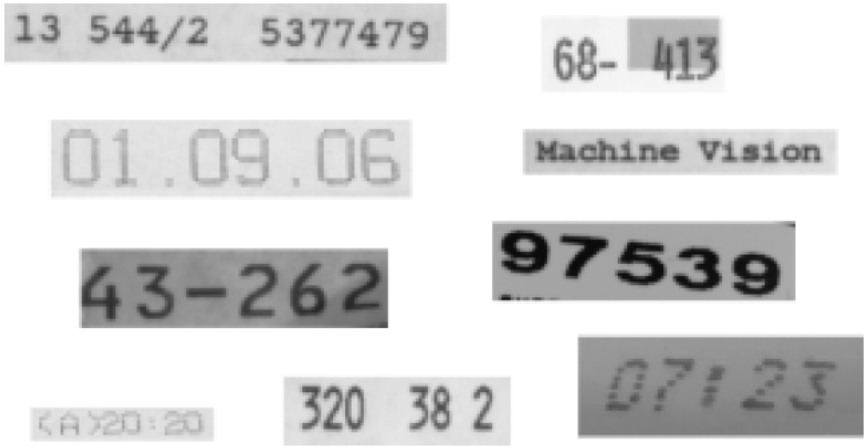


Abb. 28: Beispiele für den Zeichensatz Universal.

KAPITEL 4

Referenz

4.1 Grafische Oberfläche

4.1.1 Ribbon

Der Ribbon zeigt die Kommandos zur Bedienung von **nVision-i**.

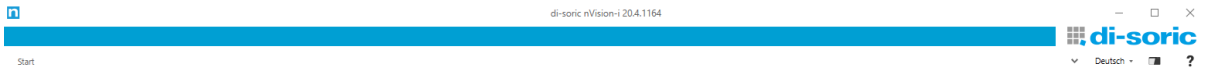


Abb. 1: Der nVision-i Ribbon

Auf der rechten Seite des Ribbon, unter dem **di-soric** Logo, sind drei Tasten und die Sprachauswahl.

Sprache

Sprache auswählen.

Theme

Umschalten des Themes zwischen Weiß, Hell und Dunkel.

Hilfe : F1

Anzeigen und Verbergen des Hilfe-Fensters.

Start

Wenn **nVision-i** mit einem Gerät verbunden ist und ein Job zum Editieren geöffnet ist, zeigt der **Home** Tab die verfügbaren Kommandos:



Abb. 2: Der Ribbon

Job Schließen

Zurück zur Startseite.

Job Speichern

Job dauerhaft speichern.

Bild Speichern : Strg-S

Bild speichern.

Werkzeug löschen : Del

Das gewählte Werkzeug entfernen.

Trigger ——-

Übersteuern

Triggerung kontrollieren.

Freilaufend

Die Bildaufnahme kontinuierlich triggern.

Einmal triggern

Die Bildaufnahme einmal triggern.

Zoom

Ein-Zoomen : Num-Plus

In das Bild einzoomen.

Aus-Zoomen : Num-Minus

Aus dem Bild auszoomen.

Zoom Rücksetzen

Zoomfaktor zurücksetzen.

Lokalisieren

 **Fläche**

Das Fläche Lokalisieren Werkzeug einfügen.

 **Kante**

Das Kante Lokalisieren Werkzeug einfügen.

 **Form**

Das Form Lokalisieren Werkzeug einfügen.

 **Barcode**

Das Barcode Lokalisieren Werkzeug einfügen.



Text

Das Text Lokalisieren Werkzeug einfügen.

Erkennen

 **Helligkeit**

Das Helligkeit Erkennen Werkzeug einfügen.

 **Kontrast**

Das Kontrast Erkennen Werkzeug einfügen.

 **Flächen Pixel**

Das Flächenpixel Erkennen Werkzeug einfügen.

 **Kanten Pixel**

Das Kantenpixel Erkennen Werkzeug einfügen.

 **Barcode**

Das Barcode Erkennen Werkzeug einfügen.



Text

Das Text Erkennen Werkzeug einfügen.

Zählen

Flächen

Das Flächen Zählen Werkzeug einfügen.

Kanten

Das Kanten Zählen Werkzeug einfügen.

Formen

Das Formen Zählen Werkzeug einfügen.

Barcodes

Das Barcodes Zählen Werkzeug einfügen.

Zeichen

Das Zeichen Zählen Werkzeug einfügen.

Messen

Winkel

Das Winkel Messen Werkzeug einfügen.

Kreis

Das Kreis Messen Werkzeug einfügen.

Distanz

Das Distanz Messen Werkzeug einfügen.

Punkt zu Punkt

Das Punkt zu Punkt Messen Werkzeug einfügen.

Punkt zu Linie

Das Punkt zu Linie Messen Werkzeug einfügen.

Statistik

Job Testen

Stellt Statistiken zu einem Job bereit.

4.1.2 Statuszeile

Die Statuszeile zeigt Informationen über den Sensor und einen Job.



Abb. 3: Die Statuszeile

Auf der linken Seite der Statuszeige werden die IP Adresse und der Name des Sensors, sowie der Name und der Slot des Jobs angezeigt.

In der Mitte der Statuszeile wird die Position des Mauszeigers in Pixeln und der Pixelwert unter dem Mauszeiger sowie seine Graustufe angezeigt.

Mit dem Schieberegler auf der rechten Seite der Statuszeile kann die Applikation gezoomt werden. Zoomen Sie auf 50 %, um mehr zu sehen. Für die Nutzung von nVision-i auf Touch Bildschirmen können Sie bis auf 200 % zoomen.

4.1.3 Job Testen



Abb. 4: Das Icon des Kommandos **Job Testen**.

Mit Kommando **Job Testen** kann die Ausführungszeit auf dem Sensor gemessen werden. Die Ausführungszeit bei verbundenem nVision-i ist aufgrund des Kommunikations-Overheads immer etwas höher als bei einem freilaufenden Sensor.

Einstellungen

Der **(Neu) starten** Button startet die Datensammlung für die rechts angezeigte Statistik. So lange diese aktiv ist, wird jedes geprüfte Bild in die Statistik aufgenommen und angezeigt.

Als Standardeinstellung ist die Option **Frei Laufend** aktiviert. In diesem Zustand werden die eingestellten Trigger ignoriert und eine neue Bildaufnahme wird automatisch gestartet nachdem ein Bild ausgewertet wurde. Wird diese Option deaktiviert wird der Job mit den gegebenen Triggereinstellungen getestet. Je nach Einstellung erfolgt eine neue Auswertung in diesem Fall nach Eingang eines digitalen oder PROFINET Triggers.

Statistik

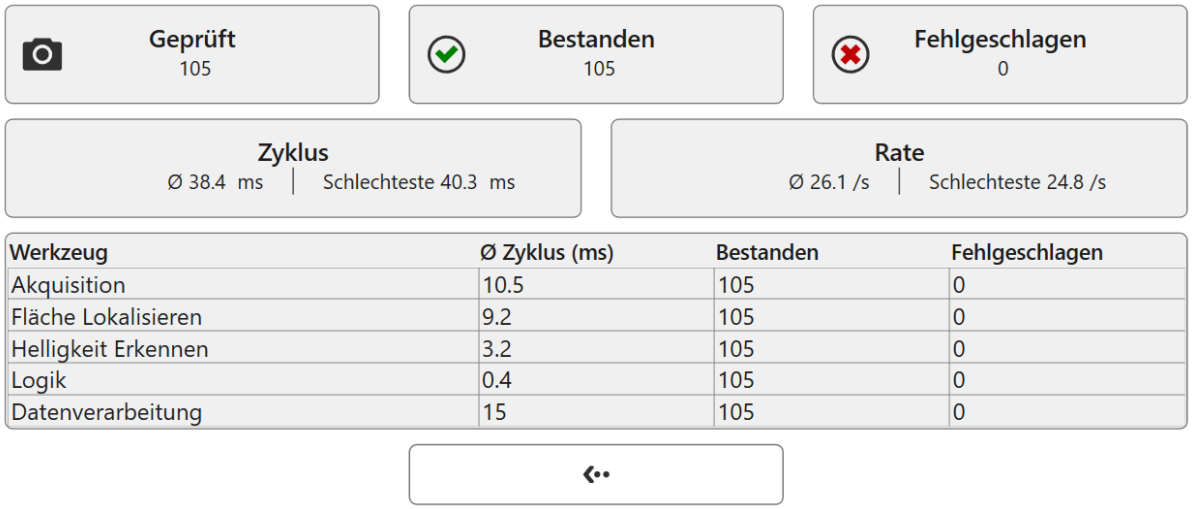


Abb. 5: Die Statistik des Werkzeugs **Job Testen**.

Auf der rechten Seite wird die aktuelle Statistik angezeigt.

In der ersten Zeile wird die Anzahl bisher ausgewerteter Bilder angezeigt, als auch die Information wie viele davon bei der Auswertung bestanden bzw. nicht bestanden haben.

In der zweiten Zeile wird die durchschnittliche als auch langsamste Zeit ausgegeben, die für die Inspektion der bisher geprüften Bilder aufgewendet wurde. Die **Zyklus**-Zeit gibt dabei die absolute Dauer in Millisekunden an. Die **Rate** gibt hingegen an, wie viele Inspektionen pro Sekunde ausgeführt werden können.

In der dritten Zeile werden die aktuell aktiven Werkzeuge separat aufgelistet. Hier wird pro Werkzeug eine durchschnittliche Zykluszeit als auch die absolute Anzahl an bestandenen, bzw. nicht bestandenen Inspektionen gelistet.

Mit dem Button in der vierten Zeile kann das **Job Testen** Werkzeug geschlossen werden. Die gesammelte Statistik wird nach dem Schließen zurückgesetzt und kann nicht erneut angezeigt werden.

4.2 Korrektur-Werkzeuge

4.2.1 Schattierung

Die Shading Korrektur kompensiert den Helligkeitsabfall in einem aufgenommenen Bild zum Rand hin (auch Vignettierung genannt). Die Abschattung zum Rand hin ist ein optischer Effekt, der mit den verwendeten Objektiven variiert, und auch durch die Beleuchtung hervorgerufen bzw. verstärkt wird. Eine Voraussetzung für eine optimale Shading Korrektur ist die orthogonale Betrachtung einer Szene (die Kamera muss unter einem rechten Winkel auf die Szene blicken).

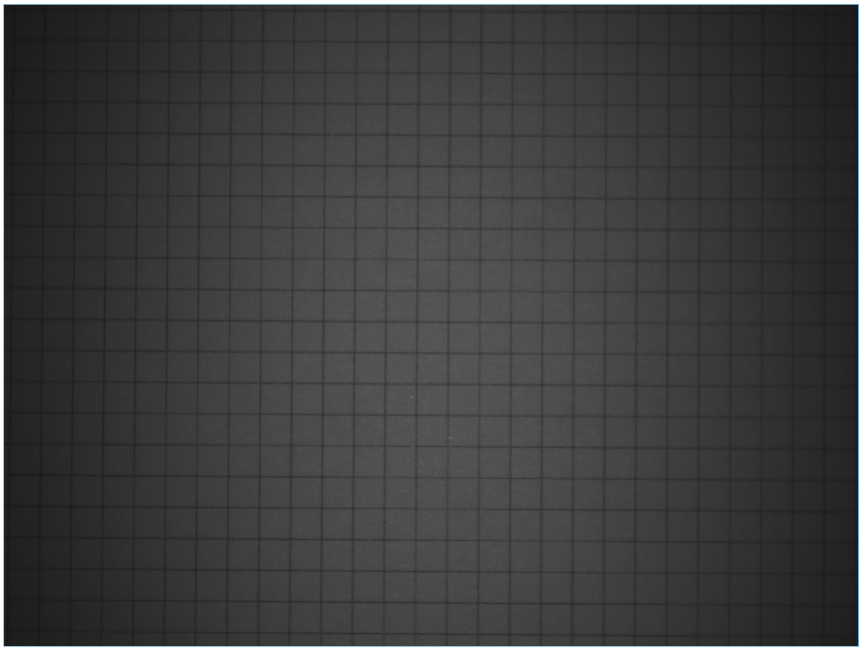


Abb. 6: Schattierung durch Randabschattung.

Die Kalibrierung sollte bei jedem Objektivwechsel bzw. bei einem Wechsel der Beleuchtung neu durchgeführt werden.

Für die Kalibrierung sollte ein möglichst gleichförmig helles Bild aufgenommen, wobei darauf zu achten ist, dass die Intensitäten weder zu dunkel noch zu hell sind. Das kann mit der Histogrammdarstellung bzw. mit den Funktionen **Unterbelichtung hervorheben** und **Überbelichtung hervorheben** aus dem Bereich **Visualisierung** erfolgen.

Wenn das Korrekturbild sowie die Parameter zur Belichtung und Beleuchtung korrekt eingestellt sind, kann mit der Taste **Korrigieren** die Korrektur berechnet werden.

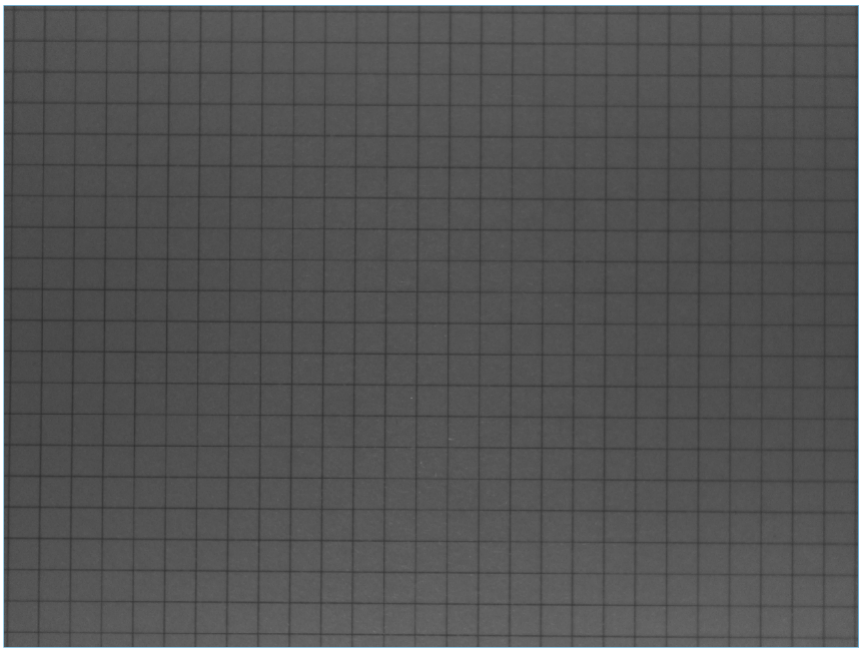


Abb. 7: Korrigierte Randabschattung.

Nach der erfolgten Berechnung kann die Korrektur mit **Korrektur speichern** gespeichert werden, damit sie in Jobs angewandt werden kann, bzw. durch die Taste **Reset** wieder gelöscht werden.

Parameter

Das Histogramm zeigt die Grauwertverteilung der erfassten Bilder an.

Belichtungszeit steuert die Belichtungsdauer eines einzelnen Bilds, beginnend bei 25 µs und bis zu 2000 µs. Bei deaktiviertem internen Blitz (unter **Blitzeinstellungen**, **Intern**) kann die **Belichtungszeit** bis zu einem Maximum von 500000 µs gewählt werden. Die Belichtungszeit beeinflusst die Helligkeit des aufgenommenen Bildes.

Verstärkung steuert die Verstärkung zwischen 1 (keine Verstärkung) und 16. Die Verstärkung wirkt linear auf die Helligkeit, d.h. ein mit der Verstärkung 2 aufgenommenes Bild hat doppelt so hohe Intensitätswerte, wie ein ohne Verstärkung aufgenommenes Bild.

Blitzeinstellungen

Der **CS-6x/ID-6x0** Vision Sensor verfügt über eine integrierte Beleuchtung. Wenn **Intern** auf **An** eingestellt ist, beleuchten vier Hochleistungs-LED die Szene. Die Beleuchtung wird mit der Bildaufnahme synchronisiert. Wenn **Intern** auf **Aus** eingestellt ist, sind die eingebauten LEDs aus und beleuchten die Szene nicht, beispielsweise wenn externe Beleuchtung verwendete wird. Die **Farbe** kann auf **Weiß** oder **Rot** eingestellt werden, um die jeweilige LED-Farbe zu verwenden.

Zusätzlich zum internen Blitz kann unter **Extern** auch ein digitaler Ausgang ausgewählt werden um eine externe Beleuchtung anzusteuern. Zur Auswahl stehen alle Digitalen Ausgänge, welche im **Logik** Werkzeug unter **Digitale Ausgänge** nicht verwendet werden (deren Eingang **Wert** nicht verbunden sind). Umgekehrt können Digitale Ausgänge, welche als externer Blitz gewählt wurden, im **Logik** Werkzeug nicht länger verwendet werden.

Kalibrierung

Wählen Sie eine gespeicherte Kalibrierung aus, um sie zu laden und zu bearbeiten.

Einstellungen

Mit **Glättung** kann eine Bildglättung zwischen **Scharf** und **Unscharf** gewählt werden, als Vorverarbeitungsschritt für die Berechnung der Shading Korrektur.

Visualisierung

Korrigiertes Bild zeigt ein in der Helligkeit korrigiertes Bild an.

Original Bild zeigt das ursprüngliche, nicht korrigierte Bild an.

Überbelichtung hervorheben zeigt überbelichtete Bildteile in gelber Farbe an.

Unterbelichtung hervorheben zeigt unterbelichtete Bildteile in blauer Farbe an.

4.2.2 Verzerrungs-Korrektur

Die Verzeichnungs-Korrektur kompensiert optische Verzerrungen, die durch verschiedene Objektive in verschiedenem Grad zum Rand hin verstärkt vorkommen. Eine Voraussetzung für eine optimale Verzerrungs-Korrektur ist die orthogonale Betrachtung einer Szene (die Kamera muss unter einem rechten Winkel auf die Szene blicken). Ein zweiter Effekt der Verzerrungs-Korrektur ist die Bestimmung der intrinsischen und extrinsischen Kamera-Parameter, insoweit sie für eine kalibrierte Messung einer orthogonalen Messebene nötig sind.

Die Kalibrierung muss bei jedem Objektivwechsel bzw. bei einem Wechsel der Mess-Ebene neu durchgeführt werden.

Für die Kalibrierung muss ein Bild eines punktförmigen Kalbrier-Targets möglichst bildfeldfüllend aufgenommen werden, und die Punktabstände eingegeben werden. Solche Targets sind käuflich zu erwerben, bzw. sie können für weniger genaue Anforderungen auch selbst mit einem Drucker ausgedruckt werden. Im Verzeichnis **Docs** der nVision-i Installation liegt eine PDF Datei zum Ausdrucken für verschiedene Bildfeldgrössen.

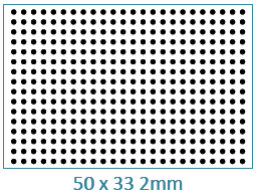


Abb. 8: Eine Vorlage für die Verzeichnungs-Korrektur und die geometrische Kalibrierung.

Das Kalibrierungs-Target muss in der Messebene angebracht werden (nur in dieser Mess-Ebene sind die Messwerte gültig). Bei den verwendeten endozentrischen Objektiven werden weiter entfernte Bildteile kleiner und näher liegende Bildteile größer abgebildet, d.h. die auch die Meßwerte werden entsprechend kleiner bzw. größer. Richtige Meßwerte werden nur in der Messebene ermittelt.

Nach der erfolgten Berechnung kann die Korrektur mit **Korrektur speichern** gespeichert werden, damit sie in Jobs angewandt werden kann, bzw. durch die Taste **Reset** wieder gelöscht werden.

Parameter

Das Histogramm zeigt die Grauwertverteilung der erfassten Bilder an.

Belichtungszeit steuert die Belichtungsdauer eines einzelnen Bilds, beginnend bei 25 µs und bis zu 2000 µs. Bei deaktiviertem internen Blitz (unter **Blitzeinstellungen**, **Intern**) kann die **Belichtungszeit** bis zu einem Maximum von 500000 µs gewählt werden. Die Belichtungszeit beeinflusst die Helligkeit des aufgenommenen Bildes.

Verstärkung steuert die Verstärkung zwischen 1 (keine Verstärkung) und 16. Die Verstärkung wirkt linear auf die Helligkeit, d.h. ein mit der Verstärkung 2 aufgenommenes Bild hat doppelt so hohe Intensitätswerte, wie ein ohne Verstärkung aufgenommenes Bild.

Blitzeinstellungen

Der **CS-6x/ID-6x0** Vision Sensor verfügt über eine integrierte Beleuchtung. Wenn **Intern** auf **An** eingestellt ist, beleuchten vier Hochleistungs-LED die Szene. Die Beleuchtung wird mit der Bildaufnahme synchronisiert. Wenn **Intern** auf **Aus** eingestellt ist, sind die eingebauten LEDs aus und beleuchten die Szene nicht, beispielsweise wenn externe Beleuchtung verwendete wird. Die **Farbe** kann auf **Weiß** oder **Rot** eingestellt werden, um die jeweilige LED-Farbe zu verwenden.

Zusätzlich zum internen Blitz kann unter **Extern** auch ein digitaler Ausgang ausgewählt werden um eine externe Beleuchtung anzusteuern. Zur Auswahl stehen alle Digitalen Ausgänge, welche im **Logik** Werkzeug unter **Digitale Ausgänge** nicht verwendet werden (deren Eingang **Wert** nicht verbunden sind). Umgekehrt können Digitale Ausgänge, welche als externer Blitz gewählt wurden, im **Logik** Werkzeug nicht länger verwendet werden.

Kalibrierung

Wählen Sie eine gespeicherte Kalibrierung aus, um sie zu laden und zu bearbeiten.

Verzerrung

Der **Schwellenwert** für die Detektion des Kalibrations-Targets kann **automatisch** bestimmt oder über einen Schieberegler eingestellt werden. Ziel ist die optimale Bestimmung der Punkte des Kalibriertargets.

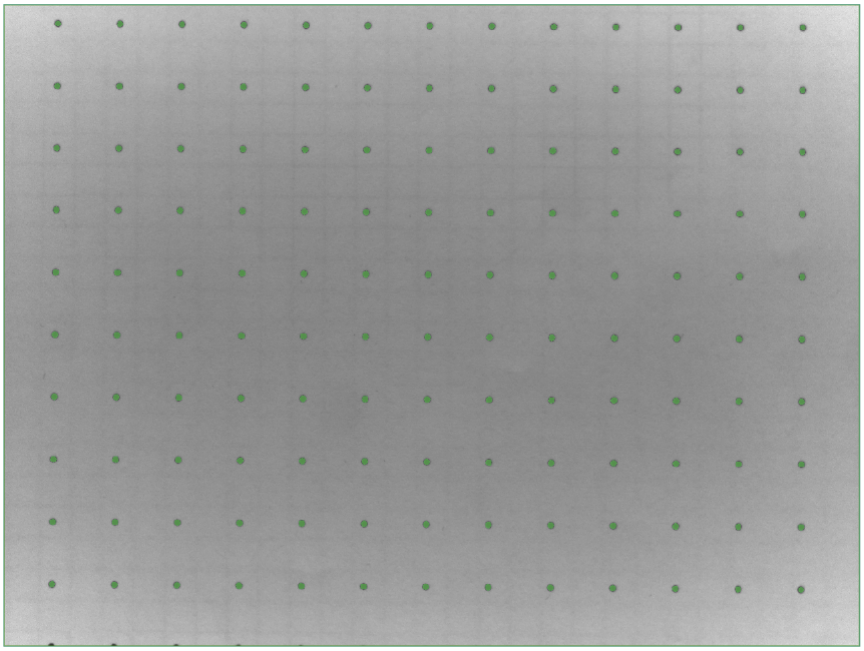


Abb. 9: Bild eines Targets für die geometrische Kalibrierung.

Der **Abstand** ist die Distanz zwischen zwei Punkten des Targets (von Mittelpunkt zu Mittelpunkt). Dieser Wert muss mit dem tatsächlichen Abstand der Punkte auf dem Target übereinstimmen, damit richtig kalibriert werden kann.

Die Region im Bild sollte dazu verwendet werden, die relevanten Punkte des Kalibrierungs-Targets einzugrenzen.

Zur korrekten Einstellung kann das **Raster** angewählt werden, um die Rasterlinien zwischen den Punkten des Targets anzuzeigen.

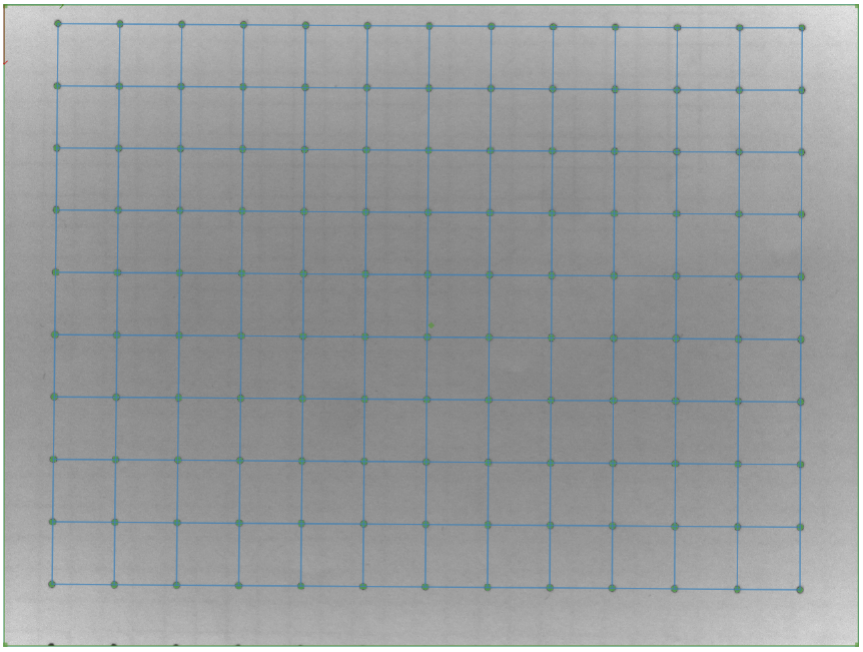


Abb. 10: Target mit Rasterlinien.

Wenn das Korrekturbild sowie die Parameter zur Belichtung und Beleuchtung korrekt eingestellt sind, kann mit der Taste **Korrigieren** die Korrektur berechnet werden.

Die **Kalibrierungsfehler** können angezeigt werden, um die Qualität der Kalibrierung zu beurteilen.

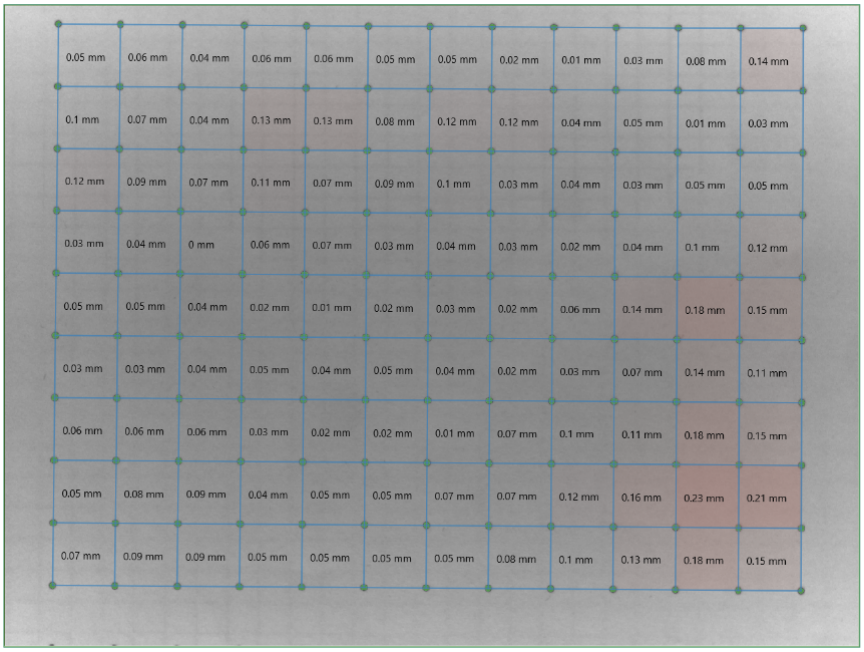


Abb. 11: Target mit Anzeige der Kalibrierungsfehler.

Nach der erfolgten Berechnung kann die Korrektur mit **Korrektur speichern** gespeichert werden, damit sie in Jobs angewandt werden kann, bzw. durch die Taste **Reset** wieder gelöscht werden.

Kalibrierungstarget

Die **Polarität** kann **Dunkel auf Hell** oder **Hell auf Dunkel** eingestellt werden, entsprechend den Gegebenheiten des verwendeten Targets.

Flächen außerhalb von **Minimale Fläche** und **Maximale Fläche** können ausgefiltert werden, um Störungen des Patterns zu vermeiden.

Visualisierung

Korrigiertes Bild zeigt ein entzerrtes Bild an.

Original Bild zeigt das ursprüngliche, nicht entzerrte Bild an.

Raster zeigt Rasterlinien für ein detektiertes Target an.

Kalibrierungsfehler zeigt Fehlerwerte der Kalibrierung an.

4.3 Werkzeuge

4.3.1 Akquisition - Bild



Abb. 12: Das Icon des Werkzeugs **Akquisition**.

Mit dem Werkzeug **Akquisition** werden die Parameter der Bildaufnahme festgelegt.

Das Werkzeug **Akquisition** ist das erste Werkzeug in einer Pipeline, es ist immer vorhanden und kann nicht gelöscht werden. In dieser Hinsicht ist das Werkzeug **Akquisition** besonders: bei jeder Inspektion ist die **Akquisition** der erste Schritt eine Verarbeitungspipeline.

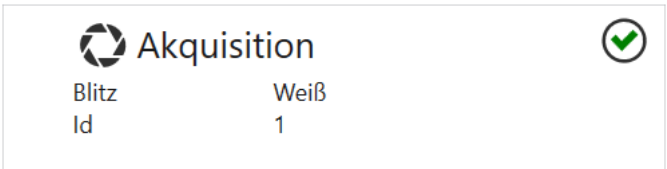


Abb. 13: Die Kachel des Werkzeugs **Akquisition**.

Das Werkzeug **Akquisition** zeigt die Blitzeinstellung sowie die Akquisitions-ID an. Wenn das Werkzeug **Akquisition** in der Pipeline ausgewählt ist, wird links eine erweiterte Benutzeroberfläche angezeigt. Auf der rechten Seite zeigt das Werkzeug das erfasste Bild an.

Parameter

Mit **Ordner** können Sie ein Verzeichnis auswählen, das Bilder enthält. Sie müssen sicherstellen, dass das Verzeichnis nur monochrome Bilder in der Bittiefe 8 Bit pro Pixel enthält, wie sie von nVision-i exportiert werden. Andernfalls können Bilder eventuell nicht geladen werden. Dieses Steuerelement wird nur angezeigt, wenn der Emulator verwendet wird.

Das Histogramm zeigt die Grauwertverteilung der erfassten Bilder an.

Belichtungszeit steuert die Belichtungsdauer eines einzelnen Bilds, beginnend bei 25 µs und bis zu 2000 µs. Bei deaktiviertem internen Blitz (unter **Blitzeinstellungen**, **Intern**) kann die **Belichtungszeit** bis zu einem Maximum von 500000 µs gewählt werden. Die Belichtungszeit beeinflusst die Helligkeit des aufgenommenen Bildes.

Verstärkung steuert die Verstärkung zwischen 1 (keine Verstärkung) und 16. Die Verstärkung wirkt linear auf die Helligkeit, d.h. ein mit der Verstärkung 2 aufgenommenes Bild hat doppelt so hohe Intensitätswerte, wie ein ohne Verstärkung aufgenommenes Bild.

Mit ROI **Anzeigen** kann eine Region positioniert wird, die den Aufnahmebereich einschränken kann. **Zurücksetzen** stellt den Aufnahmebereich auf das Vollbild zurück.

Das Bild kann wahlweise um **90**°, **180**° oder **270**° gedreht werden. Außerdem kann das Bild **Horizontal** oder **Vertikal** gespiegelt werden.

Blitzeinstellungen

Der **CS-6x/ID-6x0** Vision Sensor verfügt über eine integrierte Beleuchtung. Wenn **Intern** auf **An** eingestellt ist, beleuchten vier Hochleistungs-LED die Szene. Die Beleuchtung wird mit der Bildaufnahme synchronisiert. Wenn **Intern** auf **Aus** eingestellt ist, sind die eingebauten LEDs aus und beleuchten die Szene nicht, beispielsweise wenn externe Beleuchtung verwendet wird. Die **Farbe** kann auf **Weiß** oder **Rot** eingestellt werden, um die jeweilige LED-Farbe zu verwenden.

Zusätzlich zum internen Blitz kann unter **Extern** auch ein digitaler Ausgang ausgewählt werden um eine externe Beleuchtung anzusteuern. Zur Auswahl stehen alle Digitalen Ausgänge, welche im **Logik** Werkzeug unter **Digitale Ausgänge** nicht verwendet werden (deren Eingang **Wert** nicht verbunden sind). Umgekehrt können Digitale Ausgänge, welche als externer Blitz gewählt wurden, im **Logik** Werkzeug nicht länger verwendet werden.

Trigger

Der **Inspektionsmodus** kann auf **Standard** oder **Tor** eingestellt werden. Im **Standard**-Modus besteht ein Inspektionszyklus aus einer Bildaufnahme und der Inspektion des aufgenommenen Bilds. Im **Tor**-Modus werden kontinuierlich Bilder aufgenommen und inspiziert, solange das Tor offen ist. Das Tor wird durch digitale Trigger oder Befehle auf dem Kommandointerface geöffnet oder geschlossen. Die individuellen Zyklen werden freilaufend nacheinander ausgeführt, solange das Tor offen ist.

Der Trigger **Typ** kann im **Standard**-Modus auf **Digital**, **Frei Laufend** oder **Virtuell** eingestellt werden, im **Tor**-Modus stehen **Digital** oder **Virtuell** zur Auswahl. Wenn der Trigger auf **Digital** steht ist kein Linking der Parameter des Akquisitions-Werkzeugs möglich; entsprechende Einstellungen werden ignoriert.

Mit **Muster** kann die Aktivierung im **Tor**-Modus eingestellt werden. **Start Stop Level** öffnet das Tor mit einem Signal auf dem **Trigger** Eingang und hält es offen, solange das Signal ansteht. Wenn das Trigger-Signal nicht mehr ansteht wird das **Tor** geschlossen. Bei **Start Stop Pulse** wird das **Tor** mit dem ersten Impuls auf dem Trigger Eingang geöffnet und mit dem nächsten Impuls geschlosen. Bei **StartPulse** wird das **Tor** mit einem Impuls auf dem Trigger Eingang geöffnet und bleibt solange geöffnet, bis eine andere Bedingung es schließt: eine erfolgreiche Inspektion mit dem Ergebnis IO, ein Timeout (die Zeit kann mit **Torschluß Timeout (ms)** eingestellt werden oder eine Anzahl von Inspektionen (wird mit **Max. Anzahl Akquisitionen**) eingestellt).

Wenn der Trigger auf **Digital** eingestellt ist, wird die Bildaufnahme oder das Öffnen des Tors durch ein externes Triggersignal (am Stecker) ausgelöst. Die **Polarität** des Triggersignals kann auf **Steigende Flanke** oder **Fallende Flanke** eingestellt werden, um die entsprechende Signalflanke zu wählen. Mit der **Verzögerung** kann eine Verzögerung zwischen dem Triggersignal dem tatsächlichen Beginn der Erfassung angegeben werden.

Wenn der Trigger auf **Frei Laufend** eingestellt ist, wird der **CS-6x/ID-6x0** nach jeder Inspektion selbst ausgelöst. Wenn der Triggertyp auf **Virtuell** eingestellt ist, ist das Triggern durch Profinet, EtherNet/IP oder TCP/IP Kommandos aktiviert.

Bildkorrektur

Die Bildkorrektur beinhaltet sowohl eine Korrektur der **Schattierung** um den Helligkeitsabfall zu kompensieren, als auch eine Korrektur der **Verzerrung**, um die geometrische Verzeichnung zu kompensieren. Beide Effekte sind nicht-linear und treten zu den Bildrändern hin verstärkt auf. Die Effekte sind in ihrer Stärke vom verwendeten Objektiv und beim Shading auch von der Beleuchtung abhängig.

Die Korrekturen sind aufwändig und können aus diesem Grund optional an- oder abgewählt werden. Standardmäßig sind sie ausgeschaltet bzw. abgewählt.

Die Korrekturen sind nur anwendbar, wenn zuvor eine Kalibrierung durchgeführt und die gewünschte Kalibrierung ausgewählt wurde.

Visualisierung

Überbelichtung hervorheben zeigt überbelichtete Bildteile in gelber Farbe an.

Unterbelichtung hervorheben zeigt unterbelichtete Bildteile in blauer Farbe an.

4.3.2 Akquisition - Digitale Eingänge

Das Verknüpfen eines Werts erfolgt in zwei Schritten: Zunächst wird der Wert auf der Registerkarte **Digitale Eingänge** des Werkzeugs **Akquisition** benannt. Dann kann der benannte Wert mit der Taste als Parameter verwendet werden.

Eingabedaten werden immer dann erfasst, wenn ein Bild erfasst wird. Dies geschieht zu Beginn eines Inspektionszyklus.

Verknüpfen von Digitaleingängen

Digitale Eingänge stehen nur dann für Linking zur Verfügung, wenn Sie nicht für den Job-Wechsel benutzt werden. Die Voreinstellung kann für jeden Sensor in nVision-i getroffen werden.

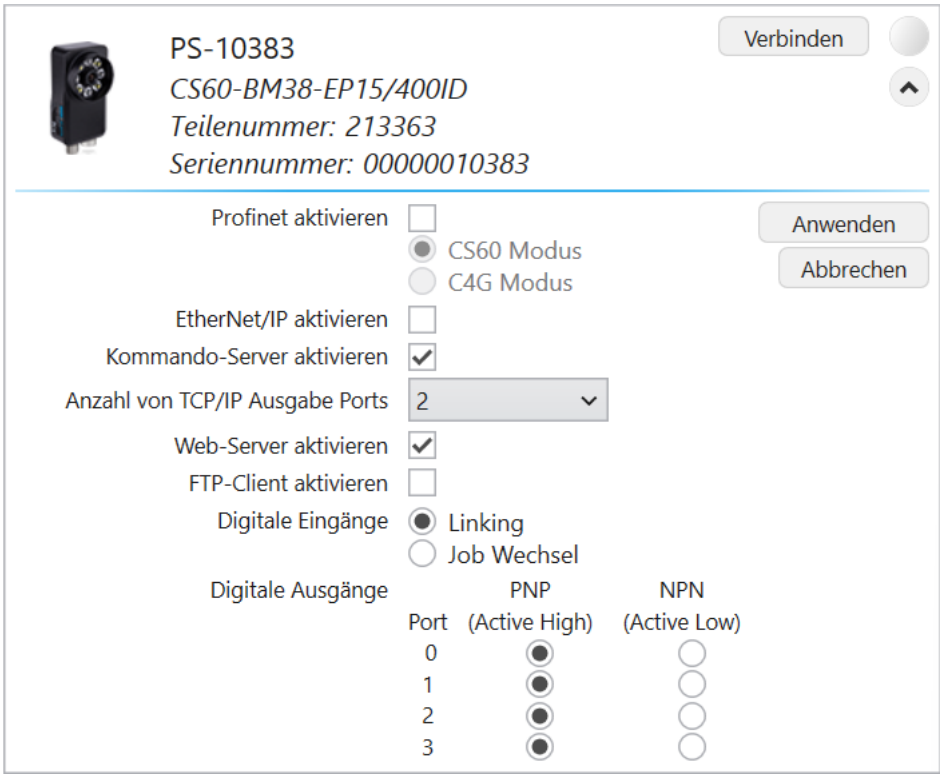


Abb. 14: **CS-6x/ID-6x0**, Voreinstellungen für digitale Eingänge.

Mit **Job Wechsel** werden die digitalen Eingänge für einen Job-Wechsel verwendet, mit **Linking** können die Werte der digitalen Eingänge eingelesen werden.

Der Vision-Sensor **CS-6x/ID-6x0** verfügt über zwei digitale Eingänge, die verknüpft werden können. Die Eingangsknoten werden auf der Seite **Digitale Eingänge** des Werkzeugs **Akquisition** angezeigt. Um sie zum Verlinken zu verwenden, müssen sie benannt werden. Ein Eingang wird benannt durch Hinzufügen eines **Link** Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden des Eingabeknoten mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

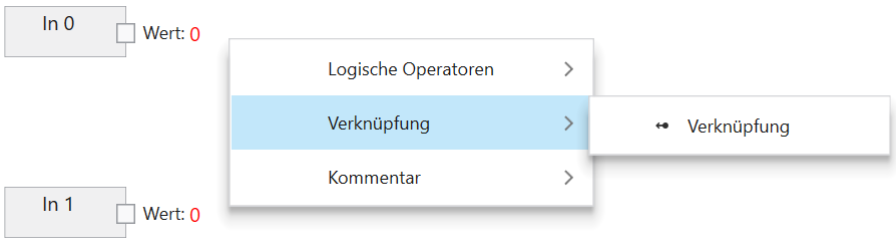


Abb. 15: Hinzufügen eines **Link** Knotens mit einen Rechts-Klick.



Abb. 16: Verbindung des Eingabeknotens mit dem **Link** Knoten.

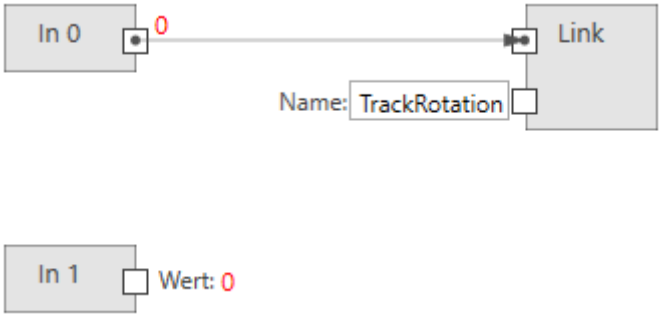


Abb. 17: Namesvergabe des **Link**

Sobald Sie Links benannt haben, können Sie diese in jedem Werkzeug (außer dem Akquisitions-Werkzeug selbst) als Boolesche Werte verwenden. Ein Beispiel Dies kann die Einstellung **Rotation verfolgen** im Werkzeug **Fläche Lokalisieren** sein.

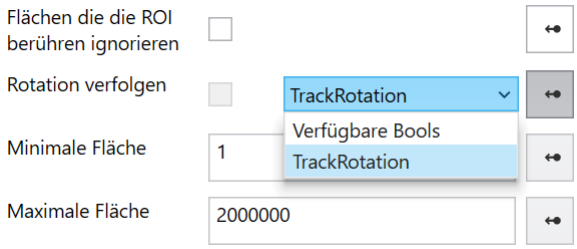


Abb. 18: Beispiel der Verwendung eines Links.

Aus technischen Gründen können digitale Eingänge nicht zum Verknüpfen von Werten in das Werkzeug **Akquisition** verwendet werden.

4.3.3 Akquisition - Feldbus Eingänge

Viele Werkzeugparameter können durch Auswahl oder Eingabe von Werten geändert werden. Die meisten dieser Parameter können auch bearbeitet werden durch **Verknüpfen** mit Feldbus-Eingängen.

Das Verknüpfen eines Werts erfolgt in mehreren Schritten auf den Registerkarten **PROFINET Eingänge** oder **EtherNet/IP Eingänge** des Werkzeugs **Akquisition**:

1. Ein PROFINET oder EtherNet/IP Modul in einem Slot wird mit einem Lese-Knoten verbunden.
2. Der Ausgang des Lese-Knotens wird mit einem Verknüpfungs-Knoten verbunden.
3. Der Verknüpfungs-Knoten definiert einen Namen, unter dem benannte Werte mit der Taste als Parameter verwendet werden.

Eingabedaten werden immer dann erfasst, wenn ein Bild erfasst wird. Dies geschieht zu Beginn eines Inspektionszyklus.

Verknüpfen von Feldbus Eingängen

Hinweis: Die Screenshots zeigen PROFINET, aber die Verfahren für EtherNet/IP sind genau dieselben, es sei denn, es ist ausdrücklich vermerkt.

Der **SPS** Knoten wird verwendet, um Module in bestimmten Steckplätzen zu definieren. Dieses Setup muss exakt mit dem PROFINET-Setup oder dem EtherNet/IP-Setup auf der SPS übereinstimmen, um die Kommunikation zwischen der SPS und dem **CS-6x/ID-6x0** zu ermöglichen. Der SPS Knoten wird sowohl im Akquisition als auch im Logik Werkzeug angezeigt. Innerhalb des SPS Knotens werden die Namen Eingang und Ausgang für die Module wie von der SPS aus gesehen verwendet (Ein SPS Ausgang wird von der SPS über PROFINET bzw. EtherNet/IP zum **CS-6x/ID-6x0** übertragen, wo er als Eingang angesehen wird. Ein SPS-Eingang wird vom **CS-6x/ID-6x0** über PROFINET bzw. EtherNet/IP zur SPS übertragen, auf dem **CS-6x/ID-6x0** wird dies als Ausgang gesehen.)

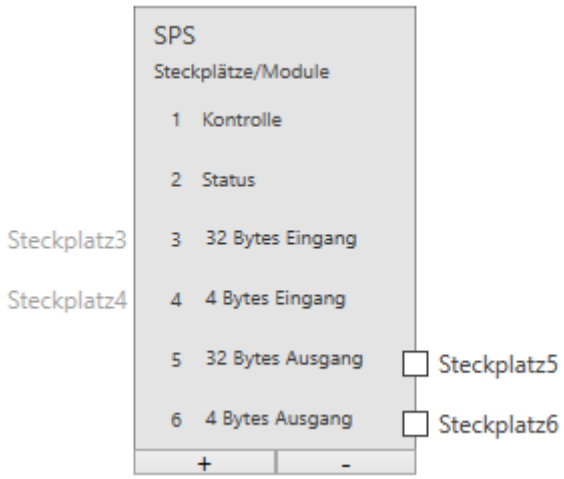


Abb. 19: Der SPS Knoten mit sechs Modulen.

Beiespielhaft sind hier sechs Module ausgewählt (die Module Kontrolle und Status sind nicht abwählbar). Auf der Seite **PROFINET Eingänge** können nur die Module 5 und 6 verbunden werden (aus Sicht der SPS sind das Ausgänge, aus Sicht des **CS-6x/ID-6x0** sind das Eingänge.)

Unter der Überschrift **Steckplätze** werden die Module grafisch angezeigt. Außerdem werden die verwendeten Regionen innerhalb der Module angezeigt.

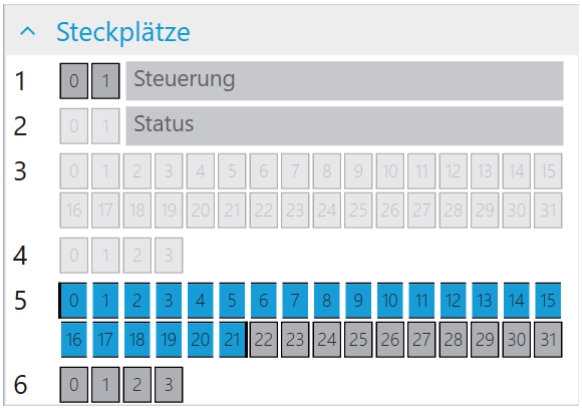


Abb. 20: Ein String mit 20 Zeichen im Modul 5 an Adresse 0.

PROFINET oder EtherNet/IP kann Eingabewerte verschiedener Typen bereitstellen: Logisch (Bool), Numerisch (Byte, Int32, Float) und Text (String). Um sie zum Verlinken zu verwenden, müssen sie definiert und benannt werden. Ein Eingang wird definiert, indem ein Leseknoten des gewünschten Typs hinzugefügt wird (mit einem Rechtsklick auf Canvas und Auswahl von **Read Boolean**, **Read Byte**, **Read Int32**, **Read Float** oder **Read String**) und mit einem bestimmten Modul in einem Steckplatz verbunden wird, der im SPS Knoten definiert ist. Ein Eingang wird benannt durch Hinzufügen eines **Verknüpfung** Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden eines Leseknotens mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

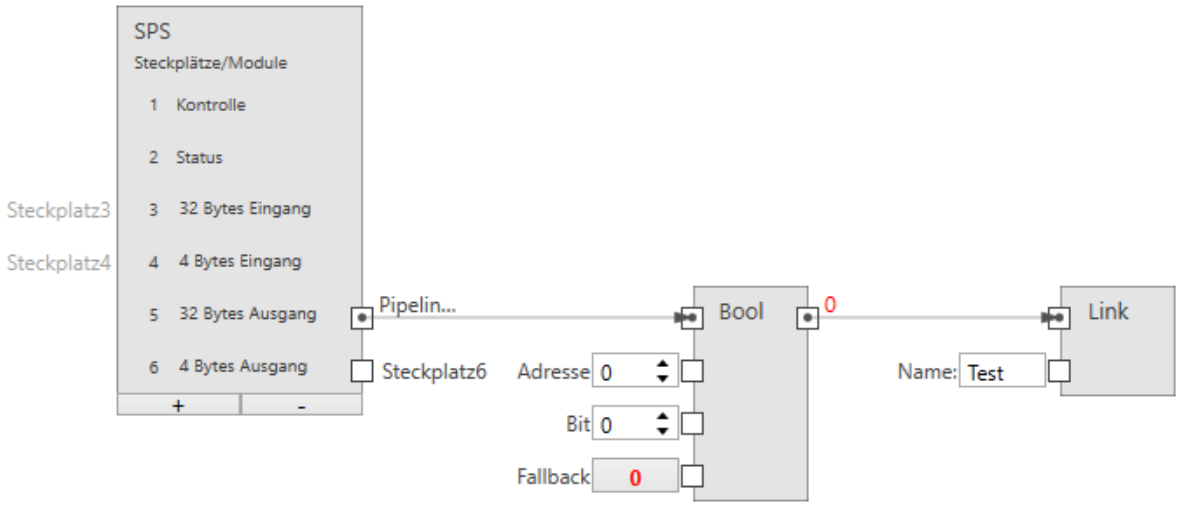


Abb. 21: Ein boolescher Eingang.

Die Größe eines booleschen Eingangs beträgt 1 Byte (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Bit gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Fallback gibt den Booleschen Wert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

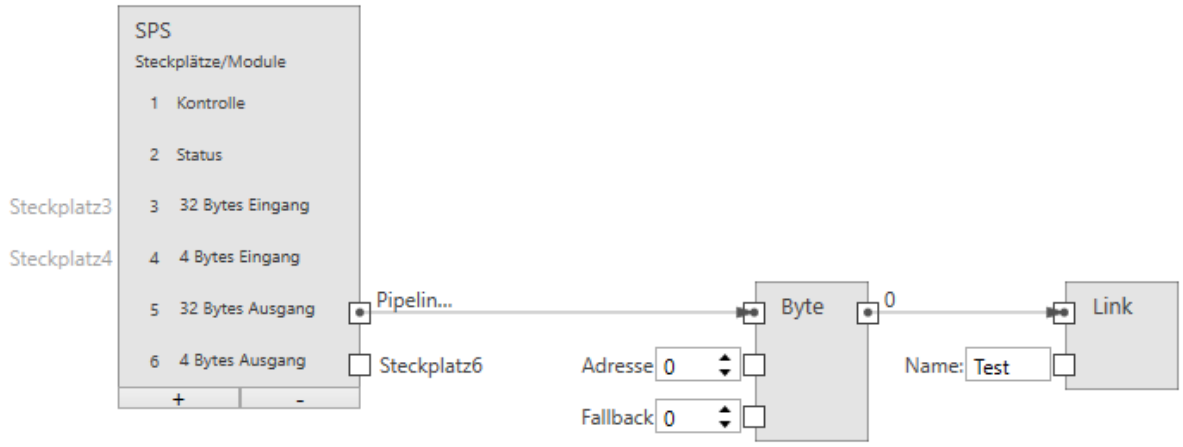


Abb. 22: Ein Byte Eingang.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Fallback gibt den Byte Wert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

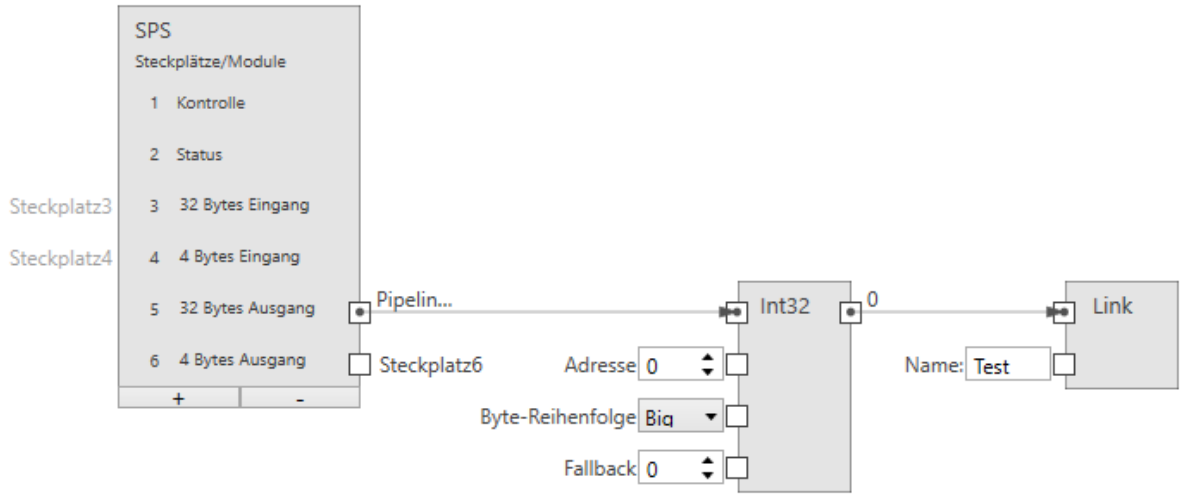


Abb. 23: Ein ganzzahliger Eingang.

Die Größe eines ganzzahligen Eingangs beträgt 4 Byte.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback gibt den Zahlenwert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

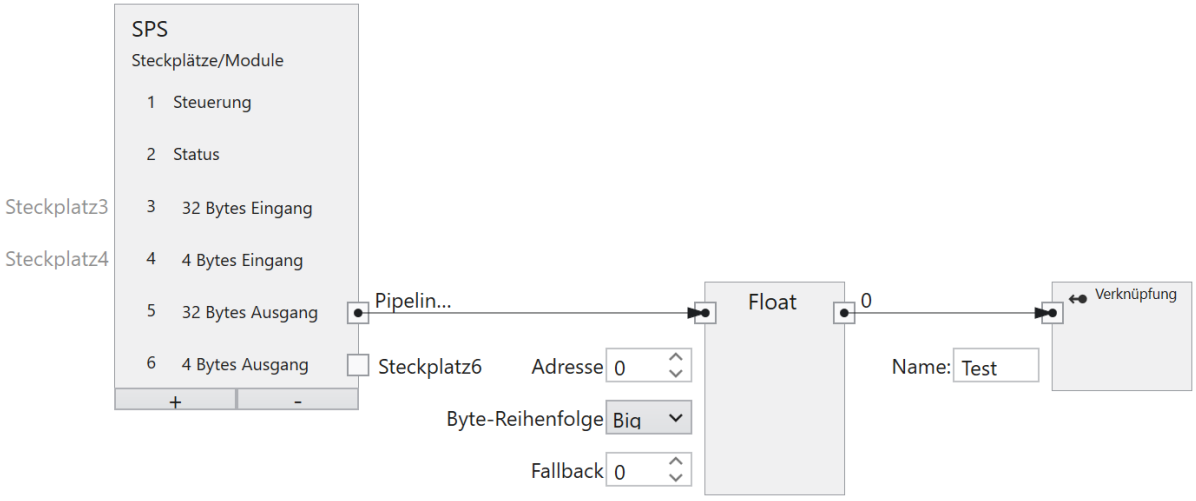


Abb. 24: Ein Fließkomma Eingang.

Die Größe eines Fließkomma Eingangs beträgt 4 Byte.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback gibt den Fließkommawert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

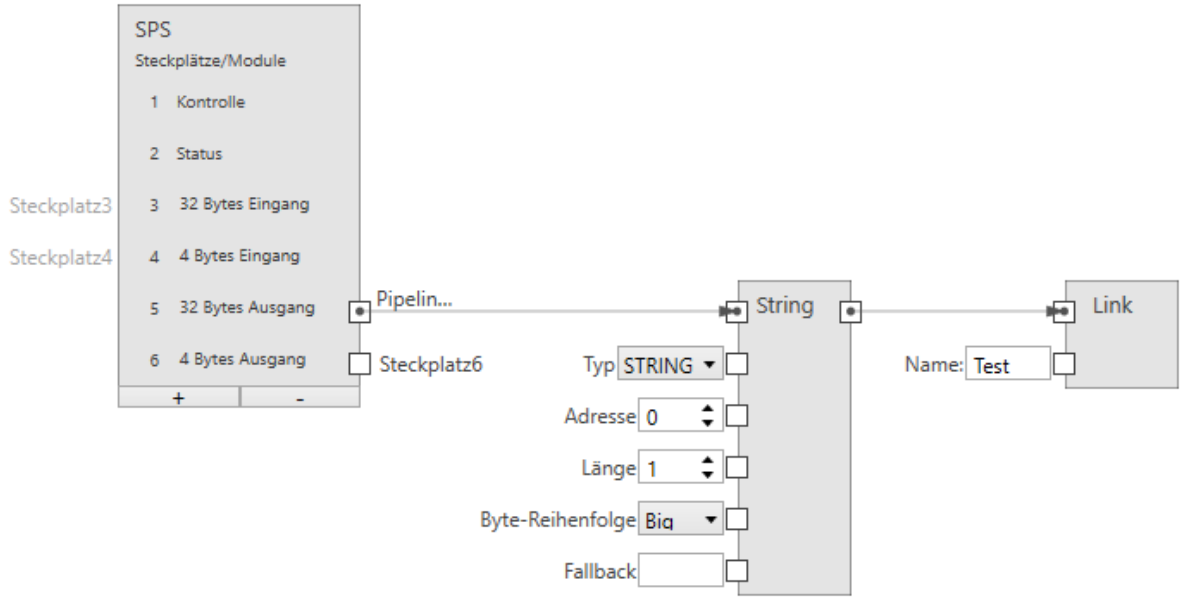


Abb. 25: Ein Text Eingang.

Profinet: Die Größe eines Texts beträgt (2 + n) * s, wobei n die maximale Anzahl von Zeichen ist und s = 1 (für einen Ansi STRING) oder s = 2 (für einen Unicode-WSTRING). Das erste Byte/Wort (Ansi/Unicode) enthält die maximale Anzahl von Zeichen, welche die Zeichenfolge enthalten kann. Das zweite Byte/Wort (Ansi/Unicode) enthält die tatsächliche Anzahl der Zeichen, welche die Zeichenfolge enthält.

EtherNet/IP: Die Größe eines Texts beträgt (4 + n), wobei n die maximale Anzahl von Zeichen ist. Die ersten 4 Byte werden als int32 interpretiert und stellt die tatsächliche Anzahl der Zeichen des strings da.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Typ gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist. Bei EtherNet/IP entfällt dieser Input, da EtherNet/IP nur Ansi Texte unterstützt.

Adresse gibt die Adresse im Modul an.

Länge gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Byte-Reihenfolge Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden. Bei EtherNet/IP entfällt dieser Input, da er nur für Unicode Texte relevant ist.

Fallback gibt den String Wert an, der verwendet wird, wenn die Feldbus-Daten nicht verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET bzw. EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

4.3.4 Akquisition - TCP/IP Kommando Eingänge

Viele Werkzeugparameter können durch Auswahl oder Eingabe von Werten geändert werden. Die meisten dieser Parameter können auch bearbeitet werden durch **Verknüpfen** mit TCP/IP Kommando Eingängen.

Das Verknüpfen eines Werts erfolgt in mehreren Schritten auf der Registerkarte **TCP/IP Variablen Eingänge** des Werkzeugs **Akquisition**:

1. Eine TCP/IP Variable wird mit einem Namen gekennzeichnet und mit einem Lese-Knoten verbunden.
2. Der Ausgang des Lese-Knotens wird mit einem Verknüpfungs-Knoten verbunden.
3. Der Verknüpfungs-Knoten definiert einen Namen, unter dem benannte Wert mit der Taste als Parameter verwendet werden.

Die Namen unter Punkt 1 und 3 können unterschiedlich sein, bezeichnen aber letztlich das selbe Objekt. Unter Punkt 1 wird der Name definiert, mit dem der Wert vom TCP/IP Kommando `set name=value` identifiziert wird. Unter Punkt 3 wird de Name definiert, unter dem der Wert mit den Links im Job identifiziert wird.

Eingabedaten werden immer dann erfasst, wenn ein Bild erfasst wird. Dies geschieht zu Beginn eines Inspektionszyklus.

Verknüpfen von TCP/IP Kommando Eingängen

Der **Inputs** Knoten wird verwendet, um Variablen im TCP/IP Namensraum zu benennen.

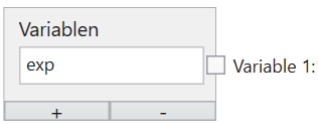


Abb. 26: Der Inputs Knoten mit einer Variablen.

Beispielhaft ist hier eine Variable definiert. Der Name muss mit einem Buchstaben beginnen und kann optional gefolgt sein von einem oder mehreren Buchstaben, Ziffern oder _ (Unterstrich). Dieser Name ist gültig im TCP/IP Namensraum, d.h. bei Verwendung mit den TCP/IP Kommando `set name=value`. Beispielsweise würde das Kommando `set exp=100` den Wert `100` der Variablen `exp` zuweisen.

TCP/IP Kommando Variablen können Eingabewerte verschiedener Typen bereitstellen: Logisch (Bool), Numerisch (Byte, Int32, Float) und Text (String). Um sie zum Verlinken zu verwenden, müssen sie definiert und benannt werden. Ein Eingang wird definiert, indem ein Leseknoten des gewünschten Typs hinzugefügt wird (mit einem Rechtsklick auf Canvas und Auswahl von **Read Boolean**, **Read Byte**, **Read Int32**, **Read Float** oder **Read String**) und mit einem bestimmten Modul in einem Steckplatz verbunden wird, der im Inputs Knoten definiert ist. Ein Eingang wird benannt durch Hinzufügen eines **Verknüpfung** Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden eines Leseknotens mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

Bool Eingang

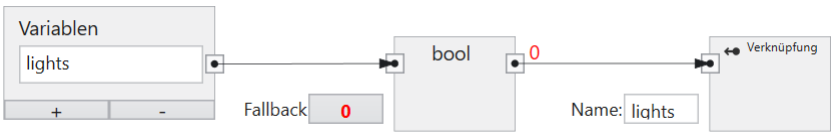


Abb. 27: Ein Bool Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den bool'schen Wert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Byte Eingang

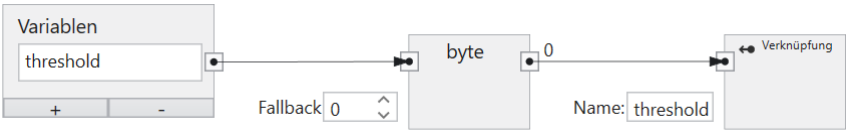


Abb. 28: Ein Byte Eingang.

Steckplatz verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Fallback gibt den Byte Wert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Ganzzahliger Eingang

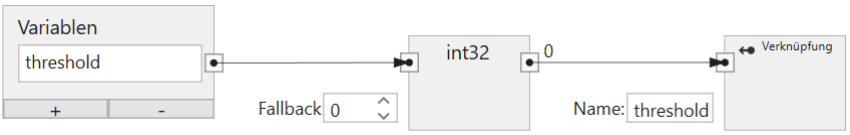


Abb. 29: Ein ganzzahliger Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den Zahlenwert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Fließkomma Eingang

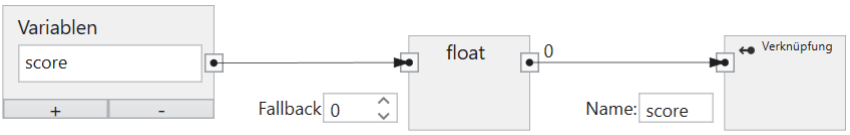


Abb. 30: Ein Fließkomma Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den Fließkommawert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

Text Eingang

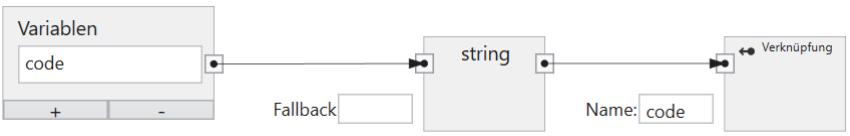


Abb. 31: Ein Text Eingang.

Variable verbindet den Knoten mit einer Variablen im Inputs Knoten.

Fallback gibt den String Wert an, der verwendet wird, wenn die TCP/IP Kommando Daten nicht verfügbar sind.

Name gibt den symbolischen Namen an, der verwendet wird, wenn dieser Wert mit einer Werkzeugeingabe verknüpft wird.

4.3.5 Flächenpixel Erkennen



Abb. 32: Das Icon des Werkzeugs **Flächenpixel Erkennen**.

Das Werkzeug **Flächenpixel Erkennen** wird verwendet, um die Anzahl der Pixel eines Teils in einer Szene mithilfe von Blob-Analyse zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Flächenpixel Erkennen**-Werkzeugs die **Anzahl** von Pixeln.



Abb. 33: Die Kachel des Werkzeugs **Flächenpixel Erkennen**.

Das Werkzeug **Flächenpixel Erkennen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken).

Ein **Rechteck** kann auch gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Beim **Polygon** können Ecken durch Klicken auf eine Randlinie hinzugefügt werden, bzw. durch Drücken der Delete Taste entfernt werden.

Einer ROI kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden.

Der **Schwellenwert** wird verwendet, um **Dunkle Flächen**, **Helle Flächen**, **Manuell** oder **Adaptiv** auszuwählen. Was dunkel oder hell ist, wird über einen Schwellenwert bestimmt, der automatisch berechnet wird, oder manuell mit einem Schieberegler eingestellt wird, unter Verwendung der logarithmisch skalierten Histogramm-Anzeige des Bildes. Der adaptive Modus arbeitet mit einem prozentualen **Schwellenwert** innerhalb einer **Fenstergröße**, optional kann die Einstellung **Löcher füllen** gewählt werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur

Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Filterung

Glättung unterzieht die Bilddaten einer einstellbaren Bildglättung.

Flächen die die ROI berühren ignorieren ignoriert alle Objekte, welche den Rand der interaktiv auf der rechten Seite eingestellten ROI berühren.

Im Modus **Fläche filtern** können **Minimale Fläche** und **Maximale Fläche** verwendet werden, um einen Größenbereich in Quadrat-Pixeln anzugeben, welcher die Flächen eingrenzt.

Im Modus **Größe filtern** können **Minimale Breite**, **Maximale Breite**, **Minimale Höhe** und **Maximale Höhe** verwendet werden um einen Größenbereich anzugeben, welcher die Flächen eingrenzt.

Im Modus **Rundheit filtern** kann die **Rundheit** eingegrenzt werden, um Flächen nach ihrer Form einzugrenzen.

Count

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI werden die Pixel aller Flächen gezählt und deren **Anzahl** angezeigt.



Abb. 34: Das Bild-GUI des Werkzeugs **Flächenpixel Erkennen**.

4.3.6 Flächen Zählen



Abb. 35: Das Icon des Werkzeugs **Flächen Zählen**.

Das Werkzeug **Flächen Zählen** wird verwendet, um die Anzahl der Teile in einer Szene mithilfe von Blob-Analyse zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Flächen Zählen**-Werkzeugs die **Anzahl** von Objekten.

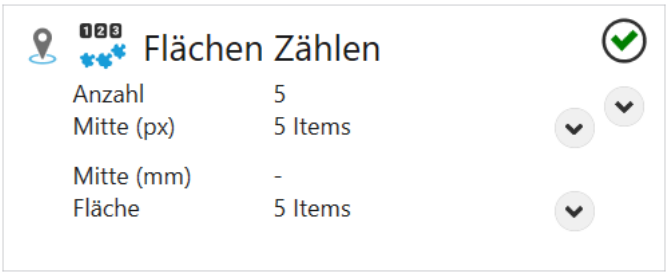


Abb. 36: Die Kachel des Werkzeugs **Flächen Zählen**.

Das Werkzeug **Flächen Zählen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken).

Ein **Rechteck** kann auch gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Beim **Polygon** können Ecken durch Klicken auf eine Randlinie hinzugefügt werden, bzw. durch Drücken der Delete Taste entfernt werden.

Einer ROI kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden.

Der **Schwellenwert** wird verwendet, um **Dunkle Flächen**, **Helle Flächen**, **Manuell** oder **Adaptiv** auszuwählen. Was dunkel oder hell ist, wird über einen Schwellenwert bestimmt, der automatisch berechnet wird, oder manuell mit einem Schieberegler eingestellt wird, unter Verwendung der logarithmisch skalierten Histogramm-Anzeige des Bildes. Der adaptive Modus arbeitet mit einem prozentualen **Schwellenwert** innerhalb einer **Fenstergröße**, optional kann die Einstellung **Löcher füllen** gewählt werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Sortierung beeinflusst die Ordnung der Resultate in Bezug auf ihren Mittelpunkt.

Rotation verfolgen kann verwendet werden, um die Drehung eines Teils zusätzlich zu seiner Position zu verfolgen. Verfolgung der Rotation ist nur mit Teilen möglich, die eine inhärente Vorzugsrichtung haben (z. B. ein kreisförmiges Teil hat unendlich viele Richtungen, damit ist seine Richtung bedeutungslos. Ein Quadrat hat vier mögliche Richtungen, was auch bedeutet, dass eine bevorzugte Richtung nicht berechnet werden kann.)

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Filterung

Glättung unterzieht die Bilddaten einer einstellbaren Bildglättung.

Flächen die die ROI berühren ignorieren ignoriert alle Objekte, welche den Rand der interaktiv auf der rechten Seite eingestellten ROI berühren.

Im Modus **Fläche filtern** können **Minimale Fläche** und **Maximale Fläche** verwendet werden, um einen Größenbereich in Quadrat-Pixeln anzugeben, welcher die Flächen eingrenzt.

Im Modus **Größe filtern** können **Minimale Breite**, **Maximale Breite**, **Minimale Höhe** und **Maximale Höhe** verwendet werden um einen Größenbereich anzugeben, welcher die Flächen eingrenzt.

Im Modus **Rundheit filtern** kann die **Rundheit** eingegrenzt werden, um Flächen nach ihrer Form einzugrenzen.

Anzahl

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Anzahl der getrennten Flächen gezählt und mit farbigen grafischen Überlagerungen angezeigt. Zusätzlich wird die **Anzahl** angezeigt.

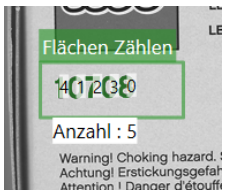


Abb. 37: Das Bild-GUI des Werkzeugs **Flächen Zählen**.

4.3.7 Zeichen Zählen



Abb. 38: Das Icon des Werkzeugs **Zeichen Zählen**.

Das Werkzeug **Zeichen Zählen** wird verwendet, um die Anzahl der Textzeichen in einer Szene zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Zeichen Zählen**-Werkzeugs die **Anzahl** von Zeichen.



Abb. 39: Die Kachel des Werkzeugs **Zeichen Zählen**.

Das Werkzeug **Zeichen Zählen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Schriftart

Die Schriftart kann aus einer Liste gewählt werden.

Es gibt mehrere Gruppen von Schriften. Die Mitglieder der einzelnen Gruppen unterscheiden sich, da sie unterschiedliche Symbolsätze enthalten. Der Inhalt einer Schriftart wird durch ihren Namen beschrieben. Für die Namen der vortrainierten Schriften gilt die folgende Nomenklatur:

Der Name beginnt mit dem Gruppennamen, z. B. **Document** oder **DotPrint**, gefolgt von Indikatoren für den Symbolsatz die in der OCR-Schriftart enthalten sind. Die Bedeutung der Indikatoren ist die folgende:

- **0-9**: Die OCR-Schriftart enthält die Ziffern 0 bis 9.

- **A-Z**: Die OCR-Schriftart enthält die Großbuchstaben A bis Z.
- **+** : Die OCR-Schriftart enthält Sonderzeichen. Die Liste der Sonderzeichen variiert leicht zwischen den einzelnen OCR-Schriften. Sie ist im Folgenden für jede OCR-Schriftart einzeln aufgeführt.

Wenn der Name der OCR-Schriftart keines der oben genannten Kennzeichen enthält, enthält die OCR-Schriftart normalerweise die Ziffern 0 bis 9, die Großbuchstaben A bis Z, die Kleinbuchstaben Kleinbuchstaben a bis z und Sonderzeichen. Einige OCR-Schriften enthalten keine Kleinbuchstaben (z. B., DotPrint). Dies wird in der Beschreibung der jeweiligen Schriftart ausdrücklich erwähnt.

Um den Zeichensatz durch Verknüpfung einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Zeichensatz, wie er unter **Schriftart** zu finden ist, entsprechen.

Document

Die Schriftart `Dokument` kann verwendet werden, um Zeichen zu lesen, die in Schriftarten wie Arial, Courier oder Times New Roman gedruckt wurden. Dies sind typische Schriftarten für den Druck von Dokumenten oder Briefen.

Beachten Sie, dass die Zeichen I und l der Schriftart Arial nicht unterschieden werden können. Das bedeutet, dass ein `l` fälschlicherweise für ein `I` gehalten werden kann und umgekehrt.

Verfügbare Sonderzeichen: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

Die OCR-Schriftart DotPrint kann verwendet werden, um mit Punktdruckern gedruckte Zeichen zu lesen. Sie enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . * :

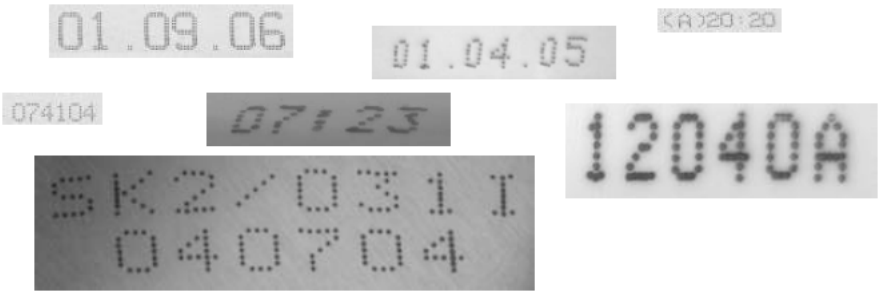


Abb. 40: Beispiele für gepunktete Schrift.

HandWritten

Die Schriftart HandWritten_0-9 kann zum Lesen handgeschriebener Zahlen verwendet werden.

Sie enthält die Ziffern 0-9.

Verfügbare Sonderzeichen: keine

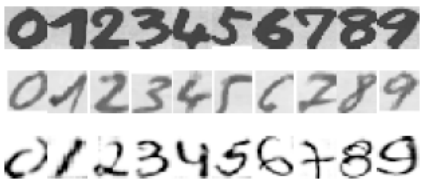


Abb. 41: Beispiele für handgeschriebene Ziffern.

Industrial

Die Schriftart Industrial kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B oder anderen serifenlosen Schriften Schriftarten gedruckt wurden. Diese Schriftarten werden normalerweise zum Drucken von z. B. Etiketten verwendet.

Verfügbare Sonderzeichen: - / + . $ % * e £ ¥

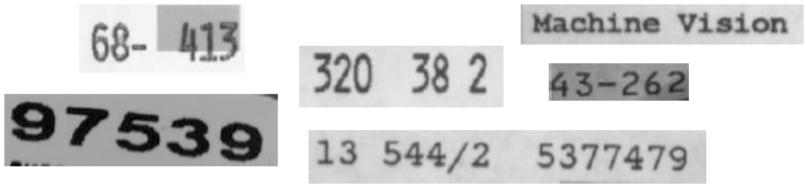


Abb. 42: Beispiele für industrielle Drucke.

OCR-A

Die Schrift OCR-A kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-A gedruckt sind.

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



Abb. 43: Ausgewählte Zeichen der OCR-A Schrift.

OCR-B

Die Schrift OCR-B kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-B gedruckt sind.

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



Abb. 44: Ausgewählte Zeichen der OCR-B Schrift.

Pharma

Die Schriftart Pharma kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B und anderen Schriftarten gedruckt sind, die typischerweise in der pharmazeutischen Industrie verwendet werden.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . ( ) :

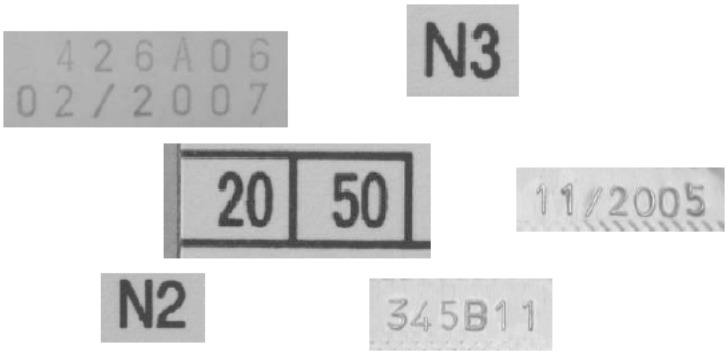


Abb. 45: Beispiele für pharmazeutische Drucke.

SEMI

Die Schriftart SEMI kann verwendet werden, um Zeichen zu lesen, die in der Schriftart SEMI gedruckt sind, die aus Zeichen besteht, die die so gestaltet sind, dass sie sich leicht voneinander unterscheiden lassen. Sie hat einen begrenzten Satz von Zeichen.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - .



Abb. 46: Zeichensatz der SEMI Schriftart.

Universal

Die OCR-Schriftart Universal kann zum Lesen einer Vielzahl unterschiedlicher Zeichen verwendet werden. Das Training dieser CNN-basierten Schriftart basiert auf den Zeichensätzen 'Document', 'DotPrint', 'SEMI' und 'Industrial'.

Verfügbare Sonderzeichen: - / = + : < > . # $ % & ( ) @ * e £ ¥

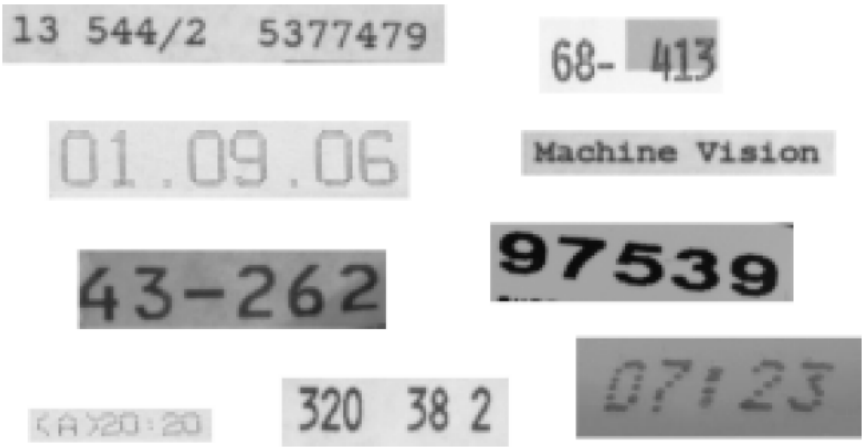


Abb. 47: Beispiele für den Zeichensatz Universal.

Segmentierung

Min. Kontrast bestimmt den minimalen Kontrast zwischen Buchstaben und ihrem Hintergrund. Der Wert kann zwischen 1 und 255 eingestellt werden, der Standardwert ist 15.

Die **Polarität** kann auf **Dunkel auf Hell** für dunkle Zeichen auf hellem Hintergrund, **Hell auf Dunkel** für helle Zeichen auf dunklem Hintergrund oder auf **Beide** eingestellt werden, wenn beide Arten von Text segmentiert werden sollen.

Wenn **Rand eliminieren** angewählt ist, werden Regionen verworfen, die den Rand der ROI berühren.

Wenn **Fragmente hinzufügen** angewählt ist, werden Fragmente wie der Punkt auf einem "i" zu den segmentierten Buchstaben hinzugefügt. Das kann allerdings auch dazu führen, dass Störungen zu den segmentierten Buchstaben hinzugefügt werden.

Berührende Zeichen bestimmt die Trennung von Paaren oder kleinen Gruppen nebeneinander liegender Zeichen, die als eine einzige Region segmentiert werden. Wird **Standard** gewählt, wer-

den solche Regionen erkannt und in zwei oder mehrere einzelne Zeichen aufgeteilt. Wird der Werte auf **falsch** gesetzt, wird keine Trennung von sich berührenden Zeichen durchgeführt.

Zeichengröße

Minimale Zeichenhöhe bestimmt die minimale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Maximale Zeichenhöhe bestimmt die maximale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Minimale Zeichenbreite bestimmt die minimale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Maximale Zeichenbreite bestimmt die maximale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Minimale Strichstärke bestimmt die minimale Strichstärke eines Buchstabens in Pixel. Soll die minimale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Minimale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Maximale Strichstärke bestimmt die maximale Strichstärke eines Buchstabens in Pixel. Soll die maximale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Maximale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Sonderzeichen

Wenn **Satzzeichen** angewählt ist, werden kleine Satzzeichen, die nahe an der Grundlinie der Textzeile liegen (wie Punkt oder Komma), zu den segmentierten Buchstaben hinzugefügt.

Wenn **Trennzeichen** angewählt ist, werden Trennzeichen wie ein Minus oder das Gleichheitszeichen zu den segmentierten Buchstaben hinzugefügt.

Punktdruck

Punktdruck muss angewählt werden, falls der zu segmentierende Text Punktdruck-Buchstaben enthält.

Enge Zeichen muss angewählt werden, wenn die Lücken zwischen benachbarten Buchstaben kleiner sind als die Lücke zwischen zwei Punkten innerhalb eines Buchstabens. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die genaue Größe der kleinsten Lücke zwischen benachbarten Buchstaben bekannt, kann diese stattdessen mittels **Minimale Zeichenlücke** gesetzt werden. In diesem Fall wird die Einstellung von **Enge Zeichen** ignoriert.

Minimale Zeichenlücke ist die Größe der kleinsten Lücke zwischen zwei Buchstaben in Pixel. Dieser Parameter kann in Fällen, in denen die Lücken zwischen Buchstaben kleiner sind als die zwischen Punkten innerhalb der Buchstaben, zu einer genaueren Segmentierung führen. Falls die minimale Größe nicht bekannt oder größer als die maximale Lücke zwischen den Punkten innerhalb eines Buchstaben ist, kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Sind Buchstaben sehr nah nebeneinander gedruckt, jedoch die Größe der minimalen Lücke unbekannt, kann stattdessen **Enge Zeichen** gesetzt werden.

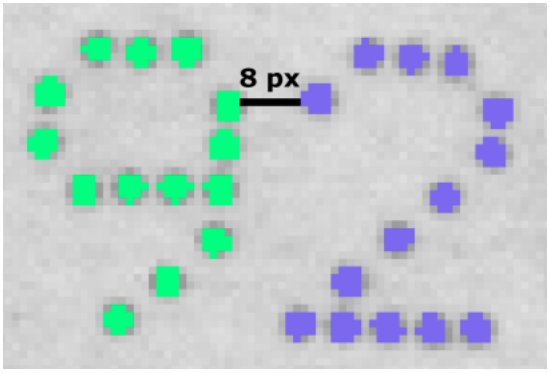


Abb. 48: Hier ist die kleinste Lücke zwischen zwei Buchstaben 8 Pixel.

Maximale Punktelücke ist die Größe der größten Lücke zwischen zwei Punkten innerhalb eines Buchstabens in Pixel. Um beliebige Punktdruck-Buchstaben zu segmentieren kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die größte Lücke zwischen zwei Punkten größer als die kleinste Lücke zwischen zwei Buchstaben, sollte zusätzlich **Enge Zeichen** angewählt werden oder **Minimale Zeichenlücke** entsprechend gesetzt werden. Wird **Maximale Punktelücke**gesetzt, kann die Performance der Texterkennung deutlich gesteigert werden.

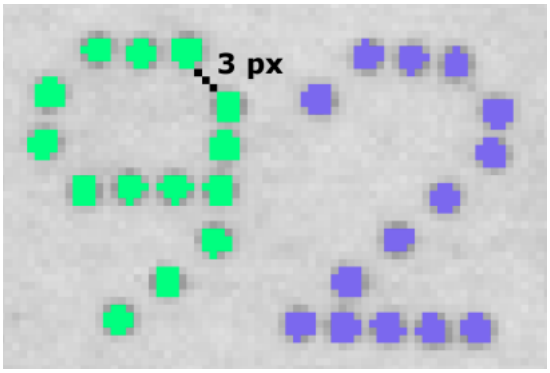


Abb. 49: Hier ist die größte Lücke zwischen zwei Punkten 3 Pixel.

Struktur

Zeilenstruktur legt die Struktur des gesuchten Textes fest. Für jede Zeile wird der Abstand zwischen aufeinanderfolgenden Buchstaben berechnet. Ausgehend von diesen Abständen wird die Zeile in einzelne Buchstabenblöcke aufgeteilt. Hierbei werden sehr kleine Zeichen wie z.B. ., _ und – vernachlässigt und als Leerzeichen interpretiert. Darüber hinaus können zusätzliche Trennzeichen angegeben werden, die ebenfalls vernachlässigt werden sollen, siehe **Zeilentrennzeichen**.

Für die entstandenen Buchstabenblöcke wird dann geprüft ob sie einer der übergebenen Strukturen entsprechen.

Eine Struktur besteht aus einem String, der die Anzahl der Buchstaben/Ziffern pro Buchstabenblock enthält, welche durch Leerzeichen getrennt sind. Soll zum Beispiel ein Datum mit jeweils zwei Ziffern für Tag, Monat und Jahr gefunden werden, wäre die Struktur `2 2 2`. Kann das Jahr auch aus vier Ziffern bestehen, wäre die Struktur `2 2 2-4`, so dass der letzte Block aus zwei bis vier Buchstaben/Ziffern bestehen kann. Wenn die Struktur einem leeren Text entspricht, kann der zu findende Texte aus jeder beliebigen Struktur bestehen.

Jede gefundene Struktur wird als einzelne Zeile gespeichert. Soll stattdessen die komplette Zeile zurückgeliefert werden, die eine Struktur enthält, kann **Ganze Zeile** angewählt werden.

Zeilentrennzeichen ist ein String von Trennzeichen, die bei der Suche nach Strukturen vernachlässigt werden sollen, siehe **Zeilenstruktur**. Hierbei ist zu beachten, dass ein solches Trennzeichen immer auch im Klassifikator enthalten sein muss. Sollen z.B. die Zeichen `:` und `\` als Trennzeichen behandelt werden, kann der String `:\\` übergeben werden. Hierbei ist zu beachten, dass das vorangestellte `\` die Verwendung des Symbols `\` als Literal ermöglicht.

Ganze Zeile falls nur die gefundenen Strukturen als Zeilen zurückgegeben werden sollen, `false` falls jede Zeile, die eine passende Stuktur enthält komplett zurückgegeben werden soll.

Parameters

Sortierung beeinflusst die Ordnung der Resultate in Bezug auf ihren Mittelpunkt.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Anzahl

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Anzahl der Zeichen gezählt und mit farbigen grafischen Überlagerungen angezeigt. Zusätzlich wird die **Anzahl** angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).



Abb. 50: Das Bild-GUI des Werkzeugs **Zeichen Zählen**.

4.3.8 Kantenpixel Erkennen



Abb. 51: Das Icon des Werkzeugs **Kantenpixel Erkennen**.

Das Werkzeug **Kantenpixel Erkennen** wird verwendet, um die Anzahl der Kantenpixel eines Teils in einer Szene mithilfe der Blob-Analyse zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Kantenpixel Erkennen**-Werkzeugs die **Anzahl** von Katenpixeln.



Abb. 52: Die Kachel des Werkzeugs **Kantenpixel Erkennen**.

Das Werkzeug **Kantenpixel Erkennen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken).

Ein **Rechteck** kann auch gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Beim **Polygon** können Ecken durch Klicken auf eine Randlinie hinzugefügt werden, bzw. durch Drücken der Delete Taste entfernt werden.

Einer ROI kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden.

Kantenpixel werden über einen **Schwellenwert** bestimmt und ausgewählt, der **Automatisch** berechnet wird, oder manuell mit einem Schieberegler eingestellt wird, unter Verwendung der logarithmisch skalierten Histogramm-Anzeige des Bildes.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Anzahl

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI werden Kanten berechnet und die Anzahl der Kantenpixel über dem Schwellenwert bestimmt und angezeigt.



Abb. 53: Das Bild-GUI des Werkzeugs **Kantenpixel Erkennen**.

4.3.9 Kanten Zählen



Abb. 54: Das Icon des Werkzeugs **Kanten Zählen**.

Das Werkzeug **Kanten Zählen** wird verwendet, um die Anzahl der Kanten entlang einer Linie zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Kanten Zählen**-Werkzeugs die **Anzahl** von Kanten.



Abb. 55: Die Kachel des Werkzeugs **Kanten Zählen**.

Das Werkzeug **Kanten Zählen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Polarität** wird verwendet, um **Dunkel -> Hell**, **Hell -> Dunkel** oder **Beide** auszuwählen.

Glättung beeinflusst den Grad der Glättung, der auf die **Kantenstärke** angewendet wird, zwischen **Scharf** und **Unscharf**.

Die zu zählenden Kanten werden mit **Kantenstärke** bestimmt, die durch Überprüfen des Verlaufs entlang der Linie visuell bestimmt werden kann.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Anzahl

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer LOI (Line of Interest). Entlang der Linie wird die Kantenstärke berechnet , die festlegt ob die Kanten gezählt werden. Zusätzlich wird die **Anzahl** angezeigt.

Die LOI (Line of Interest) kann verschoben werden (durch Ziehen der Linie oder der Endpunkte oder mit den Pfeiltasten der Tastatur). Wenn der Mauszeiger über den Endpunkten ist, wird ein Vier-Wege-Cursor angezeigt und der Endpunkt kann in alle Richtungen verschoben werden. Wenn der Mauszeiger von den Endpunkten aus ein Stück nach außen verschoben wird, wird ein Zwei-Wege-Cursor angezeigt, und der Endpunkt kann nur in Richtung der Linie verschoben werden.

Die LOI muss möglichst senkrecht zu den zu messenden Kanten liegen, d.h. optimal unter 90 °. Je näher zu 0 ° die Suchlinie liegt, desto mehr werden die Kanten durch Mittelung verschliffen und desto weniger genau werden die Ergebnisse.

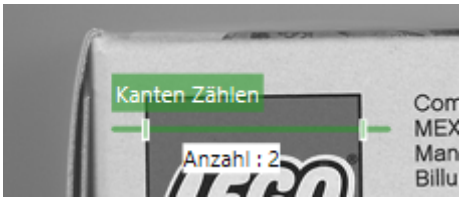


Abb. 56: Das Bild-GUI des Werkzeugs **Kanten Zählen**.

4.3.10 Formen Zählen



Abb. 57: Das Icon des Werkzeugs **Formen Zählen**.

Das Werkzeug **Formen Zählen** wird verwendet, um die Anzahl der Teile in einer Szene mithilfe von Mustersuche zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Formen Zählen**-Werkzeugs die **Anzahl** von Objekten.

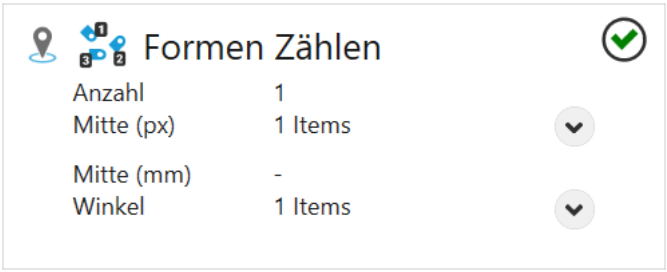


Abb. 58: Die Kachel des Werkzeugs **Formen Zählen**.

Das Werkzeug **Formen Zählen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Unter **Methode** stehen drei verschiedene Verfahren zur Lokalisierung von Bildbereichen zur Auswahl.

Bei der Methode **Korrelation** stehen Flächen im Fordergrund. Das Verfahren ist anwendbar bei gleichmäßigem Hintergrund.

Die Methode **Geometrisch** hat einen Fokus auf Kanten, und ist verlässlicher bei ungleichmäßigem Hintergrund und bei partiellen Verdeckungen.

Sowohl **Geometrisch** als auch **Korrelation** sollten in der Rotation eingeschränkt werden, da die Speicheranforderungen erheblich sind.

Die Methode **Keypoint** sucht markante Bildpunkte in der Vorlage und vergleicht sie mit denen im Suchbild. Dadurch erkennt das Verfahren bis zu einem gewissen Grad unterschiedliche Skalierungen und ist nicht anfällig bei partiellen Verdeckungen.

Die **Methode** ist nur im Trainieren-Modus wählbar.

Zulässige Rotation ermöglicht es, die Drehung der Teile zu beschränken. Dieser Parameter ist nur im Trainieren-Modus sichtbar.

Die zum Zählen verwendete **Vorlage** kann durch Klicken auf die Schaltfläche **Vorlage trainieren** erstellt werden. Sobald Sie eine Vorlage trainiert haben wechselt das Werkzeug in den Suchmodus, in dem die ROI (Region of Interest) für den Suchbereich ausgewählt wird. Mit einem Klick auf **Neue Vorlage** wechselt das Werkzeug zurück in den Trainieren-Modus.

Der trainierten Vorlage kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden. Die ausmaskierten Pixel werden für das Matching nicht berücksichtigt.

Mit **Details reduzieren** werden während der Template-Vorverarbeitung Bilddetails reduziert, um die Ausführungszeit zu verkürzen.

Mit **Match-Qualität** können schrittweise Einstellungen zwischen **Präzise** und **Schnell** vorgenommen werden. Schnelleres Matching ist weniger genau und präzisere Übereinstimmung ist langsamer. Auf diese Weise können Sie den richtigen Kompromiss für Ihre Anwendung auswählen.

Akzeptanzschwelle legt den Mindestschwellenwert für Übereinstimmungen in Prozent fest. Alle Übereinstimmungen unter diesem Schwellenwert werden ignoriert. Ein Schwellenwert von 50% ist in den meisten Fällen zu niedrig für gute Treffer.

Hintergrund entfernen schließt einheitliche Hintergrundbereiche von der Verarbeitung aus, wodurch die Verarbeitung beschleunigt werden kann. Falls der Hintergrund viele Details hat, sollten Sie diese Option nicht verwenden, da es die Verarbeitungszeit in solchen Fällen erhöhen kann.

Sortierung beeinflusst die Ordnung der Resultate in Bezug auf ihren Mittelpunkt.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Anzahl

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Im Trainieren-Modus wird mit der ROI das Muster ausgewählt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).



Abb. 59: Das Trainieren-GUI des Werkzeugs **Formen Zählen**.

Im Suchmodus wird die ROI verwendet, um den Suchbereich auszuwählen. Innerhalb des Suchbereichs wird das Muster gesucht, gezählt und angezeigt.



Abb. 60: Das Suchen-GUI des Werkzeugs **Formen Zählen**.

4.3.11 Barcodes Zählen



Abb. 61: Das Icon des Werkzeugs **Barcodes Zählen**.

Das Werkzeug **Barcodes Zählen** wird verwendet, um die Anzahl der Barcodes in einer Szene zu zählen und zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Barcodes Zählen**-Werkzeugs die **Anzahl** von Barcodes.

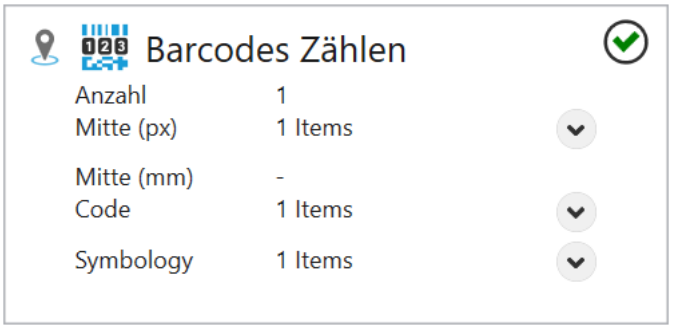


Abb. 62: Die Kachel des Werkzeugs **Barcodes Zählen**.

Das Werkzeug **Barcodes Zählen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Code

Die Symbologie kann **Auto**matisch oder **Manuell** erkannt werden. Alternativ kann der Code auch durch **Verknüpfung** durch die Eingänge gesetzt werden.

Klicken Sie im automatischen Modus auf **Code identifizieren**, um die Symbologie automatisch zu erkennen.

Filtern Sie im **Manuell**en Modus nach **1D** - oder **2D** -Codes und wählen Sie die gewünschte Symbologie aus der Liste **Code** aus.

Um den Code durch **Verknüpfung** einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Code, wie er unter **Verfügbare Codes** zu finden ist, entsprechen. Um mehrere Codes gleichzeitig zu erkennen, kann der Text auch mehrere der verfügbaren Codes beinhalten.

Parameter

Der vom Decoder benötigte **Kontrast** kann durch Aktivieren von **Niedrig** oder **Steigern** beeinflusst werden, wodurch die Leseleistung auf Kosten der Laufzeit erhöht wird.

Die **Genauigkeit** kann in vier Schritten zwischen **Präzise** und **Schnell** eingestellt werden.

Mit **DPM Markierung** kann die Dekodierung bei direkt markierten Codes beeinflusst werden:

Kein DPM (Direct Part Marking): keine spezielle Dekodierung von direkt markierten Codes. **Gelaster/Geätzt**: liest gelaserte oder chemisch geätzte direkt markierte Codes. **Tintenstrahldruck/Genadelt**: liest mit Tintenstrahldruckern gedruckte direkt markierte Codes.

Wenn diese einfachen Methoden nicht ausreichen, gibt es mit der ID-Pro Lizenz erweiterte Möglichkeiten:

Pro: Genadelt dunkel auf hell: robuste Methode für dunkel auf hell genadelte direkt markierte Codes. **Pro: Genadelt hell auf dunkel**: robuste Methode für hell auf dunkel genadelte direkt markierte Codes. **Pro: Gelasert/Geätzt**: robuste Methode für gelaserte und geätzte direkt markierte Codes. **Pro: Genadelt/Gelasert/Geätzt**: robuste Methode für genadelte, gelaserte und geätzte direkt markierte Codes (aber möglicherweise etwas langsamer als die spezialisierten Methoden).

Mit **Verifizierung** kann die Verifizierungsmethode für 2D Codes ausgewählt werden:

Keine: es wird keine Verifizierung durchgeführt. **ISO/IEC 15415**: es wird eine Verifizierung angelehnt an den ISO/IEC 15415 Standard durchgeführt. **ISO/IEC TR29158**: es wird eine Verifizierung angelehnt an den ISO/IEC TR29158 Standard durchgeführt.

Die beiden Verifizierungsmodi sind nur verfügbar, wenn gleichzeit eine **Pro** Methode bei **DPM Markierung** ausgewählt ist.

Sortierung beeinflusst die Ordnung der Resultate in Bezug auf ihren Mittelpunkt.

Die maximale Zeit für die Dekodierung kann mit **Dekodier Timeout (ms)** in Millisekunden eingestellt werden (bis zu 60.000 Millisekunden). Der voreingestellte Wert von 0 bedeutet kein Timeout.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte**

(Erode) führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Anzahl

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn die **Anzahl** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Anzahl der Barcodes gezählt und mit farbigen grafischen Überlagerungen angezeigt. Zusätzlich wird die **Anzahl** angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

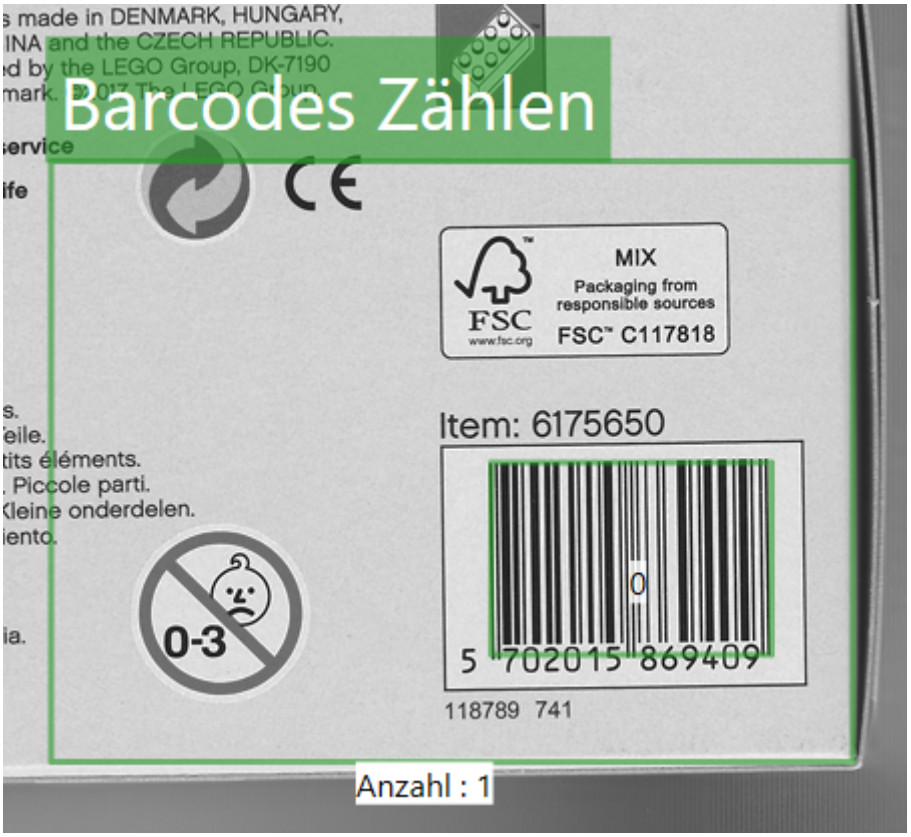


Abb. 63: Das Bild-GUI des Werkzeugs **Barcodes Zählen**.

4.3.12 Barcode Erkennen



Abb. 64: Das Icon des Werkzeugs **Barcode Erkennen**.

Das Werkzeug **Barcode Erkennen** wird verwendet, um einen Barcode oder einen Matrixcode innerhalb einer ROI (Region of interest) zu dekodieren.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Barcode Erkennen**-Werkzeugs der dekodierte **Code**.



Abb. 65: Die Kachel des Werkzeugs **Barcode Erkennen**.

Das Werkzeug **Barcode Erkennen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Code

Die Symbologie kann **Auto**matisch oder **Manuell** erkannt werden. Alternativ kann der Code auch durch **Verknüpfung** durch die Eingänge gesetzt werden.

Klicken Sie im automatischen Modus auf **Code identifizieren**, um die Symbologie automatisch zu erkennen.

Filtern Sie im **Manuell**en Modus nach **1D** - oder **2D** -Codes und wählen Sie die gewünschte Symbologie aus der Liste **Code** aus.

Um den Code durch **Verknüpfung** einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Code, wie er unter **Verfügbare Codes** zu finden ist, entsprechen.

Einstellungen

Jede der Symbologien kann spezifische Einstellungen haben.

Code 128

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Code 39

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Die **Prüfsumme** kann auf **Deaktivieren**, **Aktivieren** oder **Aktivieren und Abschneiden** eingestellt werden. Letzere aktiviert die Prüfung aber entfernt die Prüfsumme vom Ergebnis.

Vollständiges Ascii aktiviert die vollständige ASCII-Dekodierung.

Anfang/Ende abschneiden entfernt die Start- und Stoppzeichen aus dem Ergebnis.

Code 93

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Codabar

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Die **Prüfsumme** kann auf **Deaktivieren**, **Aktivieren** oder **Aktivieren und Abschneiden** eingestellt werden. Letzere aktiviert die Prüfung aber entfernt die Prüfsumme vom Ergebnis.

Anfang/Ende abschneiden entfernt die Start- und Stoppzeichen aus dem Ergebnis.

Ratio ermöglicht das Decodieren mit einem Verhältnis der Balkenbreiten außerhalb der Spezifikation von 5,5.

Pharmacode

Pharmacode ist ein Code ohne Fehlerprüfung. Die Einstellungen tragen zur Reduzierung von Lesefehlern bei. **Min. Striche** und **Max. Striche** legen den Bereich der zulässigen Anzahl von Strichen fest. **Min. Wert** und **Max. Wert** legen den zulässigen Zahlenbereich fest. **Umgekehrte Dekodierung** legt die Richtung der Dekodierung fest.

UPC/EAN

Erweiterung aktiviert oder deaktiviert die UPC E-Erweiterung.

Ergänzung 2 aktiviert oder deaktiviert die Dekodierung des zweistelligen Zusatzcodes für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien.

Ergänzung 5 aktiviert oder deaktiviert die Dekodierung des fünfstelligen Zusatzcodes für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien.

Leerzeichen an Ergänzung hinzufügen aktiviert das Leerzeichen zwischen dem UPC / EAN-Code und dem 2- oder 5-stelligen Zusatzcode für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien. Gilt nur, wenn die Decodierung des entsprechenden Zusatzcodes aktiviert ist.

Ergänzung erforderlich erfordert, dass der Zusatzcode für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien dekodiert werden kann.

Zahl entfernen aktiviert oder deaktiviert das Entfernen der Zahlensystemziffer in decodierten Daten für UPC-A- und UPC-E-Symbologien.

Buchstabe entfernen aktiviert oder deaktiviert das Entfernen der Prüfziffer für UPC-A-, UPC-E-, EAN13- und EAN8-Symbologien.

Code 2of5

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Die **Prüfsumme** kann auf **Deaktivieren**, **Aktivieren** oder **Aktivieren und Abschneiden** eingestellt werden. Letzere aktiviert die Prüfung aber entfernt die Prüfsumme vom Ergebnis.

Partielles Ablehnen aktiviert oder deaktiviert die Ablehnung der partiellen 2-von-5-Decodierung. Wenn Sie diese Funktion aktivieren, ist es weniger wahrscheinlich, dass ein Code teilweise decodiert wird, wenn er beschädigt oder aufgrund seiner Platzierung im Bild abgeschnitten ist (z. B. wenn ein Teil des Barcodes fehlt).

Aztec

Die **Polarität** kann auf **Dunkel auf Hell**, **Hell auf Dunkel** oder **Beide** eingestellt werden.

Wenn **Spiegeln** aktiviert ist, werden auch von hinten gesehene Codes dekodiert.

Data Matrix

Die **Polarität** kann auf **Dunkel auf Hell**, **Hell auf Dunkel** oder **Beide** eingestellt werden.

Wenn **Spiegeln** aktiviert ist, werden auch von hinten gesehene Codes dekodiert.

Rechteckig aktiviert oder deaktiviert rechteckige Codes.

Erweitert aktiviert oder deaktiviert erweiterte rechteckige Codes.

Dicker Rand findet auch Codes mit Modulgrößen größer als 20 Pixel.

QR Code

ECC-Codewörter senden legt fest, ob Fehlerkorrektur-Codewörter in den Ergebnissen verfügbar sind.

Parameter

Der vom Decoder benötigte **Kontrast** kann durch Aktivieren von **Niedrig** oder **Steigern** beeinflusst werden, wodurch die Leseleistung auf Kosten der Laufzeit erhöht wird.

Die **Genauigkeit** kann in vier Schritten zwischen **Präzise** und **Schnell** eingestellt werden.

Mit **DPM Markierung** kann die Dekodierung bei direkt markierten Codes beeinflusst werden.

Kein DPM (Direct Part Marking): keine spezielle Dekodierung von direkt markierten Codes. **Gelasert/Geätzt**: liest gelaserte oder chemisch geätzte direkt markierte Codes. **Tintenstrahldruck/Genadelt**: liest mit Tintenstrahldruckern gedruckte direkt markierte Codes.

Wenn diese einfachen Methoden nicht ausreichen, gibt es mit der ID-Pro Lizenz erweiterte Möglichkeiten:

Pro: Genadelt dunkel auf hell: robuste Methode für dunkel auf hell genadelte direkt markierte Codes. **Pro: Genadelt hell auf dunkel**: robuste Methode für hell auf dunkel genadelte direkt markierte Codes. **Pro: Gelasert/Geätzt**: robuste Methode für gelaserte und geätzte direkt markierte Codes. **Pro: Genadelt/Gelasert/Geätzt**: robuste Methode für genadelte, gelasterte und geätzte direkt markierte Codes (aber möglicherweise etwas langsamer als die spezialisierten Methoden).

Mit **Verifizierung** kann die Verifizierungsmethode für 2D Codes ausgewählt werden:

Keine: es wird keine Verifizierung durchgeführt. **ISO/IEC 15415**: es wird eine Verifizierung angelehnt an den ISO/IEC 15415 Standard durchgeführt. **ISO/IEC TR29158**: es wird eine Verifizierung angelehnt an den ISO/IEC TR29158 Standard durchgeführt.

Die beiden Verifizierungsmodi sind nur verfügbar, wenn gleichzeit eine **Pro** Methode bei **DPM Markierung** ausgewählt ist.

Text wenn nicht lesbar: das ist der Text, der ausgegeben wird, falls kein Code gelesen werden konnte.

Die maximale Zeit für die Dekodierung kann mit **Dekodier Timeout (ms)** in Millisekunden eingestellt werden (bis zu 60.000 Millisekunden). Der voreingestellte Wert von 0 bedeutet kein Timeout.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Erwarteter Code

Der dekodierte Inhalt wird mit dem **Erwarteten Code** verglichen, gemäß der **Methode**, die **Keine**, **Entspricht**, **Beinhaltet**, **Beginnt mit** oder **Endet mit** sein kann.

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn der Inhalt passt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird der Barcode dekodiert und der **Code** angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

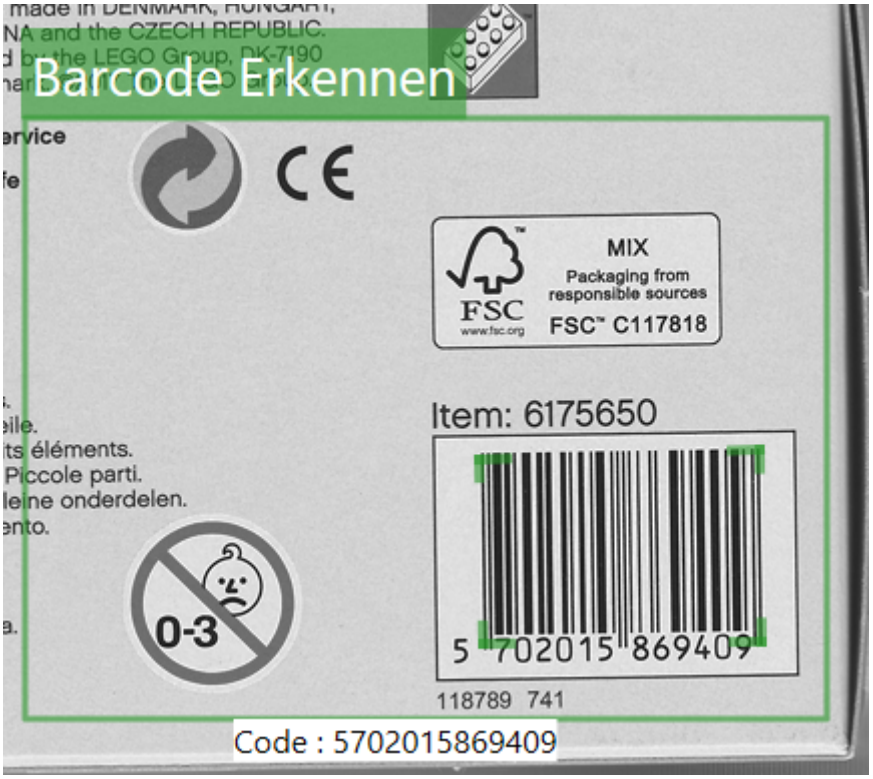


Abb. 66: Das Bild-GUI des Werkzeugs **Barcode Erkennen**.

4.3.13 Text Erkennen



Abb. 67: Das Icon des Werkzeugs **Text Erkennen**.

Das Werkzeug **Text Erkennen** wird verwendet, um einen Text innerhalb einer ROI (Region of interest) zu erkennen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Text Erkennen**-Werkzeugs der erkannte **Text**.



Abb. 68: Die Kachel des Werkzeugs **Barcode Erkennen**.

Das Werkzeug **Text Erkennen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Schriftart

Die Schriftart kann aus einer Liste gewählt werden.

Es gibt mehrere Gruppen von Schriften. Die Mitglieder der einzelnen Gruppen unterscheiden sich, da sie unterschiedliche Symbolsätze enthalten. Der Inhalt einer Schriftart wird durch ihren Namen beschrieben. Für die Namen der vortrainierten Schriften gilt die folgende Nomenklatur:

Der Name beginnt mit dem Gruppennamen, z. B. **Document** oder **DotPrint**, gefolgt von Indikatoren für den Symbolsatz die in der OCR-Schriftart enthalten sind. Die Bedeutung der Indikatoren ist die folgende:

- **0-9**: Die OCR-Schriftart enthält die Ziffern 0 bis 9.
- **A-Z**: Die OCR-Schriftart enthält die Großbuchstaben A bis Z.
- **+** : Die OCR-Schriftart enthält Sonderzeichen. Die Liste der Sonderzeichen variiert leicht zwischen den einzelnen OCR-Schriften. Sie ist im Folgenden für jede OCR-Schriftart einzeln aufgeführt.

Wenn der Name der OCR-Schriftart keines der oben genannten Kennzeichen enthält, enthält die OCR-Schriftart normalerweise die Ziffern 0 bis 9, die Großbuchstaben A bis Z, die Kleinbuchstaben Kleinbuchstaben a bis z und Sonderzeichen. Einige OCR-Schriften enthalten keine Kleinbuchstaben (z. B., DotPrint). Dies wird in der Beschreibung der jeweiligen Schriftart ausdrücklich erwähnt.

Um den Zeichensatz durch Verknüpfung einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Zeichensatz, wie er unter **Font** zu finden ist, entsprechen.

`Document`

Die Schriftart `Dokument` kann verwendet werden, um Zeichen zu lesen, die in Schriftarten wie Arial, Courier oder Times New Roman gedruckt wurden. Dies sind typische Schriftarten für den Druck von Dokumenten oder Briefen.

Beachten Sie, dass die Zeichen I und l der Schriftart Arial nicht unterschieden werden können. Das bedeutet, dass ein `l` fälschlicherweise für ein `I` gehalten werden kann und umgekehrt.

Verfügbare Sonderzeichen: - = + < > . # $ % & ( ) @ * e £ ¥

`DotPrint`

Die OCR-Schriftart DotPrint kann verwendet werden, um mit Punktdruckern gedruckte Zeichen zu lesen. Sie enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . * :

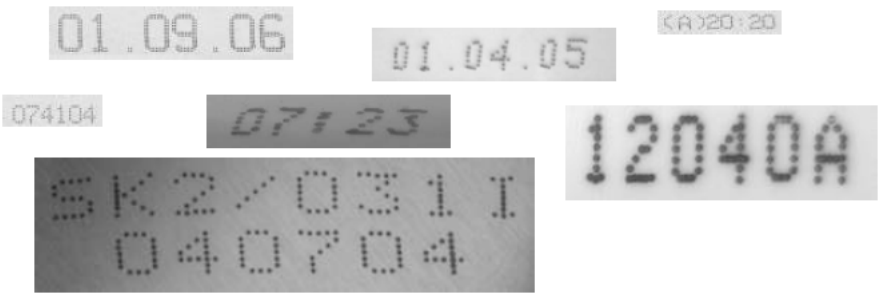


Abb. 69: Beispiele für gepunktete Schrift.

HandWritten

Die Schriftart HandWritten_0-9 kann zum Lesen handgeschriebener Zahlen verwendet werden.

Sie enthält die Ziffern 0-9.

Verfügbare Sonderzeichen: keine

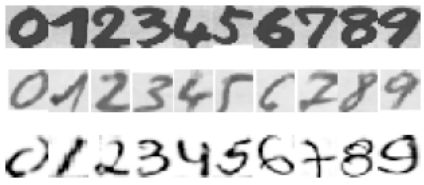


Abb. 70: Beispiele für handgeschriebene Ziffern.

Industrial

Die Schriftart Industrial kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B oder anderen serifenlosen Schriften Schriftarten gedruckt wurden. Diese Schriftarten werden normalerweise zum Drucken von z. B. Etiketten verwendet.

Verfügbare Sonderzeichen: - / + . $ % * e £ ¥

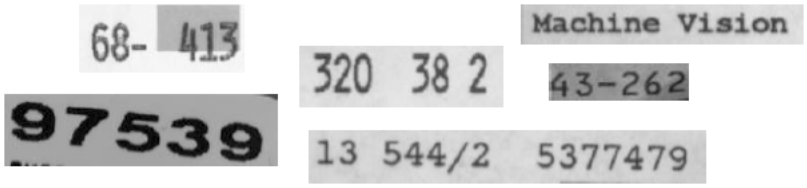


Abb. 71: Beispiele für industrielle Drucke.

OCR-A

Die Schrift OCR-A kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-A gedruckt sind.

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



Abb. 72: Ausgewählte Zeichen der OCR-A Schrift.

OCR-B

Die Schrift OCR-B kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-B gedruckt sind.

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



Abb. 73: Ausgewählte Zeichen der OCR-B Schrift.

Pharma

Die Schriftart Pharma kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B und anderen Schriftarten gedruckt sind, die typischerweise in der pharmazeutischen Industrie verwendet werden.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . ( ) :

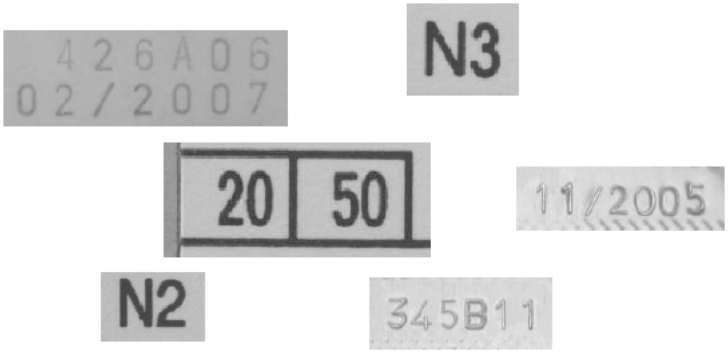


Abb. 74: Beispiele für pharmazeutische Drucke.

SEMI

Die Schriftart SEMI kann verwendet werden, um Zeichen zu lesen, die in der Schriftart SEMI gedruckt sind, die aus Zeichen besteht, die die so gestaltet sind, dass sie sich leicht voneinander unterscheiden lassen. Sie hat einen begrenzten Satz von Zeichen.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - .



Abb. 75: Zeichensatz der SEMI Schriftart.

Universal

Die OCR-Schriftart Universal kann zum Lesen einer Vielzahl unterschiedlicher Zeichen verwendet werden. Das Training dieser CNN-basierten Schriftart basiert auf den Zeichensätzen ‘Document’, ‘DotPrint’, ‘SEMI’ und ‘Industrial’.

Verfügbare Sonderzeichen: - / = + : < > . # $ % & ( ) @ * e £ ¥

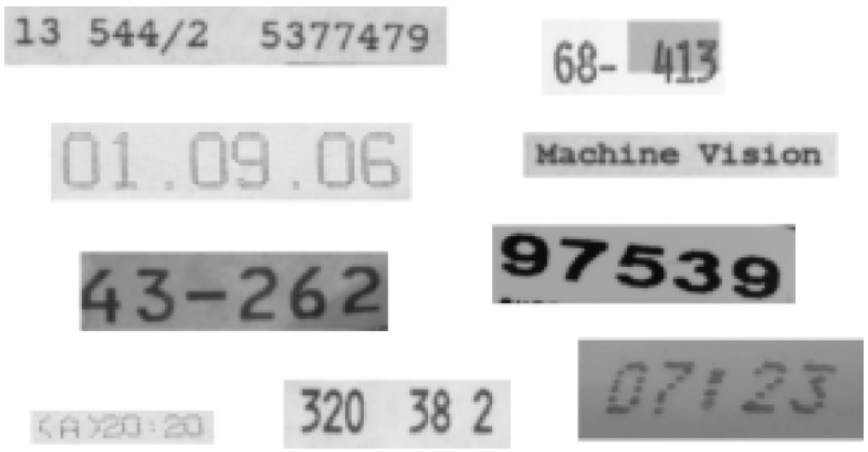


Abb. 76: Beispile für den Zeichensatz Universal.

Segmentierung

Min. Kontrast bestimmt den minimalen Kontrast zwischen Buchstaben und ihrem Hintergrund. Der Wert kann zwischen 1 und 255 eingestellt werden, der Standardwert ist 15.

Die **Polarität** kann auf **Dunkel auf Hell** für dunkle Zeichen auf hellem Hintergrund, **Hell auf Dunkel** für helle Zeichen auf dunklem Hintergrund oder auf **Beide** eingestellt werden, wenn beide Arten von Text segmentiert werden sollen.

Wenn **Rand eliminieren** angewählt ist, werden Regionen verworfen, die den Rand der ROI berühren.

Wenn **Fragmente hinzufügen** angewählt ist, werden Fragmente wie der Punkt auf einem “i” zu den segmentierten Buchstaben hinzugefügt. Das kann allerdings auch dazu führen, dass Störungen zu den segmentierten Buchstaben hinzugefügt werden.

Berührende Zeichen bestimmt die Trennung von Paaren oder kleinen Gruppen nebeneinander liegender Zeichen, die als eine einzige Region segmentiert werden. Wird **Standard** gewählt, werden solche Regionen erkannt und in zwei oder mehrere einzelne Zeichen aufgeteilt. Wird der Werte auf **falsch** gesetzt, wird keine Trennung von sich berührenden Zeichen durchgeführt.

Zeichengröße

Minimale Zeichenhöhe bestimmt die minimale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Maximale Zeichenhöhe bestimmt die maximale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Minimale Zeichenbreite bestimmt die minimale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Maximale Zeichenbreite bestimmt die maximale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Minimale Strichstärke bestimmt die minimale Strichstärke eines Buchstabens in Pixel. Soll die minimale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Minimale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Maximale Strichstärke bestimmt die maximale Strichstärke eines Buchstabens in Pixel. Soll die maximale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Maximale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Sonderzeichen

Wenn **Satzzeichen** angewählt ist, werden kleine Satzzeichen, die nahe an der Grundlinie der Textzeile liegen (wie Punkt oder Komma), zu den segmentierten Buchstaben hinzugefügt.

Wenn **Trennzeichen** angewählt ist, werden Trennzeichen wie ein Minus oder das Gleichheitszeichen zu den segmentierten Buchstaben hinzugefügt.

Punktdruck

Punktdruck muss angewählt werden, falls der zu segmentierende Text Punktdruck-Buchstaben enthält.

Enge Zeichen muss angewählt werden, wenn die Lücken zwischen benachbarten Buchstaben kleiner sind als die Lücke zwischen zwei Punkten innerhalb eines Buchstabens. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die genaue Größe der kleinsten Lücke zwischen benachbarten Buchstaben bekannt, kann diese stattdessen mittels **Minimale Zeichenlücke** gesetzt werden. In diesem Fall wird die Einstellung von **Enge Zeichen** ignoriert.

Minimale Zeichenlücke ist die Größe der kleinsten Lücke zwischen zwei Buchstaben in Pixel. Dieser Parameter kann in Fällen, in denen die Lücken zwischen Buchstaben kleiner sind als die zwischen Punkten innerhalb der Buchstaben, zu einer genaueren Segmentierung führen. Falls die minimale Größe nicht bekannt oder größer als die maximale Lücke zwischen den Punkten innerhalb eines Buchstaben ist, kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Sind Buchstaben sehr nah nebeneinander gedruckt, jedoch die Größe der minimalen Lücke unbekannt, kann stattdessen **Enge Zeichen** gesetzt werden.

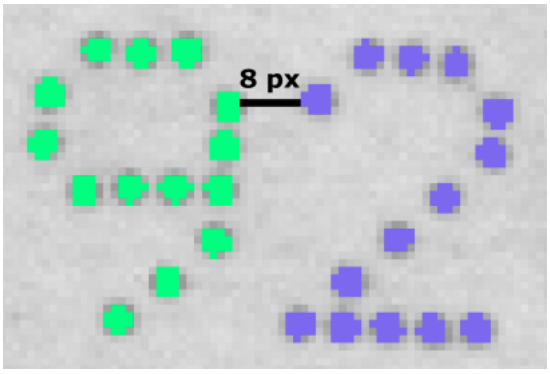


Abb. 77: Hier ist die kleinste Lücke zwischen zwei Buchstaben 8 Pixel.

Maximale Punktelücke ist die Größe der größten Lücke zwischen zwei Punkten innerhalb eines Buchstabens in Pixel. Um beliebige Punktdruck-Buchstaben zu segmentieren kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die größte Lücke zwischen zwei Punkten größer als die kleinste Lücke zwischen zwei Buchstaben, sollte zusätzlich **Enge Zeichen** angewählt werden oder **Minimale Zeichenlücke** entsprechend gesetzt werden. Wird **Maximale Punktelücke**gesetzt, kann die Performance der Texterkennung deutlich gesteigert werden.

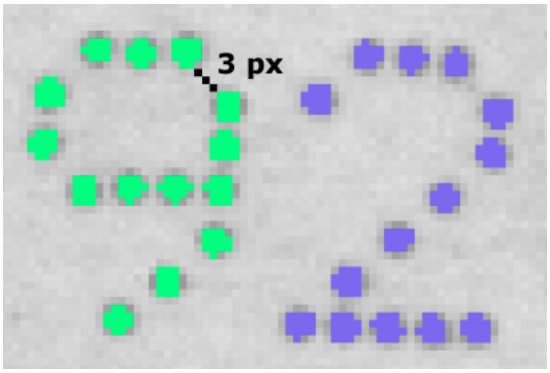


Abb. 78: Hier ist die größte Lücke zwischen zwei Punkten 3 Pixel.

Struktur

Zeilenstruktur legt die Struktur des gesuchten Textes fest. Für jede Zeile wird der Abstand zwischen aufeinanderfolgenden Buchstaben berechnet. Ausgehend von diesen Abständen wird die Zeile in einzelne Buchstabenblöcke aufgeteilt. Hierbei werden sehr kleine Zeichen wie z.B. ., _ und – vernachlässigt und als Leerzeichen interpretiert. Darüber hinaus können zusätzliche Trennzeichen angegeben werden, die ebenfalls vernachlässigt werden sollen, siehe **Zeilentrennzeichen**. Für die entstandenen Buchstabenblöcke wird dann geprüft ob sie einer der übergebenen Strukturen entsprechen.

Eine Struktur besteht aus einem String, der die Anzahl der Buchstaben/Ziffern pro Buchstabenblock enthält, welche durch Leerzeichen getrennt sind. Soll zum Beispiel ein Datum mit jeweils zwei Ziffern für Tag, Monat und Jahr gefunden werden, wäre die Struktur `2 2 2`. Kann das Jahr auch aus vier Ziffern bestehen, wäre die Struktur `2 2 2-4`, so dass der letzte Block aus zwei bis vier Buchstaben/Ziffern bestehen kann. Wenn die Struktur einem leeren Text entspricht, kann der zu findende Texte aus jeder beliebigen Struktur bestehen.

Jede gefundene Struktur wird als einzelne Zeile gespeichert. Soll stattdessen die komplette Zeile zurückgeliefert werden, die eine Struktur enthält, kann **Ganze Zeile** angewählt werden.

Zeilentrennzeichen ist ein String von Trennzeichen, die bei der Suche nach Strukturen vernachlässigt werden sollen, siehe **Zeilenstruktur**. Hierbei ist zu beachten, dass ein solches Trennzeichen immer auch im Klassifikator enthalten sein muss. Sollen z.B. die Zeichen `:` und `\` als Trennzeichen behandelt werden, kann der String `:\\` übergeben werden. Hierbei ist zu beachten, dass das vorangestellte `\` die Verwendung des Symbols `\` als Literal ermöglicht.

Ganze Zeile falls nur die gefundenen Strukturen als Zeilen zurückgegeben werden sollen, `false` falls jede Zeile, die eine passende Stuktur enthält komplett zurückgegeben werden soll.

Parameters

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Anzahl der Zeichen gezählt und mit farbigen grafischen Überlagerungen angezeigt. Zusätzlich wird die **Anzahl** angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

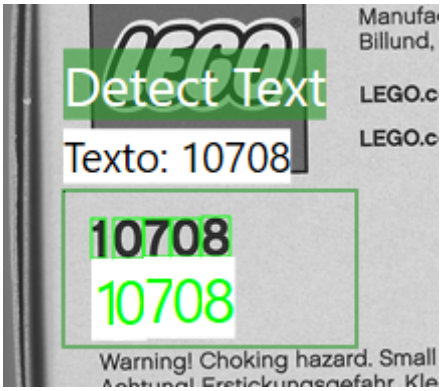


Abb. 79: Das Bild-GUI des Werkzeugs **Text Erkennen**.

4.3.14 Fläche Lokalisieren



Abb. 80: Das Icon des Werkzeugs **Fläche Lokalisieren**.

Mit dem Werkzeug **Fläche Lokalisieren** können Sie mithilfe der Blob-Analyse ein Teil in einer Szene lokalisieren.

Wenn das Werkzeug **Fläche Lokalisieren** ein Teil lokalisieren kann, wird die Position und optional die Drehung des Teils bestimmt und an andere Werkzeuge weitergeleitet. Das Hauptergebnis des Werkzeugs **Fläche Lokalisieren** ist eine **Pose**, die Position und Rotation beinhaltet.

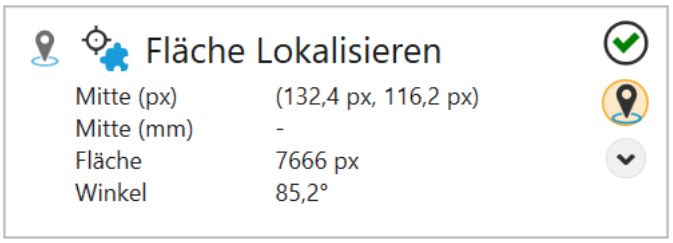


Abb. 81: Die Kachel des Werkzeugs **Fläche Lokalisieren**.

Das Werkzeug **Fläche Lokalisieren** zeigt die **Pose** sowie die **Fläche** des Teils in Pixel an.

Werkzeuge in der Pipeline können diese Pose verwenden. Sie können mehrere Ortungswerkzeuge in einer Pipeline verwenden, und in einem beliebigen Werkzeug können Sie die Pose auswählen, indem Sie sie aus dem angezeigten Kontextmenü auswählen wenn Sie sie mit der Pin-Taste auswählen.

Parameter

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken).

Ein **Rechteck** kann auch gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Beim **Polygon** können Ecken durch Klicken auf eine Randlinie hinzugefügt werden, bzw. durch Drücken der Delete Taste entfernt werden.

Einer ROI kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden.

Der **Schwellenwert** wird verwendet, um **Dunkle Flächen**, **Helle Flächen**, **Manuell** oder **Adaptiv** auszuwählen. Was dunkel oder hell ist, wird über einen Schwellenwert bestimmt, der automatisch berechnet wird, oder manuell mit einem Schieberegler eingestellt wird, unter Verwendung der logarithmisch skalierten Histogramm-Anzeige des Bildes. Der adaptive Modus arbeitet mit einem prozentualen **Schwellenwert** innerhalb einer **Fenstergröße**, optional kann die Einstellung **Löcher füllen** gewählt werden.

Rotation verfolgen kann verwendet werden, um die Drehung eines Teils zusätzlich zu seiner Position zu verfolgen. Verfolgung der Rotation ist nur mit Teilen möglich, die eine inhärente Vorzugsrichtung haben (z. B. ein kreisförmiges Teil hat unendlich viele Richtungen, damit ist seine Richtung bedeutungslos. Ein Quadrat hat vier mögliche Richtungen, was auch bedeutet, dass eine bevorzugte Richtung nicht berechnet werden kann.)

Flächen verbinden kombiniert separate Flächen, wenn aktiviert; wenn diese Option deaktiviert ist, wird nur die größte Fläche verwendet.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Filterung

Glättung unterzieht die Bilddaten einer einstellbaren Bildglättung.

Flächen die die ROI berühren ignorieren ignoriert alle Objekte, welche den Rand der interaktiv auf der rechten Seite eingestellten ROI berühren.

Im Modus **Fläche filtern** können **Minimale Fläche** und **Maximale Fläche** verwendet werden, um einen Größenbereich in Quadrat-Pixeln anzugeben, welcher die Flächen eingrenzt.

Im Modus **Größe filtern** können **Minimale Breite**, **Maximale Breite**, **Minimale Höhe** und **Maximale Höhe** verwendet werden um einen Größenbereich anzugeben, welcher die Flächen eingrenzt.

Im Modus **Rundheit filtern** kann die **Rundheit** eingegrenzt werden, um Flächen nach ihrer Form einzugrenzen.

Optionale Merkmale

Das Werkzeug kann zusätzliche Parameter der detektierten Objekte berechnen. Da diese Berechnung Zeit kostet, können die zusätzlichen Merkmale nach Bedarf ausgewählt werden.

Die optionalen Merkmale sind genauer beschrieben im Kapitel **Blob Analyse** im Handbuch.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer ROI. Innerhalb der ROI wird die **Fläche** berechnet und angezeigt.



Abb. 82: Das Bild-GUI des Werkzeugs **Fläche Lokalisieren**.

4.3.15 Barcode Lokalisieren



Abb. 83: Das Icon des Werkzeugs **Barcode Lokalisieren**.

Mit dem Werkzeug **Barcode Lokalisieren** wird ein Barcode in einer Szene lokalisiert.

Wenn das Werkzeug **Barcode Lokalisieren** einen Barcode lokalisieren kann, wird die Position und die Drehung des Barcodes bestimmt und an andere Werkzeuge weitergeleitet. Das Hauptergebnis des Werkzeugs **Barcode Lokalisieren** ist eine **Pose**, die Position und Rotation beinhaltet.



Abb. 84: Die Kachel des Werkzeugs **Barcode Lokalisieren**.

Das Werkzeug **Barcode Lokalisieren** zeigt die **Pose** des Barcodes in Pixel an.

Werkzeuge in der Pipeline können diese Pose verwenden. Sie können mehrere Ortungswerkzeuge in einer Pipeline verwenden, und in einem beliebigen Werkzeug können Sie die Pose auswählen, indem Sie sie aus dem angezeigten Kontextmenü auswählen wenn Sie sie mit der Pin-Taste auswählen.

Code

Die Symbologie kann **Auto**matisch oder **Manuell** erkannt werden. Alternativ kann der Code auch durch **Verknüpfung** durch die Eingänge gesetzt werden.

Klicken Sie im automatischen Modus auf **Code identifizieren**, um die Symbologie automatisch zu erkennen.

Filtern Sie im **Manuell**en Modus nach **1D** - oder **2D** -Codes und wählen Sie die gewünschte Symbologie aus der Liste **Code** aus.

Um den Code durch **Verknüpfung** einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Code, wie er unter **Verfügbare Codes** zu finden ist, entsprechen.

Einstellungen

Jede der Symbologien kann spezifische Einstellungen haben.

Code 128

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Code 39

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Die **Prüfsumme** kann auf **Deaktivieren**, **Aktivieren** oder **Aktivieren und Abschneiden** eingestellt werden. Letzere aktiviert die Prüfung aber entfernt die Prüfsumme vom Ergebnis.

Vollständiges Ascii aktiviert die vollständige ASCII-Dekodierung.

Anfang/Ende abschneiden entfernt die Start- und Stoppzeichen aus dem Ergebnis.

Code 93

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Codabar

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Die **Prüfsumme** kann auf **Deaktivieren**, **Aktivieren** oder **Aktivieren und Abschneiden** eingestellt werden. Letzere aktiviert die Prüfung aber entfernt die Prüfsumme vom Ergebnis.

Anfang/Ende abschneiden entfernt die Start- und Stoppzeichen aus dem Ergebnis.

Ratio ermöglicht das Decodieren mit einem Verhältnis der Balkenbreiten außerhalb der Spezifikation von 5,5.

Pharmacode

Pharmacode ist ein Code ohne Fehlerprüfung. Die Einstellungen tragen zur Reduzierung von Lesefehlern bei. **Min. Striche** und **Max. Striche** legen den Bereich der zulässigen Anzahl von Strichen fest. **Min. Wert** und **Max. Wert** legen den zulässigen Zahlenbereich fest. **Umgekehrte Dekodierung** legt die Richtung der Dekodierung fest.

UPC/EAN

Erweiterung aktiviert oder deaktiviert die UPC E-Erweiterung.

Ergänzung 2 aktiviert oder deaktiviert die Dekodierung des zweistelligen Zusatzcodes für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien.

Ergänzung 5 aktiviert oder deaktiviert die Dekodierung des fünfstelligen Zusatzcodes für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien.

Leerzeichen an Ergänzung hinzufügen aktiviert das Leerzeichen zwischen dem UPC / EAN-Code und dem 2- oder 5-stelligen Zusatzcode für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien. Gilt nur, wenn die Decodierung des entsprechenden Zusatzcodes aktiviert ist.

Ergänzung erforderlich erfordert, dass der Zusatzcode für UPC-A-, UPC-E-, EAN-13- und EAN-8-Symbologien dekodiert werden kann.

Zahl entfernen aktiviert oder deaktiviert das Entfernen der Zahlensystemziffer in decodierten Daten für UPC-A- und UPC-E-Symbologien.

Buchstabe entfernen aktiviert oder deaktiviert das Entfernen der Prüfziffer für UPC-A-, UPC-E-, EAN13- und EAN8-Symbologien.

Code 2of5

Codes mit kürzeren Inhalten als **Min. Länge** oder längeren Inhalten als **Max. Länge** werden nicht akzeptiert.

Die **Prüfsumme** kann auf **Deaktivieren**, **Aktivieren** oder **Aktivieren und Abschneiden** eingestellt werden. Letzere aktiviert die Prüfung aber entfernt die Prüfsumme vom Ergebnis.

Partielles Ablehnen aktiviert oder deaktiviert die Ablehnung der partiellen 2-von-5-Decodierung. Wenn Sie diese Funktion aktivieren, ist es weniger wahrscheinlich, dass ein Code teilweise decodiert wird, wenn er beschädigt oder aufgrund seiner Platzierung im Bild abgeschnitten ist (z. B. wenn ein Teil des Barcodes fehlt).

Aztec

Die **Polarität** kann auf **Dunkel auf Hell**, **Hell auf Dunkel** oder **Beide** eingestellt werden.

Wenn **Spiegeln** aktiviert ist, werden auch von hinten gesehene Codes dekodiert.

Data Matrix

Die **Polarität** kann auf **Dunkel auf Hell**, **Hell auf Dunkel** oder **Beide** eingestellt werden.

Wenn **Spiegeln** aktiviert ist, werden auch von hinten gesehene Codes dekodiert.

Rechteckig aktiviert oder deaktiviert rechteckige Codes.

Erweitert aktiviert oder deaktiviert erweiterte rechteckige Codes.

Dicker Rand findet auch Codes mit Modulgrößen größer als 20 Pixel.

QR Code

ECC-Codewörter senden legt fest, ob Fehlerkorrektur-Codewörter in den Ergebnissen verfügbar sind.

Parameter

Der vom Decoder benötigte **Kontrast** kann durch Aktivieren von **Niedrig** oder **Steigern** beeinflusst werden, wodurch die Leseleistung auf Kosten der Laufzeit erhöht wird.

Die **Genauigkeit** kann in vier Schritten zwischen **Präzise** und **Schnell** eingestellt werden.

Mit **DPM Markierung** kann die Dekodierung bei direkt markierten Codes beeinflusst werden.

Kein DPM (Direct Part Marking): keine spezielle Dekodierung von direkt markierten Codes. **Gelasert/Geätzt**: liest gelaserte oder chemisch geätzte direkt markierte Codes. **Tintenstrahldruck/Genadelt**: liest mit Tintenstrahldruckern gedruckte direkt markierte Codes.

Wenn diese einfachen Methoden nicht ausreichen, gibt es mit der ID-Pro Lizenz erweiterte Möglichkeiten:

Pro: Genadelt dunkel auf hell: robuste Methode für dunkel auf hell genadelte direkt markierte Codes. **Pro: Genadelt hell auf dunkel**: robuste Methode für hell auf dunkel genadelte direkt markierte Codes. **Pro: Gelasert/Geätzt**: robuste Methode für gelaserte und geätzte direkt markierte Codes. **Pro: Genadelt/Gelasert/Geätzt**: robuste Methode für genadelte, gelasterte und geätzte direkt markierte Codes (aber möglicherweise etwas langsamer als die spezialisierten Methoden).

Mit **Verifizierung** kann die Verifizierungsmethode für 2D Codes ausgewählt werden:

Keine: es wird keine Verifizierung durchgeführt. **ISO/IEC 15415**: es wird eine Verifizierung angelehnt an den ISO/IEC 15415 Standard durchgeführt. **ISO/IEC TR29158**: es wird eine Verifizierung angelehnt an den ISO/IEC TR29158 Standard durchgeführt.

Die beiden Verifizierungsmodi sind nur verfügbar, wenn gleichzeit eine **Pro** Methode bei **DPM Markierung** ausgewählt ist.

Text wenn nicht lesbar: das ist der Text, der ausgegeben wird, falls kein Code gelesen werden konnte.

Die maximale Zeit für die Dekodierung kann mit **Dekodier Timeout (ms)** in Millisekunden eingestellt werden (bis zu 60.000 Millisekunden). Der voreingestellte Wert von 0 bedeutet kein Timeout.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird der Barcode dekodiert und der **Code** angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).



Abb. 85: Das Bild-GUI des Werkzeugs **Barcode Lokalisieren**.

4.3.16 Kante Lokalisieren



Abb. 86: Das Icon des Werkzeugs **Kante Lokalisieren**.

Das Werkzeug “Kante Lokalisieren” sucht eine Kante in einer Szene.

Wenn das Werkzeug **Kante Lokalisieren** eine Kante lokalisieren kann, wird die Position und die Drehung der Kante bestimmt und an andere Werkzeuge weitergeleitet. Das Hauptergebnis des Werkzeugs **Kante Lokalisieren** ist eine **Pose**, die Position und Rotation beinhaltet.



Abb. 87: Die Kachel des Werkzeugs **Kante Lokalisieren**.

Das Werkzeug **Kante Lokalisieren** zeigt die **Pose** sowie die **Fläche** des Teils in Pixel an.

Werkzeuge in der Pipeline können diese Pose verwenden. Sie können mehrere Ortungswerkzeuge in einer Pipeline verwenden, und in einem beliebigen Werkzeug können Sie die Pose auswählen, indem Sie sie aus dem angezeigten Kontextmenü auswählen wenn Sie sie mit der Pin-Taste auswählen.

Parameter

Die **Suchrichtung** wird verwendet, um die Richtung der Scans auszuwählen. Sie können wählen **Rechts**, **Ab**, **Links** oder **Auf**.

Die **Polarität** wird verwendet, um **Dunkel -> Hell**, **Hell -> Dunkel** oder **Beide** auszuwählen. Dies gibt den Kantentyp in Kombination mit der **Suchrichtung** an.

Die **Kantenauswahl** wählt die **Erste**, **Letzte** oder **Best** Kante in Bezug auf die Scanrichtung.

Die Einstellung **Suchstrahlen** definiert die Abtastdichte der Suchstrahlen in Prozent. Bei 10 % wird jeder zehnte Pixel abgetastet, bei 50 % jeder zweite Pixel.

Glättung beeinflusst den Grad der Glättung, der auf die **Kantenstärke** angewendet wird, zwischen **Unscharf** und **Scharf**.

Die gewünschte **Kantenstärke** kann durch Überprüfen des Verlaufs entlang der Linie visuell bestimmt werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI.

Die ROI (Region of Interest) kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Die ROI muss möglichst senkrecht zu den zu messenden Kanten liegen, d.h. optimal unter 90°. Je näher zu 0° die Suchlinie liegt und je breiter der Suchbereich liegt, desto mehr werden Kanten durch Mittelung verschliffen und desto weniger genau werden die Ergebnisse.

Innerhalb der ROI werden die Scanlinien mit gepunkteten Linien dargestellt und aus den entlang der Scanlinien bestimmten Punkten wird eine Linie angepasst und angezeigt. Außerdem wird der Winkel der Linie angezeigt.



Abb. 88: Das Bild-GUI des Werkzeugs **Kante Lokalisieren**.

4.3.17 Form Lokalisieren



Abb. 89: Das Icon des Werkzeugs **Form Lokalisieren**.

Mit dem Werkzeug **Form Lokalisieren** können Sie mithilfe der Mustersuche ein Teil in einer Szene lokalisieren.

Wenn das Werkzeug **Form Lokalisieren** ein Teil lokalisieren kann, wird die Position und optional die Drehung des Teils bestimmt und an andere Werkzeuge weitergeleitet. Das Hauptergebnis des Werkzeugs **Form Lokalisieren** ist eine **Pose**, die Position und Rotation beinhaltet.



Abb. 90: Die Kachel des Werkzeugs **Form Lokalisieren**.

Das Werkzeug **Form Lokalisieren** zeigt die **Pose** sowie die **Fläche** des Teils in Pixel an.

Werkzeuge in der Pipeline können diese Pose verwenden. Sie können mehrere Ortungswerkzeuge in einer Pipeline verwenden, und in einem beliebigen Werkzeug können Sie die Pose auswählen, indem Sie sie aus dem angezeigten Kontextmenü auswählen wenn Sie sie mit der Pin-Taste auswählen.

Parameter

Unter **Methode** stehen drei verschiedene Verfahren zur Lokalisierung von Bildbereichen zur Auswahl.

Bei der Methode **Korrelation** stehen Flächen im Fordergrund. Das Verfahren ist anwendbar bei gleichmäßigem Hintergrund.

Die Methode **Geometrisch** hat einen Fokus auf Kanten, und ist verlässlicher bei ungleichmäßigem Hintergrund und bei partiellen Verdeckungen.

Sowohl **Geometrisch** als auch **Korrelation** sollten in der Rotation eingeschränkt werden, da die Speicheranforderungen erheblich sind.

Die Methode **Keypoint** sucht markante Bildpunkte in der Vorlage und vergleicht sie mit denen im Suchbild. Dadurch erkennt das Verfahren bis zu einem gewissen Grad unterschiedliche Skalierungen und ist nicht anfällig bei partiellen Verdeckungen.

Die **Methode** ist nur im Trainieren-Modus wählbar.

Im Handbuch im Kapitel **Mustererkennung** werden die verschiedenen Methoden genauer erklärt.

Zulässige Rotation ermöglicht es, die Drehung der Teile zu beschränken. Dieser Parameter ist nur im Trainieren-Modus sichtbar.

Die zum Zählen verwendete **Vorlage** kann durch Klicken auf die Schaltfläche **Vorlage trainieren** erstellt werden. Sobald Sie eine Vorlage trainiert haben wechselt das Werkzeug in den Suchmodus, in dem die ROI (Region of Interest) für den Suchbereich ausgewählt wird. Mit einem Klick auf **Neue Vorlage** wechselt das Werkzeug zurück in den Trainieren-Modus.

Der trainierten Vorlage kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden. Die ausmaskierten Pixel werden für das Matching nicht berücksichtigt.

Mit **Details reduzieren** werden während der Template-Vorverarbeitung Bilddetails reduziert, um die Ausführungszeit zu verkürzen.

Mit **Match-Qualität** (nur Korrelation und Geometrisch) können schrittweise Einstellungen zwischen **Präzise** und **Schnell** vorgenommen werden. Schnelleres Matching ist weniger genau und präzisere Übereinstimmung ist langsamer. Auf diese Weise können Sie den richtigen Kompromiss für Ihre Anwendung auswählen.

Ähnlich der Match-Qualität kann für die Keypoint Methode die **Genauigkeit** von **Zuverlässig** bis **Schnell** eingestellt werden. Je größer die trainierte Vorlage, desto zuverlässiger funktioniert auch ein schnellerer Modus.

Akzeptanzschwelle legt den Mindestschwellenwert für Übereinstimmungen in Prozent fest. Alle Übereinstimmungen unter diesem Schwellenwert werden ignoriert. Die Verfahren unterscheiden sich dabei was die zu erwartenden Schwellwerte anbelangt. Für **Korrelation** und **Keypoint** ist ein Schwellenwert von 50% ist in den meisten Fällen zu niedrig für gute Treffer. Bei **Geometrisch** hingegen wird selten eine Übereinstimmung von mehr als 50% erreicht, die Schwelle sollte hier also niedriger gewählt werden.

Vertrauensschwelle (nur Keypoint) legt fest, wie viele übereinstimmende Punkte zwischen Vorlage und Suchbild gefunden werden müssen. Je mehr Punkte desto höher ist das Vertrauen in den gefundenen Treffer.

Hintergrund entfernen (nur Korrelation und Geometrisch) schließt einheitliche Hintergrundbereiche von der Verarbeitung aus, wodurch die Verarbeitung beschleunigt werden kann. Falls der Hintergrund viele Details hat, sollten Sie diese Option nicht verwenden, da es die Verarbeitungszeit in solchen Fällen erhöhen kann.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Im Trainieren-Modus wird mit der ROI das Muster ausgewählt.



Abb. 91: Das Trainieren-GUI des Werkzeugs **Form Lokalisieren**.

Im Suchmodus wird die ROI verwendet, um den Suchbereich auszuwählen. Innerhalb des Suchbereichs wird das Muster gesucht und angezeigt.

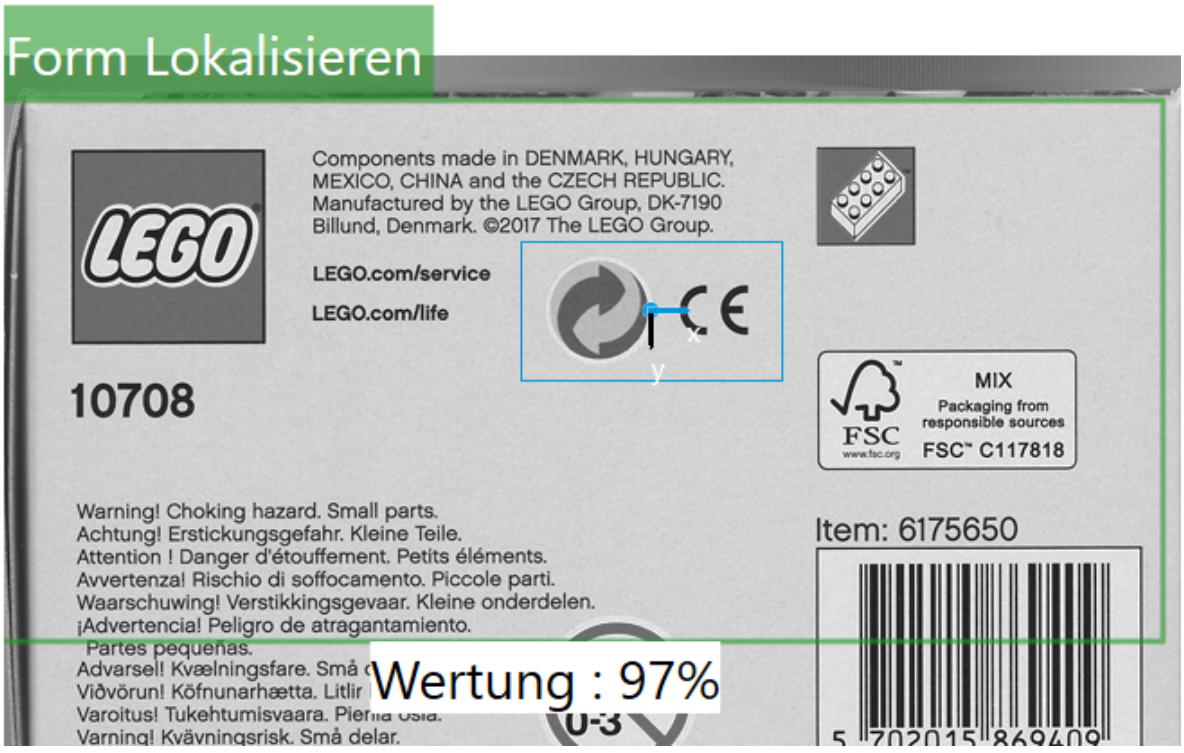


Abb. 92: Das Suchen-GUI des Werkzeugs **Form Lokalisieren**.

4.3.18 Text Lokalisieren



Abb. 93: Das Icon des Werkzeugs **Text Lokalisieren**.

Mit dem Werkzeug **Text Lokalisieren** wird ein Text in einer Szene lokalisiert.

Wenn das Werkzeug **Text Lokalisieren** einen Text lokalisieren kann, wird die Position und die Drehung des Texts bestimmt und an andere Werkzeuge weitergeleitet. Das Hauptergebnis des Werkzeugs **Text Lokalisieren** ist eine **Pose**, die Position und Rotation beinhaltet.



Abb. 94: Die Kachel des Werkzeugs **Text Lokalisieren**.

Das Werkzeug **Text Lokalisieren** zeigt die **Pose** des Texts in Pixel an.

Werkzeuge in der Pipeline können diese Pose verwenden. Sie können mehrere Ortungswerkzeuge in einer Pipeline verwenden, und in einem beliebigen Werkzeug können Sie die Pose auswählen, indem Sie sie aus dem angezeigten Kontextmenü auswählen wenn Sie sie mit der Pin-Taste auswählen.

Schriftart

Die Schriftart kann aus einer Liste gewählt werden.

Es gibt mehrere Gruppen von Schriften. Die Mitglieder der einzelnen Gruppen unterscheiden sich, da sie unterschiedliche Symbolsätze enthalten. Der Inhalt einer Schriftart wird durch ihren Namen beschrieben. Für die Namen der vortrainierten Schriften gilt die folgende Nomenklatur:

Der Name beginnt mit dem Gruppennamen, z. B. **Document** oder **DotPrint**, gefolgt von Indikatoren für den Symbolsatz die in der OCR-Schriftart enthalten sind. Die Bedeutung der Indikatoren ist die folgende:

- **0-9**: Die OCR-Schriftart enthält die Ziffern 0 bis 9.
- **A-Z**: Die OCR-Schriftart enthält die Großbuchstaben A bis Z.
- **+** : Die OCR-Schriftart enthält Sonderzeichen. Die Liste der Sonderzeichen variiert leicht zwischen den einzelnen OCR-Schriften. Sie ist im Folgenden für jede OCR-Schriftart einzeln aufgeführt.

Wenn der Name der OCR-Schriftart keines der oben genannten Kennzeichen enthält, enthält die OCR-Schriftart normalerweise die Ziffern 0 bis 9, die Großbuchstaben A bis Z, die Kleinbuchstaben Kleinbuchstaben a bis z und Sonderzeichen. Einige OCR-Schriften enthalten keine Kleinbuchstaben (z. B., DotPrint). Dies wird in der Beschreibung der jeweiligen Schriftart ausdrücklich erwähnt.

Um den Zeichensatz durch Verknüpfung einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Zeichensatz, wie er unter **Font** zu finden ist, entsprechen.

Document

Die Schriftart `Dokument` kann verwendet werden, um Zeichen zu lesen, die in Schriftarten wie Arial, Courier oder Times New Roman gedruckt wurden. Dies sind typische Schriftarten für den Druck von Dokumenten oder Briefen.

Beachten Sie, dass die Zeichen I und l der Schriftart Arial nicht unterschieden werden können. Das bedeutet, dass ein `l` fälschlicherweise für ein `I` gehalten werden kann und umgekehrt.

Verfügbare Sonderzeichen: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

Die OCR-Schriftart DotPrint kann verwendet werden, um mit Punktdruckern gedruckte Zeichen zu lesen. Sie enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . * :

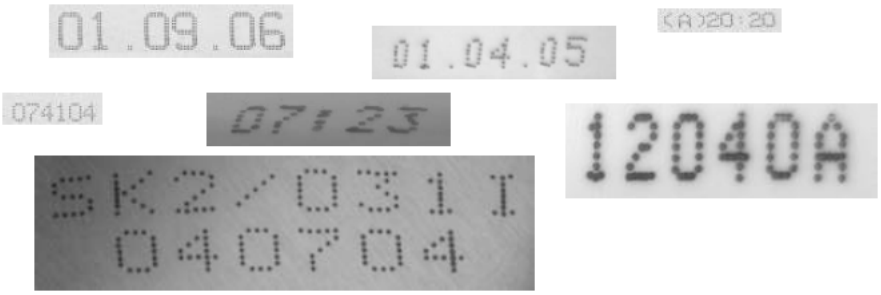


Abb. 95: Beispiele für gepunktete Schrift.

HandWritten

Die Schriftart HandWritten_0-9 kann zum Lesen handgeschriebener Zahlen verwendet werden.

Sie enthält die Ziffern 0-9.

Verfügbare Sonderzeichen: keine

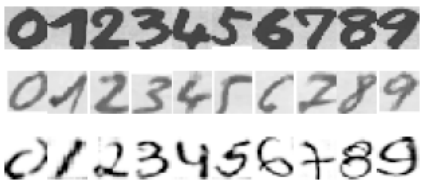


Abb. 96: Beispiele für handgeschriebene Ziffern.

Industrial

Die Schriftart Industrial kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B oder anderen serifenlosen Schriften Schriftarten gedruckt wurden. Diese Schriftarten werden normalerweise zum Drucken von z. B. Etiketten verwendet.

Verfügbare Sonderzeichen: - / + . $ % * e £ ¥

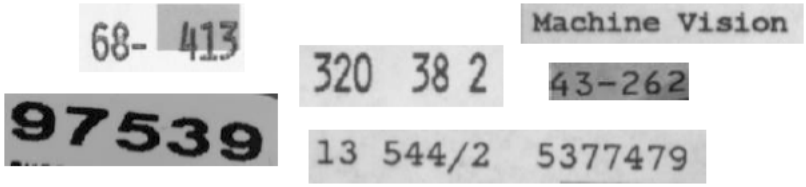


Abb. 97: Beispiele für industrielle Drucke.

OCR-A

Die Schrift OCR-A kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-A gedruckt sind.

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



Abb. 98: Ausgewählte Zeichen der OCR-A Schrift.

OCR-B

Die Schrift OCR-B kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-B gedruckt sind.

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



Abb. 99: Ausgewählte Zeichen der OCR-B Schrift.

Pharma

Die Schriftart Pharma kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B und anderen Schriftarten gedruckt sind, die typischerweise in der pharmazeutischen Industrie verwendet werden.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . ( ) :

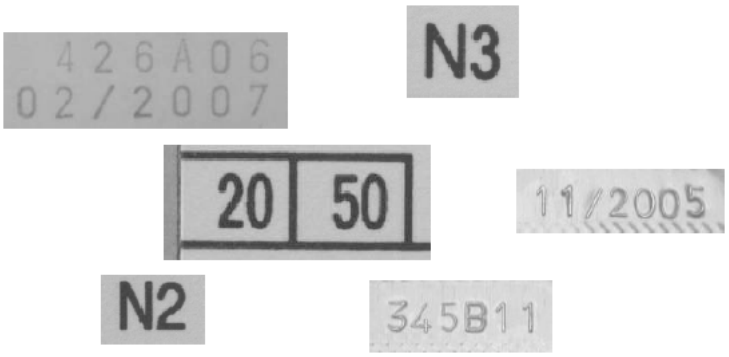


Abb. 100: Beispiele für pharmazeutische Drucke.

SEMI

Die Schriftart SEMI kann verwendet werden, um Zeichen zu lesen, die in der Schriftart SEMI gedruckt sind, die aus Zeichen besteht, die die so gestaltet sind, dass sie sich leicht voneinander unterscheiden lassen. Sie hat einen begrenzten Satz von Zeichen.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - .



Abb. 101: Zeichensatz der SEMI Schriftart.

Universal

Die OCR-Schriftart Universal kann zum Lesen einer Vielzahl unterschiedlicher Zeichen verwendet werden. Das Training dieser CNN-basierten Schriftart basiert auf den Zeichensätzen ‘Document’, ‘DotPrint’, ‘SEMI’ und ‘Industrial’.

Verfügbare Sonderzeichen: - / = + : < > . # $ % & ( ) @ * e £ ¥

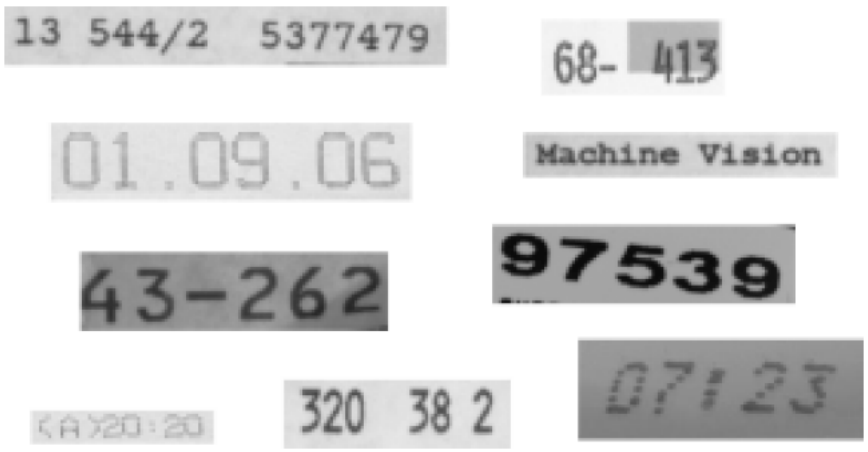


Abb. 102: Beispile für den Zeichensatz Universal.

Segmentierung

Min. Kontrast bestimmt den minimalen Kontrast zwischen Buchstaben und ihrem Hintergrund. Der Wert kann zwischen 1 und 255 eingestellt werden, der Standardwert ist 15.

Die **Polarität** kann auf **Dunkel auf Hell** für dunkle Zeichen auf hellem Hintergrund, **Hell auf Dunkel** für helle Zeichen auf dunklem Hintergrund oder auf **Beide** eingestellt werden, wenn beide Arten von Text segmentiert werden sollen.

Wenn **Rand eliminieren** angewählt ist, werden Regionen verworfen, die den Rand der ROI berühren.

Wenn **Fragmente hinzufügen** angewählt ist, werden Fragmente wie der Punkt auf einem “i” zu den segmentierten Buchstaben hinzugefügt. Das kann allerdings auch dazu führen, dass Störungen zu den segmentierten Buchstaben hinzugefügt werden.

Berührende Zeichen bestimmt die Trennung von Paaren oder kleinen Gruppen nebeneinander liegender Zeichen, die als eine einzige Region segmentiert werden. Wird **Standard** gewählt, werden solche Regionen erkannt und in zwei oder mehrere einzelne Zeichen aufgeteilt. Wird der Werte auf **falsch** gesetzt, wird keine Trennung von sich berührenden Zeichen durchgeführt.

Zeichengröße

Minimale Zeichenhöhe bestimmt die minimale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Maximale Zeichenhöhe bestimmt die maximale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Minimale Zeichenbreite bestimmt die minimale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Maximale Zeichenbreite bestimmt die maximale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Minimale Strichstärke bestimmt die minimale Strichstärke eines Buchstabens in Pixel. Soll die minimale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Minimale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Maximale Strichstärke bestimmt die maximale Strichstärke eines Buchstabens in Pixel. Soll die maximale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Maximale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Sonderzeichen

Wenn **Satzzeichen** angewählt ist, werden kleine Satzzeichen, die nahe an der Grundlinie der Textzeile liegen (wie Punkt oder Komma), zu den segmentierten Buchstaben hinzugefügt.

Wenn **Trennzeichen** angewählt ist, werden Trennzeichen wie ein Minus oder das Gleichheitszeichen zu den segmentierten Buchstaben hinzugefügt.

Punktdruck

Punktdruck muss angewählt werden, falls der zu segmentierende Text Punktdruck-Buchstaben enthält.

Enge Zeichen muss angewählt werden, wenn die Lücken zwischen benachbarten Buchstaben kleiner sind als die Lücke zwischen zwei Punkten innerhalb eines Buchstabens. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die genaue Größe der kleinsten Lücke zwischen benachbarten Buchstaben bekannt, kann diese stattdessen mittels **Minimale Zeichenlücke** gesetzt werden. In diesem Fall wird die Einstellung von **Enge Zeichen** ignoriert.

Minimale Zeichenlücke ist die Größe der kleinsten Lücke zwischen zwei Buchstaben in Pixel. Dieser Parameter kann in Fällen, in denen die Lücken zwischen Buchstaben kleiner sind als die zwischen Punkten innerhalb der Buchstaben, zu einer genaueren Segmentierung führen. Falls die minimale Größe nicht bekannt oder größer als die maximale Lücke zwischen den Punkten innerhalb eines Buchstaben ist, kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Sind Buchstaben sehr nah nebeneinander gedruckt, jedoch die Größe der minimalen Lücke unbekannt, kann stattdessen **Enge Zeichen** gesetzt werden.

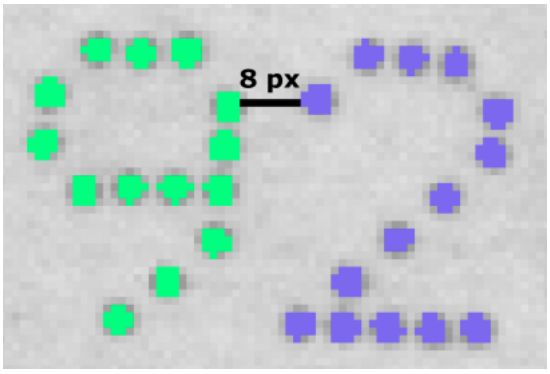


Abb. 103: Hier ist die kleinste Lücke zwischen zwei Buchstaben 8 Pixel.

Maximale Punktelücke ist die Größe der größten Lücke zwischen zwei Punkten innerhalb eines Buchstabens in Pixel. Um beliebige Punktdruck-Buchstaben zu segmentieren kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die größte Lücke zwischen zwei Punkten größer als die kleinste Lücke zwischen zwei Buchstaben, sollte zusätzlich **Enge Zeichen** angewählt werden oder **Minimale Zeichenlücke** entsprechend gesetzt werden. Wird **Maximale Punktelücke**gesetzt, kann die Performance der Texterkennung deutlich gesteigert werden.

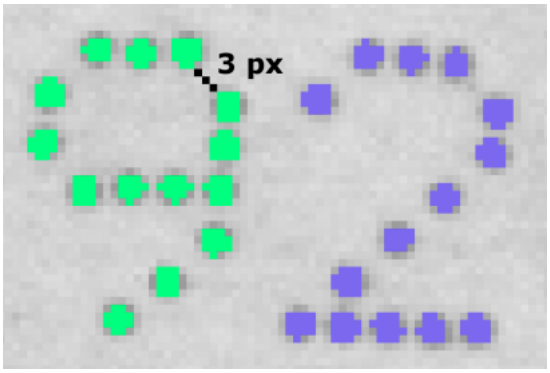


Abb. 104: Hier ist die größte Lücke zwischen zwei Punkten 3 Pixel.

Struktur

Zeilenstruktur legt die Struktur des gesuchten Textes fest. Für jede Zeile wird der Abstand zwischen aufeinanderfolgenden Buchstaben berechnet. Ausgehend von diesen Abständen wird die Zeile in einzelne Buchstabenblöcke aufgeteilt. Hierbei werden sehr kleine Zeichen wie z.B. ., _ und – vernachlässigt und als Leerzeichen interpretiert. Darüber hinaus können zusätzliche Trennzeichen angegeben werden, die ebenfalls vernachlässigt werden sollen, siehe **Zeilentrennzeichen**. Für die entstandenen Buchstabenblöcke wird dann geprüft ob sie einer der übergebenen Strukturen entsprechen.

Eine Struktur besteht aus einem String, der die Anzahl der Buchstaben/Ziffern pro Buchstabenblock enthält, welche durch Leerzeichen getrennt sind. Soll zum Beispiel ein Datum mit jeweils zwei Ziffern für Tag, Monat und Jahr gefunden werden, wäre die Struktur `2 2 2`. Kann das Jahr auch aus vier Ziffern bestehen, wäre die Struktur `2 2 2-4`, so dass der letzte Block aus zwei bis vier Buchstaben/Ziffern bestehen kann. Wenn die Struktur einem leeren Text entspricht, kann der zu findende Texte aus jeder beliebigen Struktur bestehen.

Jede gefundene Struktur wird als einzelne Zeile gespeichert. Soll stattdessen die komplette Zeile zurückgeliefert werden, die eine Struktur enthält, kann **Ganze Zeile** angewählt werden.

Zeilentrennzeichen ist ein String von Trennzeichen, die bei der Suche nach Strukturen vernachlässigt werden sollen, siehe **Zeilenstruktur**. Hierbei ist zu beachten, dass ein solches Trennzeichen immer auch im Klassifikator enthalten sein muss. Sollen z.B. die Zeichen `:` und `\` als Trennzeichen behandelt werden, kann der String `:\\` übergeben werden. Hierbei ist zu beachten, dass das vorangestellte `\` die Verwendung des Symbols `\` als Literal ermöglicht.

Ganze Zeile falls nur die gefundenen Strukturen als Zeilen zurückgegeben werden sollen, `false` falls jede Zeile, die eine passende Stuktur enthält komplett zurückgegeben werden soll.

Parameters

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Anzahl der Zeichen gezählt und mit farbigen grafischen Überlagerungen angezeigt. Zusätzlich wird die **Anzahl** angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).



Abb. 105: Das Bild-GUI des Werkzeugs **Text Lokalisieren**.

4.3.19 OCV



Abb. 106: Das Icon des **OCV** Werkzeugs

Das **OCV**-Werkzeug wird verwendet, um gedruckte Zeichen auf ihre Güte zu überprüfen.

Neben dem primären **Ok/Not Ok**-Ergebnis sind sekundäre Ergebnisse des **OCV**-Werkzeugs die **Wertung** und der erkannte **Text**.

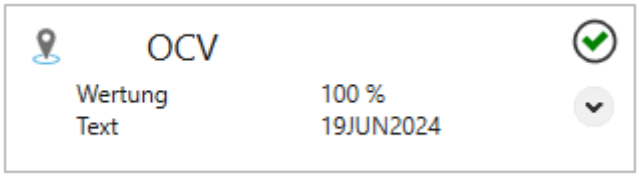


Abb. 107: Die Kachel des Werkzeugs **OCV**.

Das Werkzeug **OCV** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Modus

Das Werkzeug kann im **Trainieren**-Modus verwendet werden, um die Zeichen mit einem guten Bild zu trainieren, oder im **Prüfen**-Modus, um die Druckqualität der Zeichen anhand ihrer Profile und des Unterschieds zu den eingelernten Flächen zu überprüfen.

Im Modus **Trainieren** ist das Ergebnis des Werkzeugs immer **Falsch/Nicht Ok**. Im Modus **Prüfen** ist das Ergebnis **True/Ok**, wenn alle Zeichen innerhalb der vorgegebenen Grenzen übereinstimmen. Sobald ein einzelnes Zeichen nicht übereinstimmt, lautet das Ergebnis **Falsch/Nicht Ok**.

Akzeptanz

Die **Vertrauensschwelle** legt einen Score fest, über dem die Zeichen bei der Prüfung als gut betrachtet werden.

Maskieren

Sowohl beim Training als auch bei der Prüfung kann eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden. Die ausmaskierten Pixel werden für die Prüfung nicht berücksichtigt.

Schriftart

Die Schriftart kann aus einer Liste gewählt werden.

Es gibt mehrere Gruppen von Schriften. Die Mitglieder der einzelnen Gruppen unterscheiden sich, da sie unterschiedliche Symbolsätze enthalten. Der Inhalt einer Schriftart wird durch ihren Namen beschrieben. Für die Namen der vortrainierten Schriften gilt die folgende Nomenklatur:

Der Name beginnt mit dem Gruppennamen, z. B. **Document** oder **DotPrint**, gefolgt von Indikatoren für den Symbolsatz die in der OCR-Schriftart enthalten sind. Die Bedeutung der Indikatoren ist die folgende:

- **0-9**: Die OCR-Schriftart enthält die Ziffern 0 bis 9.
- **A-Z**: Die OCR-Schriftart enthält die Großbuchstaben A bis Z.
- **+** : Die OCR-Schriftart enthält Sonderzeichen. Die Liste der Sonderzeichen variiert leicht zwischen den einzelnen OCR-Schriften. Sie ist im Folgenden für jede OCR-Schriftart einzeln aufgeführt.

Wenn der Name der OCR-Schriftart keines der oben genannten Kennzeichen enthält, enthält die OCR-Schriftart normalerweise die Ziffern 0 bis 9, die Großbuchstaben A bis Z, die Kleinbuchstaben Kleinbuchstaben a bis z und Sonderzeichen. Einige OCR-Schriften enthalten keine Kleinbuchstaben (z. B., DotPrint). Dies wird in der Beschreibung der jeweiligen Schriftart ausdrücklich erwähnt.

Um den Zeichensatz durch Verknüpfung einzustellen muss dieser als Text übergeben werden. Der Text muss dabei exakt dem Zeichensatz, wie er unter **Font** zu finden ist, entsprechen.

`Document`

Die Schriftart `Dokument` kann verwendet werden, um Zeichen zu lesen, die in Schriftarten wie Arial, Courier oder Times New Roman gedruckt wurden. Dies sind typische Schriftarten für den Druck von Dokumenten oder Briefen.

Beachten Sie, dass die Zeichen I und l der Schriftart Arial nicht unterschieden werden können. Das bedeutet, dass ein `l` fälschlicherweise für ein `I` gehalten werden kann und umgekehrt.

Verfügbare Sonderzeichen: - = + < > . # $ % & ( ) @ * e £ ¥

DotPrint

Die OCR-Schriftart DotPrint kann verwendet werden, um mit Punktdruckern gedruckte Zeichen zu lesen. Sie enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . * :

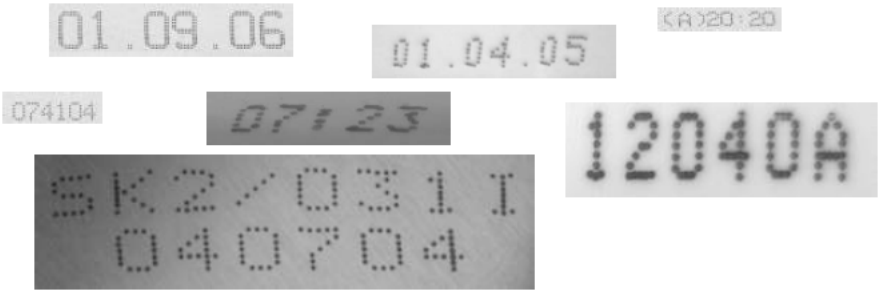


Abb. 108: Beispiele für gepunktete Schrift.

HandWritten

Die Schriftart HandWritten_0-9 kann zum Lesen handgeschriebener Zahlen verwendet werden.

Sie enthält die Ziffern 0-9.

Verfügbare Sonderzeichen: keine

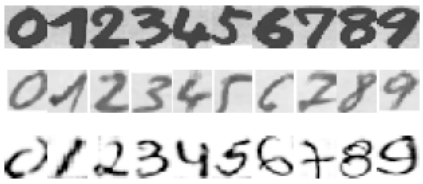


Abb. 109: Beispiele für handgeschriebene Ziffern.

Industrial

Die Schriftart Industrial kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B oder anderen serifenlosen Schriften Schriftarten gedruckt wurden. Diese Schriftarten werden normalerweise zum Drucken von z. B. Etiketten verwendet.

Verfügbare Sonderzeichen: - / + . $ % * e £ ¥

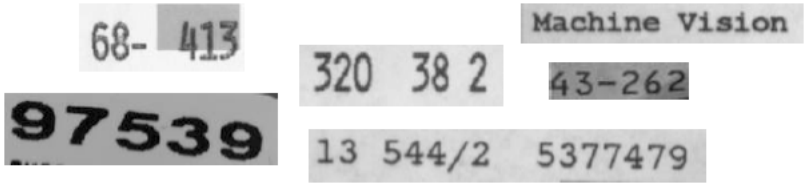


Abb. 110: Beispiele für industrielle Drucke.

OCR-A

Die Schrift OCR-A kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-A gedruckt sind.

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



Abb. 111: Ausgewählte Zeichen der OCR-A Schrift.

OCR-B

Die Schrift OCR-B kann zum Lesen von Zeichen verwendet werden, die in der Schriftart OCR-B gedruckt sind.

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



Abb. 112: Ausgewählte Zeichen der OCR-B Schrift.

Pharma

Die Schriftart Pharma kann zum Lesen von Zeichen verwendet werden, die in Schriftarten wie Arial, OCR-B und anderen Schriftarten gedruckt sind, die typischerweise in der pharmazeutischen Industrie verwendet werden.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - / . ( ) :

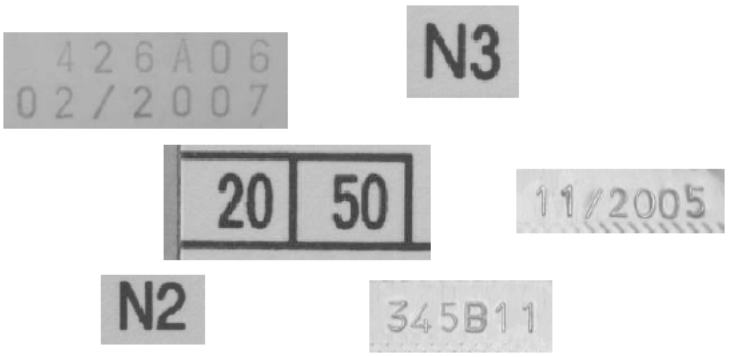


Abb. 113: Beispiele für pharmazeutische Drucke.

SEMI

Die Schriftart SEMI kann verwendet werden, um Zeichen zu lesen, die in der Schriftart SEMI gedruckt sind, die aus Zeichen besteht, die die so gestaltet sind, dass sie sich leicht voneinander unterscheiden lassen. Sie hat einen begrenzten Satz von Zeichen.

Diese OCR-Schriftart enthält keine Kleinbuchstaben.

Verfügbare Sonderzeichen: - .



Abb. 114: Zeichensatz der SEMI Schriftart.

Universal

Die OCR-Schriftart Universal kann zum Lesen einer Vielzahl unterschiedlicher Zeichen verwendet werden. Das Training dieser CNN-basierten Schriftart basiert auf den Zeichensätzen ‘Document’, ‘DotPrint’, ‘SEMI’ und ‘Industrial’.

Verfügbare Sonderzeichen: - / = + : < > . # $ % & ( ) @ * e £ ¥

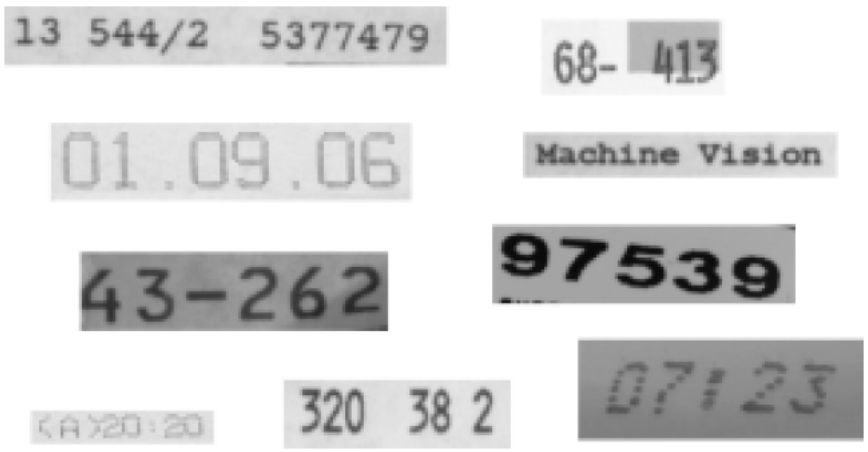


Abb. 115: Beispile für den Zeichensatz Universal.

Segmentierung

Min. Kontrast bestimmt den minimalen Kontrast zwischen Buchstaben und ihrem Hintergrund. Der Wert kann zwischen 1 und 255 eingestellt werden, der Standardwert ist 15.

Die **Polarität** kann auf **Dunkel auf Hell** für dunkle Zeichen auf hellem Hintergrund, **Hell auf Dunkel** für helle Zeichen auf dunklem Hintergrund oder auf **Beide** eingestellt werden, wenn beide Arten von Text segmentiert werden sollen.

Wenn **Rand eliminieren** angewählt ist, werden Regionen verworfen, die den Rand der ROI berühren.

Wenn **Fragmente hinzufügen** angewählt ist, werden Fragmente wie der Punkt auf einem “i” zu den segmentierten Buchstaben hinzugefügt. Das kann allerdings auch dazu führen, dass Störungen zu den segmentierten Buchstaben hinzugefügt werden.

Berührende Zeichen bestimmt die Trennung von Paaren oder kleinen Gruppen nebeneinander liegender Zeichen, die als eine einzige Region segmentiert werden. Wird **Standard** gewählt, werden solche Regionen erkannt und in zwei oder mehrere einzelne Zeichen aufgeteilt. Wird der Werte auf **falsch** gesetzt, wird keine Trennung von sich berührenden Zeichen durchgeführt.

Zeichengröße

Minimale Zeichenhöhe bestimmt die minimale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Maximale Zeichenhöhe bestimmt die maximale Höhe eines Buchstabens in Pixel. Um Buchstaben jeglicher Höhe zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenhöhe** schränkt die Höhe von Buchstaben ein, jedoch nicht die Höhe von Satz- oder Trennzeichen.

Minimale Zeichenbreite bestimmt die minimale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Minimale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Maximale Zeichenbreite bestimmt die maximale Breite eines Buchstabens in Pixel. Um Buchstaben jeglicher Breite zu segmentieren, kann der Wert 0 übergeben werden. **Maximale Zeichenbreite** schränkt die Breite von Buchstaben ein, jedoch nicht die Breite von Satz- oder Trennzeichen.

Minimale Strichstärke bestimmt die minimale Strichstärke eines Buchstabens in Pixel. Soll die minimale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Minimale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Maximale Strichstärke bestimmt die maximale Strichstärke eines Buchstabens in Pixel. Soll die maximale Strichstärke während der Textsegmentierung automatisch geschätzt werden, kann der Wert 0 übergeben werden. **Maximale Strichstärke** schränkt die Strichstärke von Buchstaben ein, jedoch nicht die Strichstärke von Satz- oder Trennzeichen.

Sonderzeichen

Wenn **Satzzeichen** angewählt ist, werden kleine Satzzeichen, die nahe an der Grundlinie der Textzeile liegen (wie Punkt oder Komma), zu den segmentierten Buchstaben hinzugefügt.

Wenn **Trennzeichen** angewählt ist, werden Trennzeichen wie ein Minus oder das Gleichheitszeichen zu den segmentierten Buchstaben hinzugefügt.

Punktdruck

Punktdruck muss angewählt werden, falls der zu segmentierende Text Punktdruck-Buchstaben enthält.

Enge Zeichen muss angewählt werden, wenn die Lücken zwischen benachbarten Buchstaben kleiner sind als die Lücke zwischen zwei Punkten innerhalb eines Buchstabens. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die genaue Größe der kleinsten Lücke zwischen benachbarten Buchstaben bekannt, kann diese stattdessen mittels **Minimale Zeichenlücke** gesetzt werden. In diesem Fall wird die Einstellung von **Enge Zeichen** ignoriert.

Minimale Zeichenlücke ist die Größe der kleinsten Lücke zwischen zwei Buchstaben in Pixel. Dieser Parameter kann in Fällen, in denen die Lücken zwischen Buchstaben kleiner sind als die zwischen Punkten innerhalb der Buchstaben, zu einer genaueren Segmentierung führen. Falls die minimale Größe nicht bekannt oder größer als die maximale Lücke zwischen den Punkten innerhalb eines Buchstaben ist, kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Sind Buchstaben sehr nah nebeneinander gedruckt, jedoch die Größe der minimalen Lücke unbekannt, kann stattdessen **Enge Zeichen** gesetzt werden.

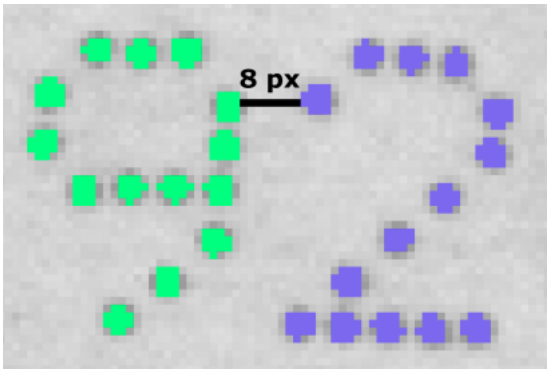


Abb. 116: Hier ist die kleinste Lücke zwischen zwei Buchstaben 8 Pixel.

Maximale Punktelücke ist die Größe der größten Lücke zwischen zwei Punkten innerhalb eines Buchstabens in Pixel. Um beliebige Punktdruck-Buchstaben zu segmentieren kann der Wert 0 übergeben werden. Falls **Punktdruck** nicht angewählt ist, hat dieser Parameter keine Auswirkung. Ist die größte Lücke zwischen zwei Punkten größer als die kleinste Lücke zwischen zwei Buchstaben, sollte zusätzlich **Enge Zeichen** angewählt werden oder **Minimale Zeichenlücke** entsprechend gesetzt werden. Wird **Maximale Punktelücke**gesetzt, kann die Performance der Texterkennung deutlich gesteigert werden.

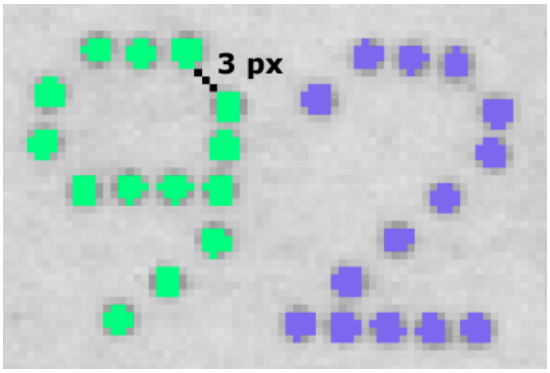


Abb. 117: Hier ist die größte Lücke zwischen zwei Punkten 3 Pixel.

Struktur

Zeilenstruktur legt die Struktur des gesuchten Textes fest. Für jede Zeile wird der Abstand zwischen aufeinanderfolgenden Buchstaben berechnet. Ausgehend von diesen Abständen wird die Zeile in einzelne Buchstabenblöcke aufgeteilt. Hierbei werden sehr kleine Zeichen wie z.B. ., _ und – vernachlässigt und als Leerzeichen interpretiert. Darüber hinaus können zusätzliche Trennzeichen angegeben werden, die ebenfalls vernachlässigt werden sollen, siehe **Zeilentrennzeichen**. Für die entstandenen Buchstabenblöcke wird dann geprüft ob sie einer der übergebenen Strukturen entsprechen.

Eine Struktur besteht aus einem String, der die Anzahl der Buchstaben/Ziffern pro Buchstabenblock enthält, welche durch Leerzeichen getrennt sind. Soll zum Beispiel ein Datum mit jeweils zwei Ziffern für Tag, Monat und Jahr gefunden werden, wäre die Struktur `2 2 2`. Kann das Jahr auch aus vier Ziffern bestehen, wäre die Struktur `2 2 2-4`, so dass der letzte Block aus zwei bis vier Buchstaben/Ziffern bestehen kann. Wenn die Struktur einem leeren Text entspricht, kann der zu findende Texte aus jeder beliebigen Struktur bestehen.

Jede gefundene Struktur wird als einzelne Zeile gespeichert. Soll stattdessen die komplette Zeile zurückgeliefert werden, die eine Struktur enthält, kann **Ganze Zeile** angewählt werden.

Zeilentrennzeichen ist ein String von Trennzeichen, die bei der Suche nach Strukturen vernachlässigt werden sollen, siehe **Zeilenstruktur**. Hierbei ist zu beachten, dass ein solches Trennzeichen immer auch im Klassifikator enthalten sein muss. Sollen z.B. die Zeichen `:` und `\` als Trennzeichen behandelt werden, kann der String `:\\` übergeben werden. Hierbei ist zu beachten, dass das vorangestellte `\` die Verwendung des Symbols `\` als Literal ermöglicht.

Ganze Zeile falls nur die gefundenen Strukturen als Zeilen zurückgegeben werden sollen, `false` falls jede Zeile, die eine passende Stuktur enthält komplett zurückgegeben werden soll.

Parameters

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild mit einem bestimmten Overlay angezeigt.

Im **Trainieren**-Modus wird der erkannte Text hervorgehoben. Außerdem werden die Zeichen umrandet.



Abb. 118: Das Werkzeug **OCV** im **Trainieren** Modus

Im **Prüfen** Modus werden die Zeichen angezeigt und entsprechend ihrem Score grün bzw. rot eingefärbt..



Abb. 119: Das Werkzeug **OCV** im **Prüfen** Modus

4.3.20 Winkel Messen



Abb. 120: Das Icon des Werkzeugs **Winkel Messen**.

Das Werkzeug **Winkel Messen** wird verwendet, um den Winkel einer Kante in einer Szene zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Winkel Messen**-Werkzeugs der **Winkel** einer Kante.



Abb. 121: Die Kachel des Werkzeugs **Winkel Messen**.

Das Werkzeug **Winkel Messen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Suchrichtung** wird verwendet, um die Richtung der Scans auszuwählen. Sie können wählen **Rechts**, **Ab**, **Links** oder **Auf**.

Die **Polarität** wird verwendet, um **Dunkel -> Hell**, **Hell -> Dunkel** oder **Beide** auszuwählen. Dies gibt den Kantentyp in Kombination mit der ** Scanrichtung ** an.

Die **Kantenauswahl** wählt die **Erste**, **Letzte** oder **Best** Kante in Bezug auf die Scanrichtung.

Die Einstellung **Suchstrahlen** definiert die Abtastdichte der Suchstrahlen in Prozent. Bei 10 % wird jeder zehnte Pixel abgetastet, bei 50 % jeder zweite Pixel.

Glättung beeinflusst den Grad der Glättung, der auf die **Kantenstärke** angewendet wird, zwischen **Unscharf** und **Scharf**.

Die gewünschte **Kantenstärke** kann durch Überprüfen des Verlaufs entlang der Linie visuell bestimmt werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Messung

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn der Winkel innerhalb der **Toleranz (+/-)** in Grad in Bezug auf **Erwarteter Winkel** liegt, sonst ist das Ergebnis **Falsch/Nicht Ok**.

Zusätzlich ist das Werkzeugergebnis nur dann **Wahr/Ok**, wenn der in **Vollständigkeit** angegebene Prozentwert erreicht wird. Dieser definiert, welcher Anteil der Suchstrahlen auf der detektierten Linie liegen muss.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI werden die Suchrichtung mit gepunkteten Linien dargestellt, die gefundenen Punkte werden mit grünen Punkten markiert und die gefundene Linie wird angezeigt. Die Winkeltoleranz wird mit einem “Schmetterling”s-Bereich angezeigt.

Die ROI (Region of Interest) kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Die ROI muss möglichst senkrecht zu den zu messenden Kanten liegen, d.h. optimal unter 90°. Je näher zu 0° die Suchlinie liegt und je breiter der Suchbereich liegt, desto mehr werden Kanten durch Mittelung verschliffen und desto weniger genau werden die Ergebnisse.



Abb. 122: Das Bild-GUI des Werkzeugs **Winkel Messen**.

Zusätzlich visualisiert ein “Schmetterling” die Winkeltoleranz.

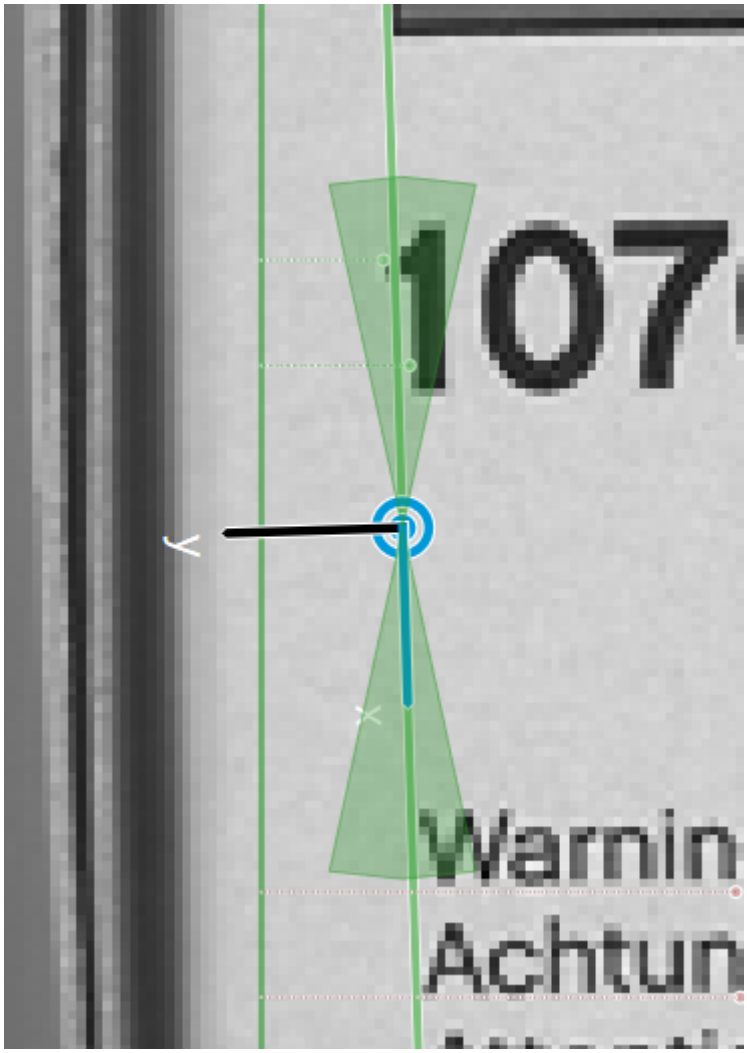


Abb. 123: Der “Schmetterling” des Werkzeugs **Winkel Messen**.

4.3.21 Helligkeit Erkennen



Abb. 124: Das Icon des Werkzeugs **Helligkeit Erkennen**.

Das Werkzeug **Helligkeit Erkennen** wird verwendet, um die mittlere Helligkeit in einer ROI (region of interest) oder mehreren ROIs zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis sind sekundäre Ergebnisse des **Helligkeit Erkennen**-Werkzeugs die mittlere Helligkeit **Mittel**, sowie die minimale Helligkeit **Minimum** und die maximale Helligkeit **Maximum**..



Abb. 125: Die Kachel des Werkzeugs **Helligkeit Erkennen**.

Das Werkzeug **Helligkeit Erkennen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Messung** kann zwischen **Einzeln** und **Mehrfach** umgestellt werden, um entweder eine einzelne ROI oder mehrere ROIs gleichzeitig zu messen.

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken).

Ein **Rechteck** kann auch gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Beim **Polygon** können Ecken durch Klicken auf eine Randlinie hinzugefügt werden, bzw. durch Drücken der Delete Taste entfernt werden.

Im **Einzeln** Modus kann einer ROI eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden.

Die mittlere Helligkeit wird im **Einzeln** Modus innerhalb einer ROI gemessen, bzw. im **Mehrfach** Modus in allen ROIs.

Mit den Werten **min. Mittel**, **max. Mittel**, **min. Helligkeit** bzw. **max. Helligkeit** kann das primäre Ergebnis des Werkzeugs beeinflusst werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn **Mittel** zwischen **min. Mittel** und **max. Mittel** liegt, sowie **min. Helligkeit** kleiner/gleich dem gemessenen Minimum und **max. Helligkeit** größer/gleich dem gemessenen Maximum ist, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Sekundäre Ergebnisse sind der gemessene Mittelwert sowie die gemessenen Minimum und Maximum Werte. Im **Mehrfach** Modus werden diese Werte über alle ROIs gesammt ermittelt, sowie für jede einzelne ROI noch ein einzelnen Mittelwert.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit der ROI oder den ROIs. Im **Einzeln** Modus wird die Helligkeit innerhalb der ROI angezeigt. Im **Mehrfach** Modus werden die sequentiellen Nummern der ROIs angezeigt.



Abb. 126: Das Bild-GUI des Werkzeugs **Helligkeit Erkennen**.

4.3.22 Kreis Messen



Abb. 127: Das Icon des Werkzeugs **Kreis Messen**.

Das Werkzeug **Kreis Messen** wird verwendet, um ein kreisförmiges Teil in einer Szene zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis sind sekundäre Ergebniss des **Kreis Messen**-Werkzeugs der **Durchmesser** und der **Mitte**lpunkt des Kreises.

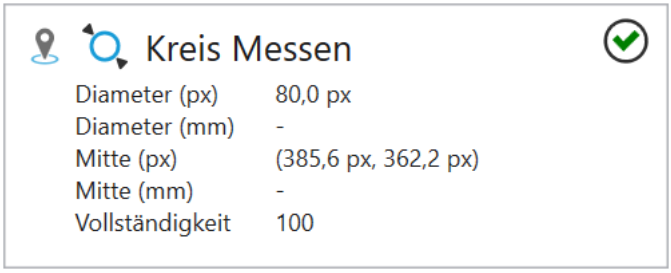


Abb. 128: Die Kachel des Werkzeugs **Kreis Messen**.

Das Werkzeug **Kreis Messen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Suchrichtung** wird verwendet, um die Richtung der Scans auszuwählen. Sie können **Außen** oder **Innen** wählen.

Die **Polarität** wird verwendet, um **Dunkel -> Hell**, **Hell -> Dunkel** oder **Beide** auszuwählen. Dies gibt den Kantentyp in Kombination mit der ** Scanrichtung ** an.

Die **Kantenauswahl** wählt die **Erste**, **Letzte** oder **Best** Kante in Bezug auf die Scanrichtung.

Suchstrahlen wählt die Anzahl der radialen Suchstrahlen aus, die zum Finden des Kreises verwendet werden. Eine Zahl von 36 würde alle 10 Grad einen Suchstrahl positionieren.

Glättung beeinflusst den Grad der Glättung, der auf die **Kantenstärke** angewendet wird, zwischen **Unscharf** und **Scharf**.

Die gewünschte **Kantenstärke** kann durch Überprüfen des Verlaufs entlang der Linie visuell bestimmt werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Kreis

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn der **Durchmesser** zwischen **Min. Durchmesser** und **Max. Durchmesser** liegt, und wenn die **Min. Vollständigkeit** übertroffen wird, sonst ist das Ergebnis **Falsch/Nicht Ok**. Die Vollständigkeit kann nur dann richtig berechnet werden, wenn der Mittelpunkt der Region of Interest möglichst nahe am Kreismittelpunkt liegt.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer ROI. Innerhalb der ROI werden die gefundenen Punkte werden mit grünen oder roten (Ausreißer) Punkten markiert und der angepasste Kreis wird angezeigt.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien).



Abb. 129: Das Bild-GUI des Werkzeugs **Kreis Messen**.

4.3.23 Kontrast Erkennen



Abb. 130: Das Icon des Werkzeugs **Kontrast Erkennen**.

Das Werkzeug **Kontrast Erkennen** wird verwendet, um den Kontrast in einer ROI (region of interest) zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Kontrast Erkennen**-Werkzeugs der **Kontrast**, normiert auf 100 %.



Abb. 131: Die Kachel des Werkzeugs **Kontrast Erkennen**.

Das Werkzeug **Kontrast Erkennen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Messung** kann zwischen **Einzeln** und **Mehrfach** umgestellt werden, um entweder eine einzelne ROI oder mehrere ROIs gleichzeitig zu messen.

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die **Form der ROI** kann als **Rechteck**, **Kreis**, **Ring** oder **Polygon** gewählt werden. Abhängig von der Auswahl wird die Form der interaktiv auf der rechten Seite eingestellten ROI angepasst.

Die Region of Interest kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken).

Ein **Rechteck** kann auch gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Beim **Polygon** können Ecken durch Klicken auf eine Randlinie hinzugefügt werden, bzw. durch Drücken der Delete Taste entfernt werden.

Im **Einzeln** Modus kann einer ROI eine **Maske** hinzugefügt werden. Die Maske kann in einem interaktiven Editor pixel-genau editiert werden.

Der Kontrast wird im **Einzeln** Modus innerhalb einer ROI gemessen, bzw. im **Mehrfach** Modus in allen ROIs.

Mit den Werten **Minimum**, **Maximum** kann das primäre Ergebnis des Werkzeugs beeinflusst werden.

Mit dem Parameter **Empfindlichkeit** kann eine bestimmte Anzahl an dunklesten und hellsten Pixel von der Berechnung des Kontrastes ausgeschlossen werden. Je geringer die Empfindlichkeit, desto mehr dunkle und mehr helle Pixel werden ignoriert.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur

Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn **Kontrast** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit der ROI oder den ROIs. Im **Einzeln** Modus wird der Kontrast innerhalb der ROI angezeigt. Im **Mehrfach** Modus werden die sequentiellen Nummern der ROIs angezeigt.

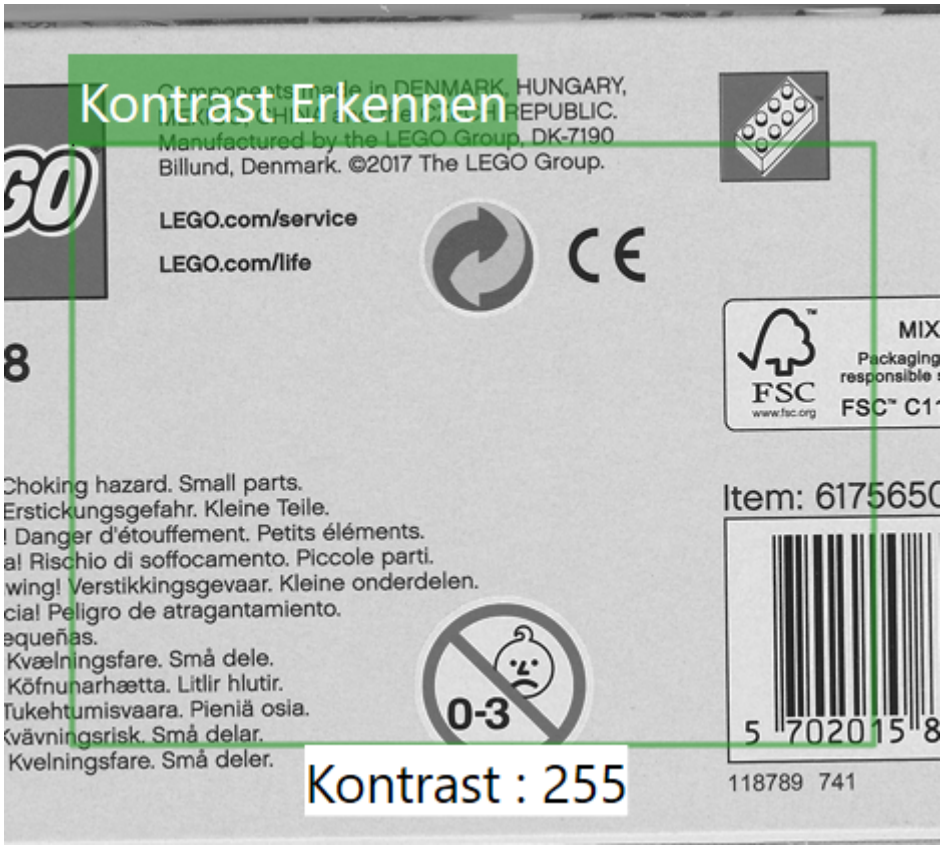


Abb. 132: Das Bild-GUI des Werkzeugs **Kontrast Erkennen**.

4.3.24 Distanz Messen



Abb. 133: Das Icon des Werkzeugs **Distanz Messen**.

Das Werkzeug **Distanz Messen** wird verwendet, um den Abstand zweier Kanten in einer Szene zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Distanz Messen**-Werkzeugs der **Abstand** zwischen zwei Kanten.



Abb. 134: Die Kachel des Werkzeugs **Distanz Messen**.

Das Werkzeug **Distanz Messen** kann die Pose von jedem Lokalisierungswerkzeug in der Pipeline verwenden. Wählen Sie dazu mit der Pin-Taste ein bestimmtes Lokalisierungswerkzeug.

Parameter

Die **Suchrichtung** wird verwendet, um die Richtung der Scans auszuwählen. Sie können wählen **Rechts**, **Ab**, **Links** oder **Auf**.

Die **Polarität** wird verwendet, um **Dunkel -> Hell**, **Hell -> Dunkel** oder **Beide** auszuwählen. Dies gibt den Kantentyp in Kombination mit der ** Scanrichtung ** an.

Glättung beeinflusst den Grad der Glättung, der auf die **Kantenstärke** angewendet wird, zwischen **Scharf** und **Unscharf**.

Die gewünschte **Kantenstärke** kann durch Überprüfen des Verlaufs entlang der Linie visuell bestimmt werden.

Bildvorverarbeitung kann verwendet werden, um eine optionale Vorverarbeitungsoperation im Werkzeug auszuwählen. **Trenne Objekte (Open)** führt eine morphologische Öffnung durch, **Verbinde Objekte (Close)** führt eine morphologische Schließung durch, **Verkleinere Objekte (Erode)** führt eine Erosion durch, **Vergrößere Objekte (Dilate)** führt eine morphologische Dilatation durch. Bei den morphologischen Operationen ist zu beachten, dass sie einen Bezug zur Helligkeit der Objekte haben: **Trenne Objekte (Open)** gilt für helle Objekte, auf dunkle Objekte angewandt ist der Effekt gegensätzlich, d.h. er verbindet dunkle Objekte. So wie **Trenne Objekte (Open)** und **Verbinde Objekte (Close)** sind auch **Verkleinere Objekte (Erode)** und **Vergrößere Objekte (Dilate)** komplementär. Der **Gradientfilter (Sobel)** verstärkt die Kanten und der **Rauschfilter (Lowpass)** reduziert das Rauschen.

Wenn Kalibrierungsdaten für den Sensor gespeichert wurden, können diese Daten für eine Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) herangezogen werden. Mit **Kalibrierung anwenden** wird die Konvertierung von Bildkoordinaten (px) in Weltkoordinaten (mm) eingeschaltet.

Diese Kontrollelemente werden nur angezeigt, wenn Kalibrierungsdaten gespeichert wurden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Distanz

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn der **Abstand** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**. Der Vergleich kann je nach Wahl der **Einheit** in Pixeln (px) oder Millimetern (mm) erfolgen, wobei letzterer nur möglich ist, wenn der Sensor kalibriert wurde.

Bildanzeige mit Overlay

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. In der ROI werden die Messlinie und der Abstand in Pixeln angezeigt.

Die ROI (Region of Interest) kann verschoben werden (durch Ziehen im Inneren) oder in der Größe geändert werden (durch Ziehen der Randlinien oder Ecken) oder gedreht werden (durch Ziehen des Drehpunktes in der Mitte der oberen Kante).

Die ROI muss möglichst senkrecht zu den zu messenden Kanten liegen, d.h. optimal unter 90°. Je näher zu 0° die Suchlinie liegt und je breiter der Suchbereich liegt, desto mehr werden Kanten durch Mittelung verschliffen und desto weniger genau werden die Ergebnisse.



Abb. 135: Das Bild-GUI des Werkzeugs **Distanz Messen**.

4.3.25 Punkt zu Linie Distanz Messen



Abb. 136: Das Icon des Werkzeugs **Punkt zu Linie Distanz Messen**.

Das Werkzeug **Punkt zu Linie Distanz Messen** wird verwendet, um die Distanz zwischen einem Punkt und einer Linie zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Punkt zu Linie Distanz Messen**-Werkzeugs die **Distanz**.



Abb. 137: Die Kachel des Werkzeugs **Punkt zu Linie Distanz Messen**.

Das Werkzeug **Punkt zu Linie Distanz Messen** verwendet einen Punkt und eine Linie von Werkzeugen, die Punkte oder Linien messen (d. h. **Kreis Messen**, **Kante Lokalisieren**, etc.).

Punkt bzw. Linie können durch Klicken auf die jeweiligen Schaltflächen ausgewählt werden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Distance

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn **Distanz** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Abstandslinie und die **Distanz** angezeit.



Abb. 138: Das Bild-GUI des Werkzeugs **Punkt zu Linie Distanz Messen**.

4.3.26 Punkt zu Punkt Distanz Messen



Abb. 139: Das Icon des Werkzeugs **Punkt zu Punkt Distanz Messen**.

Das Werkzeug **Punkt zu Punkt Distanz Messen** wird verwendet, um die Distanz zwischen zwei Punkten zu messen.

Neben dem primären **Ok/Not Ok**-Ergebnis ist ein sekundäres Ergebnis des **Punkt zu Punkt Distanz Messen**-Werkzeugs die **Distanz**.



Abb. 140: Die Kachel des Werkzeugs **Punkt zu Punkt Distanz Messen**.

Das Werkzeug **Punkt zu Punkt Distanz Messen** verwendet zwei Punkte von Werkzeugen, die Punkte messen (d. h. **Kreis Messen**, **Kante Lokalisieren**, etc.).

Die beiden Punkte können durch Klicken auf die jeweiligen Schaltflächen ausgewählt werden.

Werkzeug aktiv schaltet das komplette Werkzeug aktiv oder inaktiv.

Distance

Das primäre Werkzeugergebnis ist **Wahr/Ok**, wenn **Distanz** zwischen **Minimum** und **Maximum** liegt, andernfalls ist das Ergebnis **Falsch/Nicht Ok**.

Auf der rechten Seite wird das Bild angezeigt, überlagert mit einer rechteckigen ROI. Innerhalb der ROI wird die Abstandslinie und die **Distanz** angezeit.

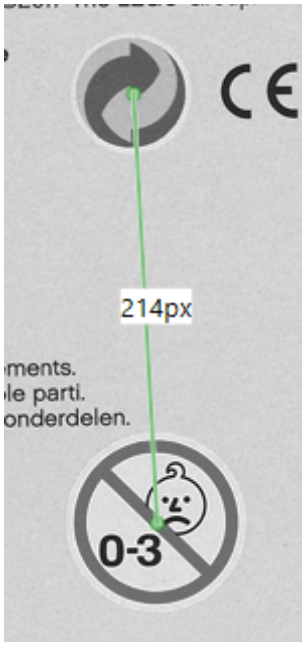


Abb. 141: Das Bild-GUI des Werkzeugs **Punkt zu Punkt Distanz Messen**.

4.3.27 Logik - Resultate



Abb. 142: Das Icon des Werkzeugs **Logik**.

Das Werkzeug **Logik** wird verwendet, um die Logik zu definieren und eine Ausgabe aus den Ergebnissen verschiedener Werkzeuge abzuleiten.

Das Werkzeug **Logik** ist besonders: bei jeder Inspektion ist **Logik** der letzte Schritt eine Verarbeitungspipeline.

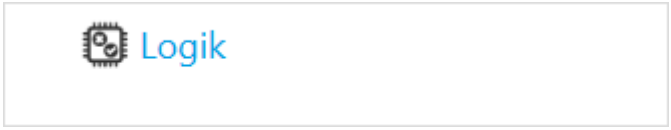


Abb. 143: Die Kachel des Werkzeugs **Logik**.

Mit dem Werkzeug **Logik** werden Prüfergebnisse durch ein Funktionsblockdiagramm erstellt.

Resultate

Standardmäßig ist das Gesamt-**Resultat** des Logik-Werkzeugs eine Kombination der Ergebnisse aller Werkzeuge in der Pipeline. Nur wenn alle Werkzeuge **IO** (oder logisch **1**) lieferen, ist das Ergebnis des **Logik**-Werkzeugs selbst - das heißt das Ergebnis der Gesamtprüfung - **IO**.

Sie können dies jedoch übersteuern, indem Sie Ihre eigene Logik auf der Registerkarte **Resultate** implementieren. Wählen Sie ein Werkzeug im Menü **Werkzeuge** aus, um es dem Funktionsplanbereich hinzuzufügen. Sie können dann das vordefinierte **Resultat** des Werkzeugs verwenden. Hier ist ein Beispiel:

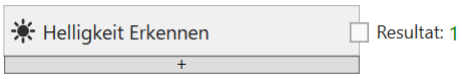


Abb. 144: Ein Werkzeug Knoten.

Alternativ können Sie einen Vergleichsausdruck für jedes Ergebnis eines sekundären Werkzeugs erstellen:

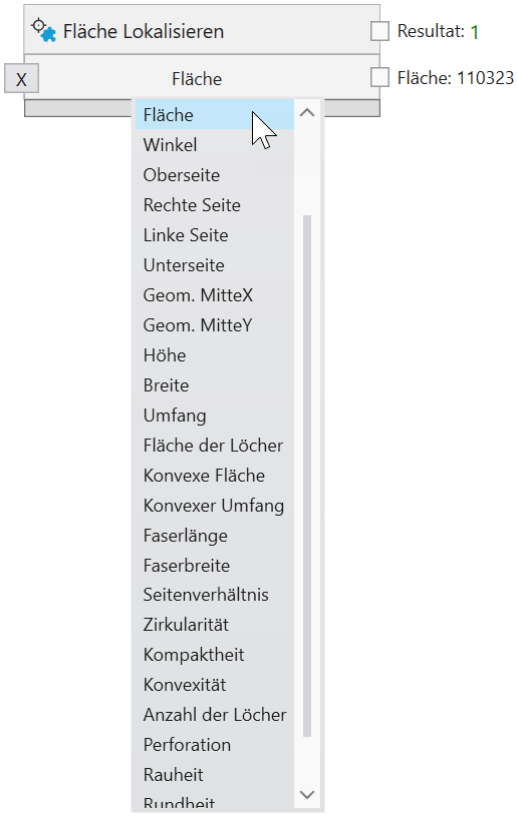


Abb. 145: Zusätzliche logische Ausgänge mit Sekundärresultaten erzeugen.

Jedes dieser Ergebnisse verschiedener Werkzeuge kann mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die Sie im Menü **Logische Operatoren** auswählen können:

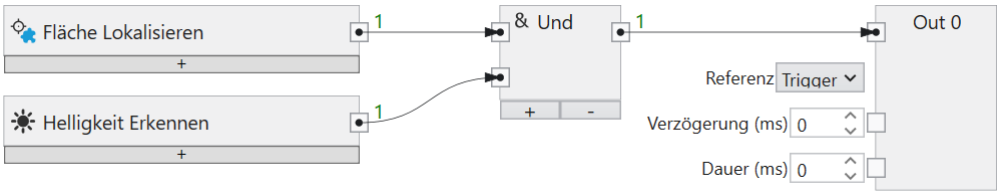


Abb. 146: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Schließlich kann das Ergebnis der logischen Kombination mit dem **Resultat**-Knoten verbunden werden und wird dann als Gesamtergebnis der Prüfung verwendet.

4.3.28 Logik - Digitale Ausgänge

Mit dem Werkzeug **Logik** werden Prüfergebnisse durch ein Funktionsblockdiagramm erstellt. Funktionsblockdiagramme sind in der IEC EN 61131-3 genormt.

Digitale Ausgänge

Standardmäßig werden die vier digitalen Ausgänge des **CS-6x/ID-6x0** als Knoten auf der Zeichenfläche angezeigt.

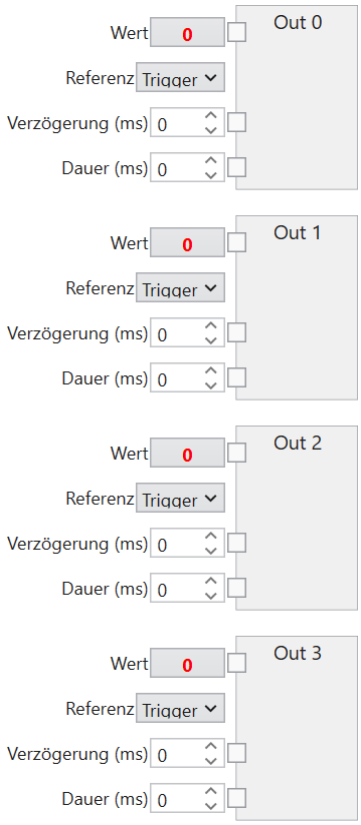


Abb. 147: Die digitalen Ausgangsknoten des **Logic** Werkzeugs.

Die **Verzögerung** und die **Dauer** eines digitalen Ausgangssignals können angegeben werden (in

Millisekunden). Die Verzögerung wird ab der steigenden Flanke des Triggers (**Reference: Trigger**) oder ab dem Ende der Inspektion (**Reference: EndOfInspection**) gemessen, eine Dauer von 0 bedeutet unendliche Dauer.

Werkzeuge können entweder durch eine Auswahl im Kontextmenü links oder mit der Maus aus der Werkzeugliste gezogen werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die noch logisch kombiniert werden können, um die gewünschten Ausgaben zu erstellen. Hier ist ein Beispiel:

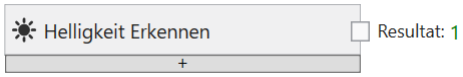


Abb. 148: Ein Werkzeug Knoten.

Das primäre Ergebnis des Werkzeugs ist **True** oder **False** und kann mit jedem der vier Ausgänge verbunden werden (durch Ziehen vom kleinen quadratischen Ausgangs-Pin mit der Bezeichnung **Resultat** des Werkzeugs auf den entsprechenden Pin) - auf den Pin des Ausgangs mit dem Namen **Wert**.



Abb. 149: Ein Werkzeug mit einem Ausgang verbinden.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

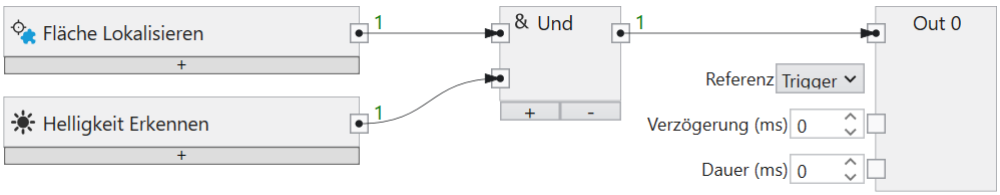


Abb. 150: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Ein Knoten auf der Leinwand oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

Ein Klick auf die kleine Schaltfläche **+** am unteren Rand eines Werkzeugknotens fügt dem Werkzeugknoten eine zusätzliche Zeile hinzu, in der Sie einen logischen Ausdruck für jedes sekundäre Werkzeugergebnis erstellen können.

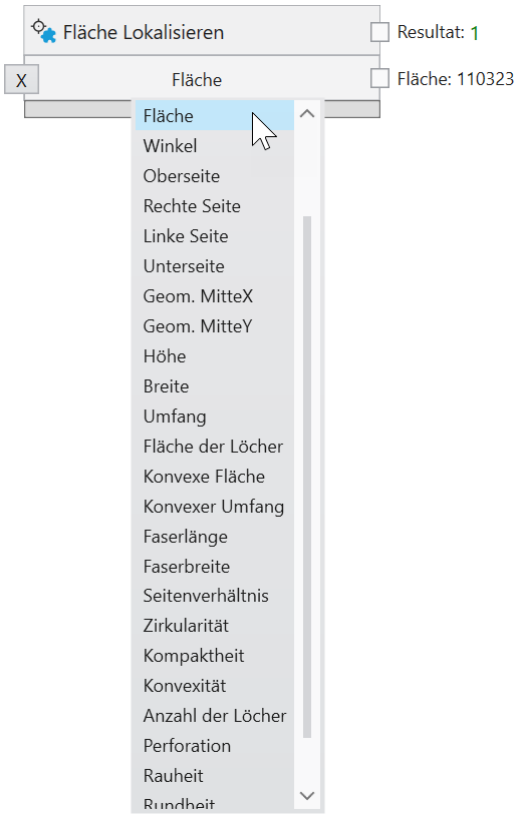


Abb. 151: Zusätzliche logische Ausgänge mit Sekundärresultaten erzeugen.

Geben Sie minimale oder maximale Grenzen ein und erstellen Sie zusätzliche Logikergebnisse aus den Ergebnissen des sekundären numerischen Resultats.

Die Taste **X** löscht eine Zeile, wenn sie nicht mehr benötigt wird.

4.3.29 Logik - Feldbus Ausgänge

Mit dem Werkzeug **Logik** werden Prüfergebnisse durch ein Funktionsblockdiagramm erstellt.

Inspektionsergebnisse können über PROFINET an eine SPS übertragen werden, durch Verknüfen mit PROFINET Ausgängen.

Das Verknüpfen eines Werts erfolgt in mehreren Schritten auf den Registerkarten **PROFINET Ausgänge** oder **EtherNet/IP Ausgänge** des Werkzeugs **Logik**: 1. Ein Werkzeug-Knoten wird mit einem Schreib-Knoten verbunden. 2. Der Schreib-Knoten wird mit einem PROFINET oder EtherNet/IP Modul in einem Slot verbunden.

Ausgangsdaten werden am Ende eines Inspektionszyklus übertragen.

Verknüpfen von Feldbus Ausgängen

Hinweis: Die Screenshots zeigen PROFINET, aber die Verfahren für EtherNet/IP sind genau dieselben, es sei denn, es ist ausdrücklich vermerkt.

Feldbus-Ausgänge können definiert werden, indem Sie im linken Menü **Write Boolean**, **Write Byte**, **Write Int32**, **Write Float** oder **Write String** auswählen oder mit der rechten Maustaste in die Zeichenfläche klicken um eine Auswahl zu treffen.

Werkzeuge können aus der Werkzeugliste gewählt werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die noch logisch kombiniert werden können, um die gewünschten Ausgaben zu erstellen. Hier ist ein Beispiel:

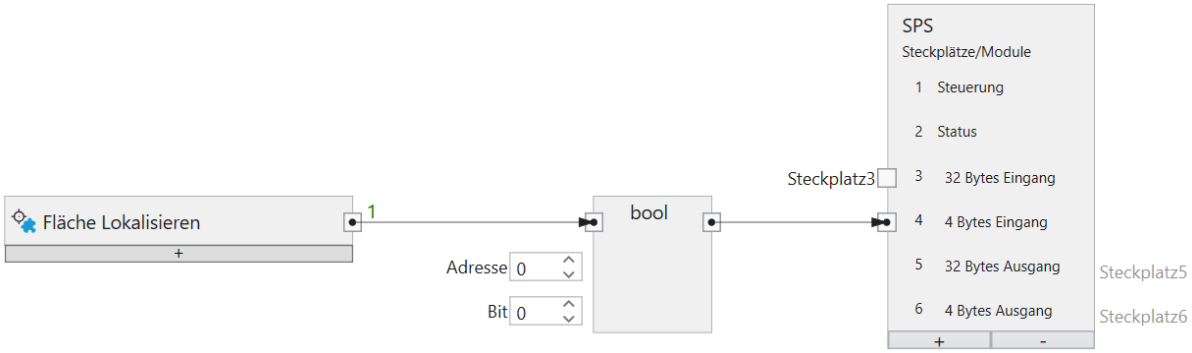


Abb. 152: Ein Werkzeug Knoten für PROFINET.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

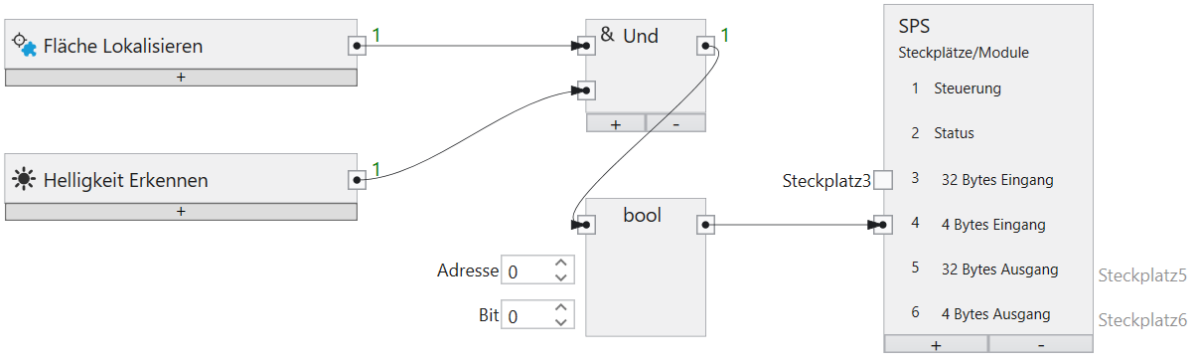


Abb. 153: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Ein Knoten oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

PROFINET oder EtherNet/IP kann neben logischen Werten auch numerische Werte oder Texte übertragen.

Mit den PROFINET oder EtherNet/IP Ausgangsknoten (**Write Boolean**, **Write Byte**, **Write Int32**, **Write Float** oder **Write String**) können sie bestimmen, an welche Adresse innerhalb eines Moduls die Werte geschrieben werden.



Abb. 154: Ein boolescher Ausgang.

Die Größe eines booleschen Ausgangs beträgt 1 Byte (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Bit gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).



Abb. 155: Ein Byte Ausgang.

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

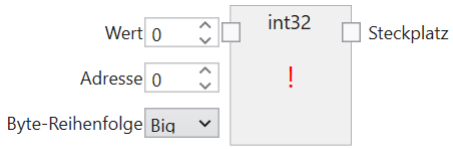


Abb. 156: Ein ganzzahliger Ausgang.

Die Größe eines ganzzahligen Ausgangs beträgt 4 Byte.

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

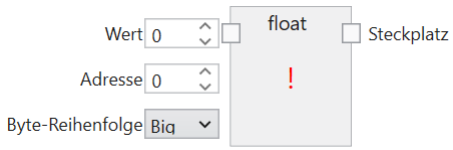


Abb. 157: Ein Fließkomma Ausgang.

Die Größe eines Fließkomma Ausgangs beträgt 4 Byte.

Wert ist der Wert, der übertragen werden soll.

Adresse gibt die Adresse im Modul an.

Byte-Reihenfolge Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

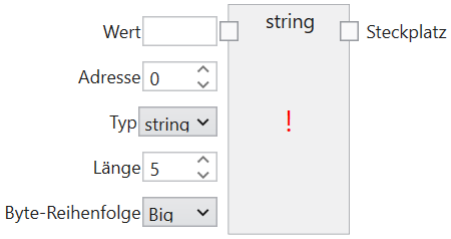


Abb. 158: Ein Text Ausgang.

Profinet: Die Größe eines Texts beträgt (2 + n) * s, wobei n die maximale Anzahl von Zeichen ist und s = 1 (für einen Ansi STRING) oder s = 2 (für einen Unicode-WSTRING). Das erste Byte/Wort (Ansi/Unicode) enthält die maximale Anzahl von Zeichen, welche die Zeichenfolge enthalten kann. Das zweite Byte/Wort (Ansi/Unicode) enthält die tatsächliche Anzahl der Zeichen, welche die Zeichenfolge enthält.

EtherNet/IP: Die Größe eines Texts beträgt (4 + n), wobei n die maximale Anzahl von Zeichen ist. Die ersten 4 Byte werden als ganze Zahl interpretiert und stellen die tatsächliche Anzahl der Zeichen des Strings da.

Wert ist der Wert, der übertragen werden soll.

Typ gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist. Bei EtherNet/IP entfällt dieser Input, da EtherNet/IP nur Ansi Texte unterstützt.

Adresse gibt die Adresse im Modul an.

Länge gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Byte-Reihenfolge Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden. Bei EtherNet/IP entfällt dieser Input, da er nur für Unicode Texte relevant ist.

Steckplatz verbindet den Knoten mit einem Eingangs Modul (**CS-6x/ID-6x0** -> SPS).

Der **SPS** Knoten wird verwendet, um Module in bestimmten Steckplätzen zu definieren. Dieses Setup muss exakt mit dem Setup auf der SPS übereinstimmen, um die Kommunikation zwischen der SPS und dem **CS-6x/ID-6x0** zu ermöglichen. Der SPS Knoten wird sowohl im Akquisition als auch im Logik Werkzeug angezeigt. Innerhalb des SPS Knotens werden die Namen Eingang und Ausgang für die Module wie von der SPS aus gesehen verwendet. Ein SPS Ausgang wird von der

SPS zum **CS-6x/ID-6x0** übertragen, wo er als Eingang angesehen wird. Ein SPS-Eingang wird vom **CS-6x/ID-6x0** zur SPS übertragen, auf dem **CS-6x/ID-6x0** wird dies als Ausgang gesehen.

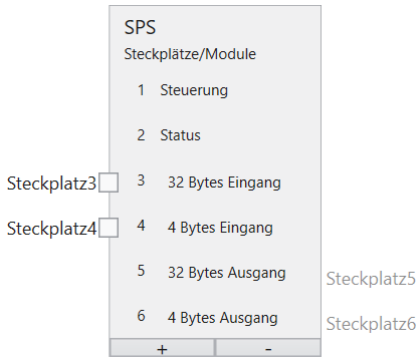


Abb. 159: Der SPS Knoten mit sechs Modulen.

Beiespielhaft sind hier sechs Module ausgewählt.

Unter der Überschrift **Steckplätze** werden die Module grafisch angezeigt. Außerdem werden die verwendeten Regionen innerhalb der Module angezeigt.

4.3.30 Logik - TCP/IP Kommando Ausgänge

Mit dem Werkzeug **Logik** werden Prüfergebnisse durch ein Funktionsblockdiagramm erstellt. Funktionsblockdiagramme sind in der IEC EN 61131-3 genormt.

Inspektionsergebnisse können über TCP/IP Kommandos an eine Gegenstelle übertragen werden, durch Verknüfen mit TCP/IP Variablen Ausgängen.

Das Verknüpfen eines Werts erfolgt auf der Registerkarte **TCP/IP Variablen Ausgänge** des Werkzeugs **Logik**: Ein Werkzeug-Knoten wird mit einer Variablen im Outputs Knoten verbunden. Ausgangsdaten werden am Ende eines Inspektionszyklus übertragen.

Verknüpfen von TCP/IP Variablen Ausgängen

Werkzeuge können aus der Werkzeugliste gewählt werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die mit dem TCP/IP Outputs Knoten verbunden werden können. Hier ist ein Beispiel:

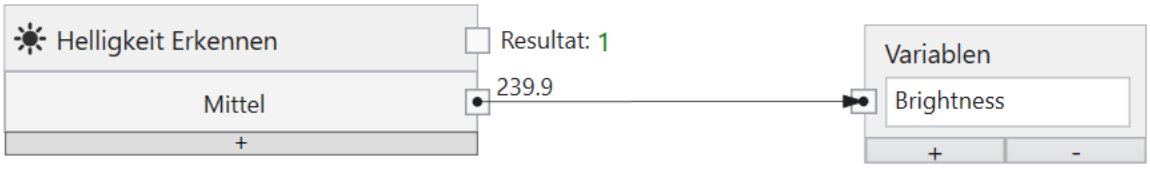


Abb. 160: Ein Werkzeug Knoten für TCP/IP Variablen Ausgänge.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

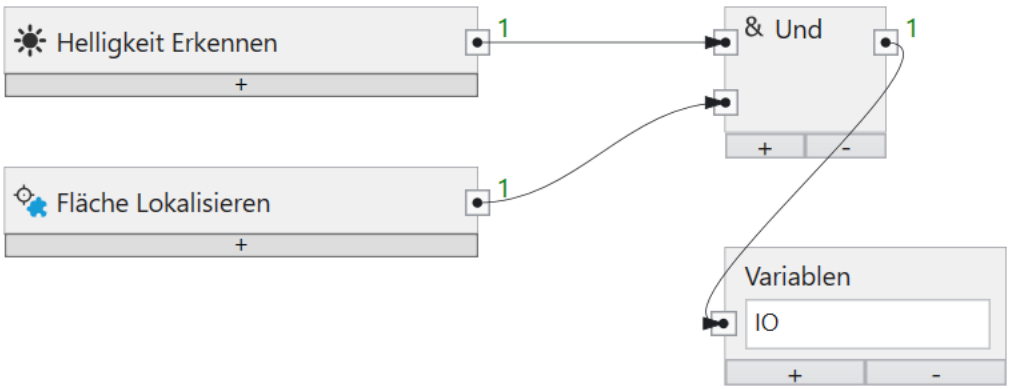


Abb. 161: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

Ein Knoten oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

TCP/IP Kommando Variablen können neben logischen Werten auch numerische Werte oder Texte übertragen.

4.3.31 Logik - FTP

Mit dem Werkzeug **Logik** werden Prüfergebnisse durch ein Funktionsblockdiagramm erstellt. Funktionsblockdiagramme sind in der IEC EN 61131-3 genormt.

FTP/SFTP

Der **CS-6x/ID-6x0** Vision-Sensor kann optional Bilder über FTP (File Transfer Protocol) oder SFTP (Secure File Transfer Protocol) auf einem Server speichern. Die FTP-Option muss aktiviert sein (siehe **Protokoll Einstellungen** des **CS-6x/ID-6x0**) und die Adresse und der Port des FTP-Servers müssen definiert sein.

Das FTP bzw. SFTP Protokoll wird gewählt, indem dem Host ein `ftp://` bzw. `sftp://` vorangestellt wird.

Außerdem müssen die Benutzeranmeldeinformationen (Benutzername und Kennwort) angegeben werden, damit eine Verbindung zum FTP/SFTP-Server hergestellt werden kann. Die Kommunikation zwischen dem eingebauten optionalen FTP/SFTP-Client erfolgt bei FTP standardmäßig über Port `21`, bei SFTP standardmäßig über Port `22`, den Standardports für FTP bzw. SFTP.

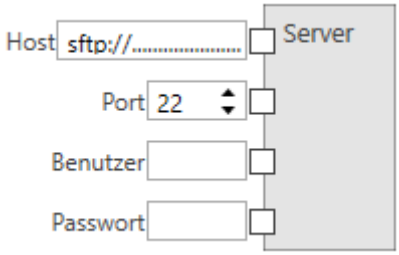


Abb. 162: Die FTP/SFTP Server Einstellungen.

Wenn FTP aktiviert ist, kann ein Job Bilder auf einen FTP/SFTP-Server hochladen. Die Adresse und die Anmeldeinformationen des FTP/SFTP-Servers müssen im Job auf der Registerkarte FTP des Logik-Werkzeugs konfiguriert werden.

Hochladen von Bildern

Bilder von Inspektionen können in zwei Verzeichnissen gespeichert werden: **IO Ordner** wird für Inspektionen verwendet, die in Ordnung waren, **NIO Ordner** wird für fehlgeschlagene Inspektionen verwendet. Das Verzeichnis und/oder der Dateiname (**IO Dateiname** und **NIO Dateiname**) des **Bilder Hochladen** Knotens können als feste Werte eingetragen oder durch einfügen des **Text** Knotens (siehe Beschreibung zu **Text Knoten**) variabel gehalten werden. Wenn Sie den **Text** Knoten verwenden möchten, verbinden Sie die Ausgabe des Knotens **Text** mit den Eingaben **IO Ordner**, **NIO Ordner**, **IO Dateiname** oder **NIO Dateiname** des **Bild Hochladen**-Knotens.

Wenn der FTP/SFTP-Server nicht zum Hochladen verfügbar ist, werden die Bilder auf dem **CS-6x/ID-6x0** zwischengespeichert, bis die maximale Anzahl von Bildern erreicht wurde. Wenn die maximale Anzahl erreicht wurde, werden zuerst ältere IO-Bilder und dann ältere NIO-Bilder überschrieben, sodass diese Bilder für den FTP/SFTP-Upload für immer verloren gehen. Dieses Verhalten gibt NIO-Bildern Vorrang.

Alle Bilder, die sich im FTP-Cache befinden und aus irgendeinem Grund nicht hochgeladen werden konnten, können mit nVision-i auf dem PC gespeichert werden.

Hochladen von Text

Neben der Option Bilder über FTP zu übertragen besteht die Option auch Text an eine Datei anzuhängen (bzw. diese neu zu erstellen, sollte sie noch nicht existieren). Hierzu kann im Menü links unter **Hochladen** der Knoten **Text** ausgewählt werden. Dieser kann genau wie der **Bild Hochladen** Knoten (siehe **Hochladen von Bildern**) konfiguriert werden. Zusätzlich verfügt der **Text Hochladen** Knoten über den Eingang **Text** indem beliebige Texte an die Angegebene Datei angehängt werden können. Hierzu kann der unter **Text Knoten** beschriebene Knoten verwendet werden. Der gewünschte Text-Ausgang muss mit dem im vorgesehenen Eingang im **Text hochladen** Knoten verbunden werden.

Text Knoten

Fügen Sie dem Funktionsblockdiagramm einen **Text**-Knoten hinzu und tippen Sie einen Namen oder wählen Sie einen Eintrag aus der Dropdown-Liste, um zwischen **Akquisition-ID**, **Job Name**, **Job Nummer**, **SensorName**, **Sensor Seriennummer**, **Jahr**, **Monat**, **Tag**, **Stunde**, **Minute**, **Sekunde** und **Millisekunde** zu wählen. Zur Formatierung stehen die Optionen **Carriage return**, **Line feed**, **Carriage return & Line feed**, sowie **Tab** zur Verfügung. Wenn Sie einen der vordefinierten Einträge entfernen möchten, wählen Sie das erste Element **Trennen** aus der Dropdown-Liste. Sie können auch Resultate von Werkzeugen mit dem **Text** Knoten verbinden indem Sie einen entsprechenden Ausgang eines Werkzeugs mit einem der Eingänge des **Text** Knotens verbinden. Sie können beliebig viele Eingänge zum **Text** Knoten durch das drücken des “+” Buttons hinzufügen oder den letzten Eingang durch drücken des “-” Buttons entfernen.

4.3.32 Smart Tool



Abb. 163: Das Icon des Werkzeugs **Smart Tool**.

Mit dem Werkzeug **Smart Tool** können Berechnungsschritte in eine Pipeline eingefügt werden.

Neben reinen Berechnungs-Schritten und Visualisierungs-Schritten kann mit dem Smart Tool auch ein Nachführungs-Schritt erstellt werden.

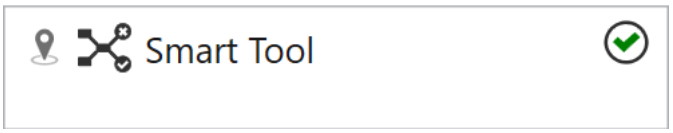


Abb. 164: Die Kachel des Werkzeugs **Smart Tool**.

Die Anzeige ist aufgeteilt in **Bild** und **Editor**.

Bild

Im **Bild** wird das Bild inkl. einer eventuellen Visualisierung angezeigt.

Editor

im **Editor** wird die Logik für den Schritt definiert.

4.4 Grafische Programmierung

4.4.1 Grafische Programmierung

Manche Aspekte in der Erstellung von Jobs werden beim **CS-6x/ID-6x0** Vision-Sensor mittels grafischer Programmierung durchgeführt. Beispiele sind die Verknüpfung der logischen Digitaleingänge oder über Feldbusse bzw. TCP/IP übermittelte Eingangswerte im Akquisitons-Werkzeug. Auch im Logik-Werkzeug wird grafische Programmierung verwendet, um Ausgangswerte für die digitalen Ausgänge oder für PROFINET und TCP/IP oder FTP zu berechnen.

Pipelines

Pipelines sind die Programme aus denen nVision-i **Jobs** bestehen. Daten fließen durch Pipelines, ähnlich wie Wasser durch Rohre fließt.

Knoten sind die Bausteine von Pipelines. Knoten haben **Pins**, mit denen Knoten mittels **Verbindungen** verbunden werden.

Hier ist ein Beispiel für eine Pipeline, die aus zwei Knoten besteht:



Abb. 165: Eine Pipeline aus zwei Knoten.

Eine Pipeline stellt einen gerichteten Graphen dar. Daten fließen in der Richtung, die mit den Pfeilen am Ende der Verbindungen angezeigt werden. Zyklen in den Graphen sind verboten und werden automatisch aufgebrochen.

Pipelines werden live ausgeführt. Bei jeder Wertänderung wird die Pipeline erneut ausgeführt und erzeugt neue Ausgaben. Knoten werden nur dann ausgeführt, wenn die Werte benötigt werden, die sie produzieren.

Knoten

Knoten sind die Grundbausteine in Pipelines.

Ein Knoten kann **Eingangs-Pins** und **Ausgangs-Pins** besitzen, wobei nicht jeder Knoten beides hat. Die Pins werden verwendet, um Knoten mittels **Verbindungen** zu verbinden.

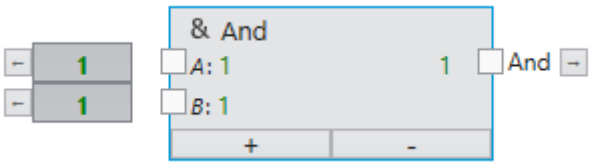


Abb. 166: Ein Knoten mit Eingangs- und Ausgangs-Pins.

Knoten können vollständig (wie in der Abbildung oben) oder kompakt dargestellt werden.

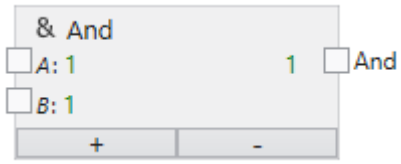


Abb. 167: Ein Knoten in kompakter Darstellung.

In den Editor Einstellungen kann zwischen beiden Darstellungen gewechselt werden.

Es gibt auch Knoten, bei denen Eingänge vorhanden sind, die keinen entsprechenden Pin haben. Dementsprechend kann keine Verbindung gezogen werden, aber mit den verfügbaren Editoren können bestimmte Werte eingestellt werden.

Pins

Pins sind typisiert. Verbindungen sind nur möglich zwischen Pins mit kompatiblen **Typen**, und zwar jeweils ein Ausgangs-Pin mit einem oder mehreren Eingangs-Pins.

Pins sind kompatibel, wenn ihre **Typen** gleich sind, oder wenn der auf der Verbindung fließende Typ automatisch konvertiert werden kann.

Pins mit inkompatiblen Typen können nicht verbunden werden.

Eingangs Pins mit bestimmten Typen haben typ-spezifische Editoren, mit denen ein Wert bestimmt werden kann. Editoren sind beispielsweise verfügbar für bool'sche Eingänge, für numerische Eingänge und für Text-Eingänge. Links neben einem Eingangs-Pin befindet sich eine Taste mit einem Pfeil-Symbol. Mit dieser Taste kann ein Menü geöffnet werden, das nur die Knoten enthält, die zum jeweiligen Eingangs-Pin kompatible Ausgangs-Pins besitzen. Es ist weiterhin möglich, mit Drag'n'Drop aus einem Eingangs-Pin heraus eine Verbindung zu ziehen, womit beim Drop ebenfalls dieses Menü mit kompatiblen Knoten/Pins geöffnet wird.

Ausgangs Pins haben rechts neben ihrem Namen ebenfalls eine Taste mit einem kleinen Pfeil-Symbol. Damit kann ein Menü geöffnet werden, das nur die Knoten enthält, die zum jeweiligen Ausgangs-Pin kompatible Eingangs-Pins besitzen. Dieses Menü öffnet sich auch, wenn aus dem Ausgangs-Pin heraus eine Verbindung gezogen wird.

Falls am Ausgangs-Pin komplexe Typen mit Properties anliegen (z.B. hat eine Zeichenkette auch eine Länge), wird links neben dem Pin ein nach unten zeigender Pfeil angezeigt. Damit können die Properties des Typs angewählt werden, die bei Anwahl einen weiteren Ausgangs-Pin anzeigen mit der gewählten Property.

Typen

Die in Pipelines fließenden Daten sind typisiert. Hier sind ein paar Beispiele für Typen:

Typ	Beschreibung
`Boolean`	Ein bool'scher Wert der die beiden binären Zustände `True` (`1`) und `False` (`0`) annehmen kann.
`Byte`	Ein ganzzahliger Wert zwischen 0 und 255 (inklusive).
`Integer`	Ein ganzzahliger Wert zwischen -2147483648 und 2147483647 (inklusive).
`Single`	Ein Fließkomma-Wert zwischen -3.40282347 x 10^38 und 3.40282347 x 10^38.
`Double`	Ein Fließkomma-Wert zwischen -1.7976931348623157 x 10^308 und 1.7976931348623157 x 10^308.
`String`	Eine Zeichenkette, z.B. `Text`.

Konvertierung

Die Verbindungen zwischen Pins stellen die Wege für den Datenfluß dar. Die Daten können nur fließen, wenn die Typen der Pins gleich sind, oder wenn sie automatisch konvertiert werden können.

Erweiternde Konvertierungen können automatisch durchgeführt werden. Die Konvertierung eines Int32 Werts in einen Float Wert is erweiternd und wird automatisch ausgeführt.

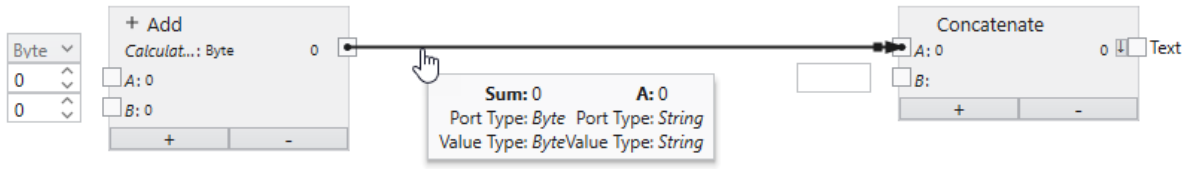


Abb. 168: Beispiel für konvertierbare Typen.

Ein numerischer Wert kann automatisch in einen Text konvertiert werden. Die Konvertierung wird durch einen kleinen Punkt am Ende des Verbindungspfeils visualisiert.

Einengende Konvertierungen können nicht automatisch durchgeführt werden. Die Konvertierung eines Float Werts in einen Int32 Wert würde die Nachkommastellen abschneiden und wird deshalb nicht automatisch ausgeführt.



Abb. 169: Beispiel für inkompatible Typen.

Ein Text kann nicht automatisch in einen numerischen Wert konvertiert werden. Das wird durch einen rot ausgefüllten Pin visualisiert.

4.4.2 Grafischer Editor

Grafische Programme werden mit dem grafischen Editor bearbeitet.

Dazu können verschiedene Knoten im Editor platziert werden, um anschließend durch die grafische Verbindung ihrer Pins zu einem funktionalen Programm zusammengebaut werden.

Der grafische Editor ist an verschiedenen Stellen verfügbar:

- Akquisition
  - Digitale Eingänge
  - PROFINET Eingänge
  - EtherNet/IP Eingänge
  - TCP/IP Variablen Eingänge
- Logik
  - Digitale Ausgänge
  - PROFINET Ausgänge

  - EtherNet/IP Ausgänge
  - TCP/IP Variablen Ausgänge
  - FTP
- Smart Tool
  - Editor

Die Knoten sind in Kategorien eingeteilt und können über das Menü auf der linken Seite ausgewählt werden:

Alternativ kann mit der rechten Maustaste ein Kontextmenü im Editor geöffnet werden:

Nicht an allen Stellen sind alle Kategorien und Knoten verfügbar. Je nach Anwendung sind die Kategorien evtl. eingeschränkt. Beispielsweise sind die PROFINET Knoten nur in den PROFINET Seiten verfügbar.

Tools

Hier finden sich die Werkzeuge, die im Job verfügbar sind. Damit können Resultate dieser Werkzeuge in die Programmierung eingebunden werden.

Create

Hier finden sich Knoten, mit denen Konstanten, Listen oder geometrische Objekte erzeugt werden können.

Constants

Boolean - Definiert einen bool'schen Wert.

Byte - Definiert einen Byte Wert.

Int16 - Definiert einen Int16 Wert.

Int32 - Definiert einen Int32 Wert.

Int64 - Definiert einen Int64 Wert.

Float - Definiert einen Float Wert.

Double - Definiert einen Double Wert.

String - Definiert einen Text.

List

Make List - Konvertiert ein oder mehrere Elemente in eine Liste mit diesen Elementen.

Concatenate - Baut eine Liste aus zwei Listen zusammen.

Geometry

FromData

Angle - Definiert einen Winkel (im Bogenmaß oder in Grad).

Point - Dwefiniert einen Punkt.

Line with Angle - Definiert ein Liniensegment mit Ursprung, Länge und Winkel.

Line from Points - Definiert ein Liniensegment zwischen zwei Punkten.

Circle - Definiert einen Kreis.

FromUI

UI Point - Definiert einen Punkt durch grafische Eingabe im Benutzerinterface.

UI Line - Definiert ein Liniensegment durch grafische Eingabe im Benutzerinterface.

UI Circle - Definiert einen Kreis durch grafische Eingabe im Benutzerinterface.

UI Rectangle - Definiert ein Rechteck durch grafische Eingabe im Benutzerinterface.

Logic

Hier sind die Knoten für logische Verknüpfungen einsortiert.

And - Logische Und-Verknüpfung von zwei oder mehr bool'schen Werten.

Or - Logische Oder-Verknüpfung von zwei oder mehr bool'schen Werten.

Xor - Logische Exklusive-Oder-Verknüpfung von zwei bool'schen Werten.

Not - Logische Negierung eines bool'schen Werts.

IsNull - Liefert 'True', falls kein Objekt ('Null') am Eingang ist.

Math

Hier sind die Knoten für mathematische Verknüpfungen einsortiert.

Arithmetic

Add - Addiert zwei oder mehr Zahlen.

Subtract - Subtrahiert zwei Zahlen.

Multiply - Multipliziert zwei oder mehr Zahlen.

Divide - Dividiert zwei Zahlen.

Compare - Vergleicht zwei Zahlenwerte.

Within Range - Überprüft ob sich ein Wert innerhalb eines gegebenen Bereichs befindet.

Round - Rundet eine Fließkomma-Zahl auf eine ganze Zahl.

Trigonometry

Radians -> Degrees - Konvertiert einen Winkel vom Bogenmaß in Grad.

Degrees -> Radians - Konvertiert einen Winkel von Grad ins Bogenmaß.

Sine - Berechnet den Sinus eines Winkels.

Arcus Sine - Berechnet den Arcus-Sinus. Der Arcus-Sinus ist die Umkehrfunktion des Sinus.

Cosine - Berechnet den Kosinus eines Winkels.

Arcus Cosine - Berechnet den Arcus-Cosinus. Der Arcus-Cosinus ist die Umkehrfunktion des Cosinus.

Tangent - Berechnet den Tangens eines Winkels.

Arcus Tangent - Berechnet den Arcus-Tangens. Der Arcus-Tangens ist die Umkehrfunktion des Tangens.

Text

Hier sind die Knoten für Textverarbeitung einsortiert.

Arrange

Concatenate - Baut einen Text aus einem oder mehreren Teilen zusammen.

Split - Trennt einen Text an bestimmten Trennzeichen.

Join - Verbindet eine Liste von Texten zu einem String.

Compare

Compare - Vergleicht zwei Texte.

Contains - Prüft ob ein Text in einem anderen enthalten ist.

Format

Format number - Formatiert eine Zahl als Text mit einer angegebenen Länge und Anzahl von Nachkommastellen.

Replace - Ersetzt einen Such-Text in einem Text.

Substring - Liefert eine Teilzeichenfolge.

Trim - Entfernt alle führenden und nachfolgeneen Vorkommen einer in einer anderen Zeichenkette angegebenen Gruppe von Zeichen.

Trim Left - Entfernt alle führenden Vorkommen einer in einer anderen Zeichenkette angegebenen Gruppe von Zeichen.

Trim Right - Entfernt alle nachfolgenen Vorkommen einer in einer anderen Zeichenkette angegebenen Gruppe von Zeichen.

Pad Left - Gibt einen Text der angegebenen Länge zurück, in dem der Anfang des aktuellen Texts mit einem angegebenen Zeichen aufgefüllt ist.

Pad Right - Gibt einen Text der angegebenen Länge zurück, in dem das Ende des aktuellen Texts mit einem angegebenen Zeichen aufgefüllt ist.

Truncate - Kürzt eine Zeichenkette auf eine bestimmte Länge.

List

Hier sind die Knoten für die Listenverarbeitung einsortiert.

Access

Get Item - Wählt ein Element aus einer Liste aus.

First - Wählt das erste Element aus einer Liste aus.

Last - Wählt das letzte Element aus einer Liste aus.

Modify

Take - Gibt eine angegebene Anzahl von zusammenhängenden Elementen ab dem Angang einer Liste zurück.

Skip - Überspringt eine festgelegte Anzahl von Elementen in einer Sequenz und gibt die übrigen Elemente zurück.

Sort - Sortiert die Elemente in einer Liste.

Reverse - Kehrt die Reihenfolge der Elemente in einer Liste um.

Distinct - Entfernt doppelte Werte in einer Liste.

Transform - Transformiert eine Liste in eine zweite Liste.

Accumulate - Iteriert über die Elemente in einer Liste und berechnet einen akkumulierten Wert.

Set Operators

Union - Bildet die mengen-theoretische Vereinigungsmenge aus zwei Listen.

Intersect - Bildet die mengen-theoretische Schnittmenge aus zwei Listen.

Except - Bildet die mengen-theoretische Differenz aus zwei Listen.

Geometry

Hier sind die Knoten für geometrische Berechnungen einsortiert:

Distance

Point to Point - Berechnet die Distanz zwischen zwei Punkten.

Point to Line - Berechnet die Distanz zwischen einem Punkt und einer Linie.

Point to Circle - Berechnet die Distanz zwischen einem Punkt und einem Kreis.

Measurement

Angle between lines - Berechnet den Winkel zwischen zwei Linien.

Center of Points - Berechnet den Mittelpunkt einer Liste von Punkten.

Intersection

Line with Line - Berechnet den Schnittpunkt zweier Linien-Segmente.

Line with Circle - Berechnet die Schnittpunkte eines Linien-Segments und eines Kreises.

Circle with Circle - Berechnet die Schnittpunkte zweier Kreise.

Sequence Control

BinarySelector - Wählt einen von zwei Eingängen aus, abhängig vom Zustand eines bool'schen Eingangs.

Fallback Value - Wählt einen Standardwert, wenn kein Wert eingegeben wird.

State Writer - Schreibt einen Zustandswert in den Zustandsspeicher.

State Reader - Liest einen Zustandswert aus dem Zustansspeicher.

Visualization

Hier sind die Knoten einsortiert, mit denen eine kundenspezifische Visualisierung gebaut werden kann.

Text Display - Erzeugt eine visuelle Darstellung eines Textes.

Point Display - Erzeugt eine visuelle Darstellung eines Punktes.

Line Display - Erzeugt eine visuelle Darstellung einer Linie.

Circle Display - Erzeugt eine visuelle Darstellung eines Kreises.

Rectangle Display - Erzeugt eine visuelle Darstellung eines Rechtecks.

Brush - Erzeugt einen farbigen Brush.

Link

Link - Erzeugt eine Verknüpfung.

Results

Primary Result - Erzeugt das primäre Resultat für ein Smart Tool.

Secondary Result - Erzeugt ein sekundäres Resultat für ein Smart Tool.

Advanced Result - Erzeugt ein weiteres Resultat für ein Smart Tool.

Pose Result - Erzeugt eine Nachführungspose im Smart Tool.

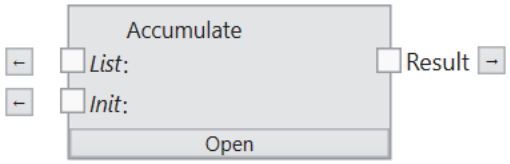
Comment

Comment - Ein Kommentar in einem Funktionsdiagramm.

4.5 Knoten

4.5.1 Accumulate

Iteriert über die Elemente in einer Liste und berechnet einen akkumulierten Wert.



Die Berechnung des akkumulierten Wertes wird durch ein grafisches Programm erzeugt. Die Taste **Open** öffnet einen Editor, in dem das grafische Programm erstellt werden kann.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ `T` kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

Init (Type: `U`)

Der Initialwert für die Akkumulation. Der Typ `U` kann beliebig sein. `U` kann gleich sein wie `T`, muss es aber nicht.

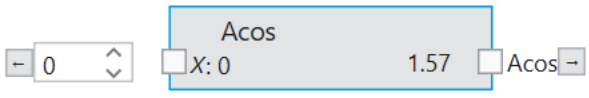
Ausgänge

Result (Type: `U`)

Der akkumulierte Wert.

4.5.2 Acos

Berechnet den Arcus Cosinus. Der Arcus Cosinus ist die Umkehrfunktion des Cosinus.



Eingänge

X (Type: `Double`)

Das Argument der Arcus Cosinus Funktion, im Bereich von -1 bis +1.

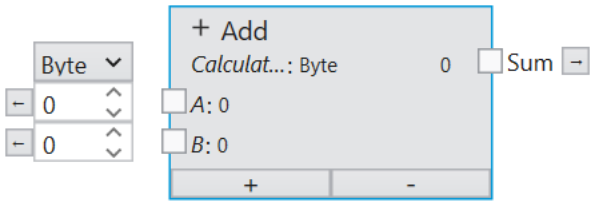
Ausgänge

Acos (Type: `Double`)

Der Winkel im Bogenmaß.

4.5.3 Add

Addiert zwei oder mehr Zahlen.



Die Addition erfolgt mit einem spezifizierten Datentyp.

Mit den **+** und **-** Tasten können Eingänge am Ende hinzugefügt oder vom Ende gelöscht werden.

Eingänge

Calculate as (Type: `String`)

Die Berechnung erfolgt im spezifizierten Datentyp: **Byte**, **Int16**, **Int32**, **Int64**, **Float** oder **Double**.

- Byte erlaubt ganzzahlige Zahlenwerte zwischen 0 und 255.
- Int16 erlaubt ganzzahlige Zahlenwerte zwischen -32.768 und 32.767.
- Int32 erlaubt ganzzahlige Zahlenwerte zwischen -2.147.483.648 und 2.147.483.647.
- Int64 erlaubt ganzzahlige Zahlenwerte zwischen -9.223.327.036.854.775.808 und 9.223.372.036.854.775.807.
- Float erlaubt Fließkommazahlen zwischen +/-1,5 x 10^-45 und +/-3,4 x 10^38 mit einer Präzision von 6-9 Stellen.
- Double erlaubt Fließkommazahlen zwischen +/-5,0 x 10^-324 und +/-1,7 x 10^308 mit einer Präzision von 15-17 Stellen.

A...Z (Type: `Double`)

Die zu addierenden Zahlen. Alle numerischen Datentypen werden nach Bedarf konvertiert.

Ausgänge

Sum (Type: `Byte|Int16|Int32|Int64|Single|Double`)

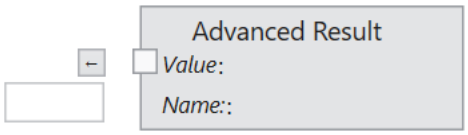
Die Summe der Zahlen, entsprechend dem spezifizierten Datentyp.

Sobald der Ausgäng verbunden ist, ist der Typ festgelegt und kann nicht mehr geändert werden. Zum Ändern des Typs muss die Verknüpfung des Ausgangs gelöscht werden.

Wenn das Ergebnis im spezifizierten Datentype nicht darstellbar ist, produziert der Knoten kein numerisches Ergebnis.

4.5.4 Advanced Result

Erzeugt ein weiteres Resultat für ein Smart Tool.



Das Smart Tool kann verschiedene Resultate erzeugen.

- Das primäre Ergebnis ist ein bool'scher Wert, der anzeigt, ob das Werkzeug Ergebnis in Ordnung ist (`1`) oder ein Fehler auftrat (`0`).
- Sekundäre Ergebnisse können Messwerte sein, die prominent im Werkzeug angezeigt werden.
- Weitere Ergebnisse können Messwerte sein, die im Werkzeug nur angezeigt werden, wenn sie ausgeklappt werden.
- Eine Pose wird benötigt, um eine spezifische Nachführung zu erstellen.

Eingänge

Value (Type: `Object`)

Ein weiteres Werkzeug Ergebnis.

Name (Type: `String`)

Die Bezeichnung des Ergebnisses.

4.5.5 All Ok

Liefert das zusammengefasste Ergebnis aller Werkzeuge.



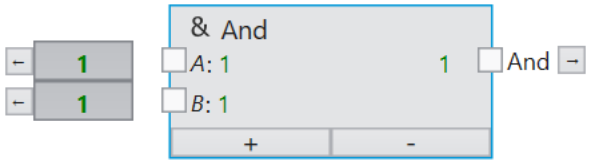
Ausgänge

Result (Type: `Boolean`)

Die Und-Verknüpfung der Ergebnisse aller Werkzeuge.

4.5.6 And

Logische Und-Verknüpfung von zwei oder mehr bool'schen Werten.



Die Wertetabelle für eine logische Und Verknüpfung mit zwei Variablen sieht so aus:

Und	A	B
0	0	0
0	0	1
0	1	0
1	1	1

Mit den **+** und **-** Tasten können weitere Eingänge hinzugefügt bzw. entfernt werden.

Eingänge

A...Z (Type: `Boolean`)

Die zu verknüpfenden bool'schen Eingangswerte.

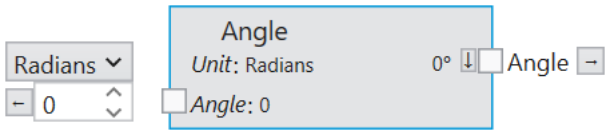
Ausgänge

And (Type: `Boolean`)

Die Und-Verknüpfung der Eingänge. Das Ergebnis ist True/1 wenn alle Eingänge True/1 sind, andernfalls ist das Ergebnis False/0.

4.5.7 Angle

Definiert einen Winkel (im Bogenmaß oder in Grad).



Eingänge

Unit (Type: `String`)

Der Winkel kann im Bogenmaß **Radians** oder in Grad **Degrees** eingegeben werden.

Angle (Type: `Double`)

Der Winkel im Bogenmaß oder in Grad.

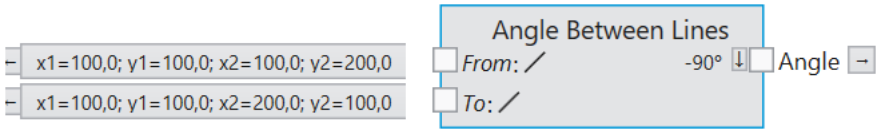
Ausgänge

Angle (Type: `Angle`)

Der Winkel.

4.5.8 Angle Between Lines

Berechnet den Winkel zwischen zwei Linien.



Eingänge

From, To (Type: `LineSegmentDouble`)

Die Linien.

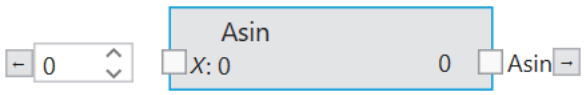
Ausgänge

Angle (Type: `Angle`)

Der Winkel zwischen den Linien.

4.5.9 Asin

Berechnet den Arcus Sinus. Der Arcus Sinus ist die Umkehrfunktion des Sinus.



Inputs

X (Type: `Double`)

Das Argument der Arcus Sinus Funktion, im Bereich von -1 bis +1.

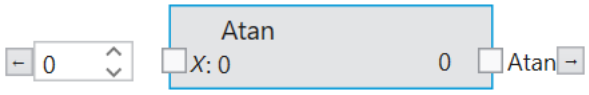
Outputs

Asin (Type: `Double`)

Der Winkel im Bogenmaß.

4.5.10 Atan

Berechnet den Arcus Tangens. Der Arcus Tangens ist die Umkehrfunktion des Tangens.



Inputs

X (Type: `Double`)

Das Argument der Arcus Tangens Funktion.

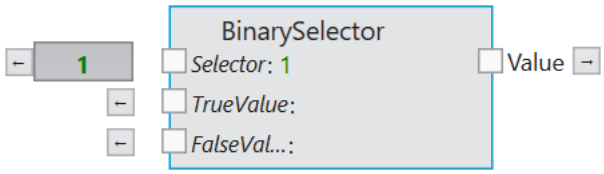
Outputs

Atan (Type: `Double`)

Der Winkel im Bogenmaß.

4.5.11 BinarySelector

Wählt einen von zwei Eingängen aus, abhängig vom Zustand eines booleschen Eingangs.



Eingänge

Selector (Type: `Boolean`)

Der Selektor. Wenn dieser Eingang `True` ist, wird der Wert am *TrueValue* Eingang am Ausgang angezeigt. Ist dieser Eingang `False`, wird der Wert am *FalseValue*-Eingang am Ausgang angezeigt.

TrueValue (Type: `Object`)

Dieser Wert wird am *Value*-Ausgang angezeigt, wenn der Wert am *Selector*-Eingang `True` ist.

FalseValue (Type: `Object`)

Dieser Wert wird am *Value*-Ausgang angezeigt, wenn der Wert am *Selector*-Eingang `False` ist.

Ausgänge

Value (Type: `Object`)

Der Wert entweder des Wertes am *TrueValue* Eingang oder des Wertes am *FalseValue* Eingang, abhängig vom Wert am *Selector* Eingang.

Kommentare

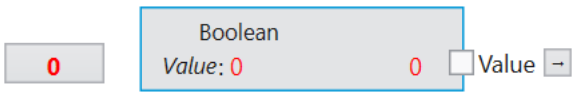
Die Typen der Eingänge und der Ausgänge können beliebig sein, d.h. sie müssen mindestens vom Typ `Object` sein. Wenn ein Typ konkreter ist, müssen die anderen Typen vom gleichen Typ sein (oder null, d.h. nicht verbunden).

Die einfachste Verwendung des Knotens **BinarySelector** ist die Auswahl zwischen zwei Werten.

Eine andere Verwendung besteht darin, die Ausführung eines Teils einer Pipeline zu verhindern, indem `null` in den Rest der Pipeline eingegeben wird. Wenn die Eingabe offen bleibt, wird "null" an die Ausgabe weitergegeben, und infolgedessen können die davon abhängigen Knoten nicht ausgeführt werden.

4.5.12 Boolean

Definiert einen bool'schen Wert.



Eingänge

Value (Type: `Boolean`)

Mit dem Editor am Eingang *Value* kann der Wert auf `0` (False) oder `1` (True) gesetzt werden.

Ausgänge

Value (Type: `Boolean`)

Der bool'sche Wert.

4.5.13 Brush

Erzeugt einen farbigen Brush.

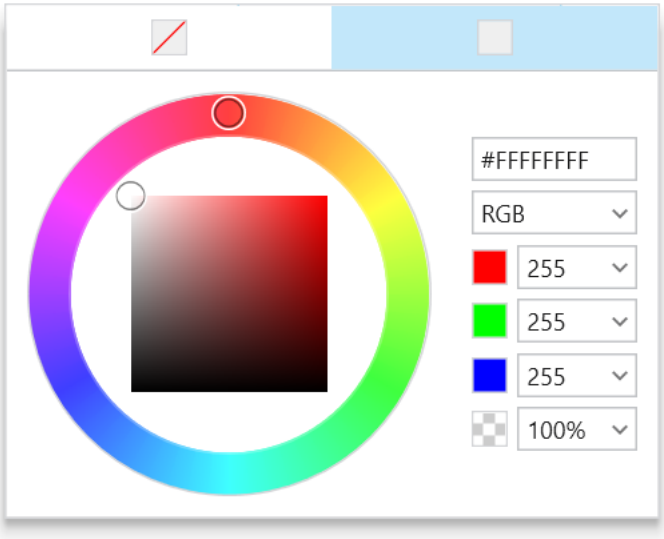


Eingänge

Brush (Type: `SolidColorBrushByte`)

Die Definition des Brushes.

Der Brush kann mit einem Editor grafisch eingegeben werden (Klick auf den Abwärts-Pfeil):



Alternativ kann der Brush als Name für eine Farbe eingeben werden, z.B. `red` für Rot. Die möglichen Farbnamen sind an die Farbdefinitionen in HTML/CSS angelehnt (siehe https://htmlcolorcodes.com/color-names/).

Weiterhin kann der Brush als hexadezimale Zahl eingegeben werden, z.B. `0xFFFFFFFF` für Weiß.

Die ersten beiden Ziffern stehen dabei für die Transparenz, z.B. 00 für vollständig transparent, 80 für halb-transparent und FF für vollständig undurchsichtig.

Die nächsten sechs Ziffern stehen für die Farbe, die ersten zwei Ziffern für Rot, die mittleren zwei Ziffern für Grün und die letzen zwei Ziffern für Blau.

Ausgänge

Brush (Type: `SolidColorBrushByte`)

Der eingestellte Brush.

4.5.14 Byte

Definiert einen Byte Wert.



Eingänge

Value (Type: `Byte`)

Mit dem Editor am Eingang *Value* kann der Wert auf ganzzahlige Werte zwischen `0` und `255` gesetzt werden.

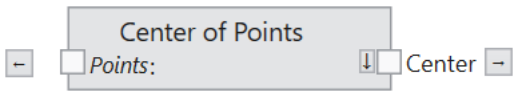
Ausgänge

Value (Type: `Byte`)

Der Byte Wert.

4.5.15 Center of Points

Berechnet den Mittelpunkt einer Liste von Punkten.



Eingänge

Points (Type: `PointList`)

Die Punkte.

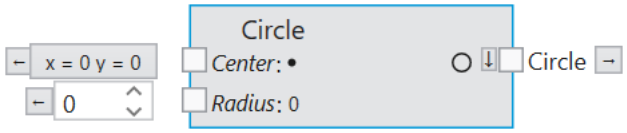
Ausgänge

Point (Type: `Point`)

Der Mittelpunkt.

4.5.16 Circle

Definiert einen Kreis.



Eingänge

Center (Type: `Point`)

Der Mittelpunkt des Kreises.

Radius (Type: `Double`)

Der Radius des Kreises.

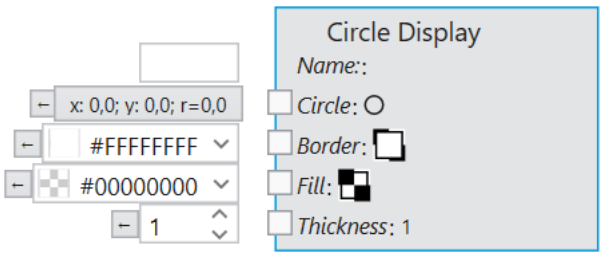
Ausgänge

Circle (Type: `Circle`)

Der Kreis.

4.5.17 Circle Display

Erzeugt einen visuelle Darstellung einese Kreises.



Der Kreis wird auf dem Smart Tool Display und auf der HTML Visualisierung angezeigt.

Eingänge

Name (Type: `String`)

Der Name des Kreises, der auf der HTML Visualisierung angezeigt wird, um den Kreis ein bzw. auszuschalten.

Position (Type: `Circle`)

Die Position des Kreises.

Border (Type: `SolidColorBrushByte`)

Die Farbe des Umrisses. Standardmäßig ist die Farbe weiß.

Fill (Type: `SolidColorBrushByte`)

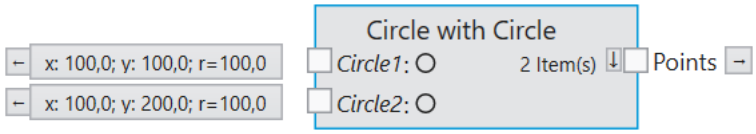
Die Farbe der Füllung. Standardmäßig ist die Farbe durchsichtig.

Thickness (Type: `Double`)

Die Dicke der Umrisslinie.

4.5.18 Circle with Circle

Berechnet die Schnittpunkte zweier Kreise.



Eingänge

Circle1, Circle2 (Type: `Circle`)

Die Kreise.

Ausgänge

Points (Type: `PointListPointDouble`)

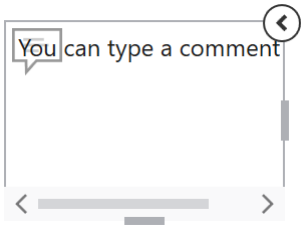
Die Schnittpunkte.

4.5.19 Comment

Ein Kommentar in einem Funktionsdiagramm.

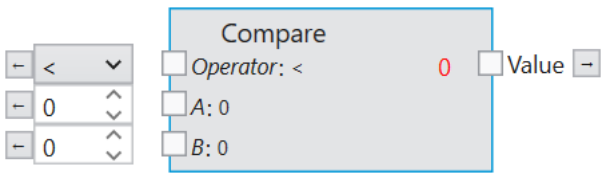


Der Kommentar dient dazu, den Zweck eines Funktionsdiagramms oder eines Teils zu beschreiben. Das Kommentarfeld kann ein- oder ausgeklappt werden.



4.5.20 Compare

Vergleicht zwei Zahlenwerte.



Die Zahlen werden mit verschiedenen Operatoren verglichen. Wenn der Operator zutrifft wird True/1 bzw. wenn es nicht zutrifft wird False/0 zurückgeliefert.

Eingänge

Operator (Type: `String`)

Der Vergleichsoperator. '<', '<=', '==', '!=', '>' und '>=' stehen zur Verfügung.

A, B (Type: `Double`)

Die Zahlen, die verglichen werden sollen. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

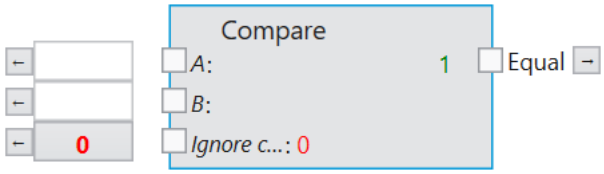
Ausgänge

Value (Type: `Boolean`)

Das Vergleichsergebnis.

4.5.21 Compare

Vergleicht zwei Texte.



Bei Gleichheit wird True/1 bzw. bei Ungleichheit False/0 zurückgeliefert.

Eingänge

A,B (Type: `String`)

Die Texte, die verglichen werden sollen.

Ignore case (Type: `Boolean`)

Die Groß- bzw. Kleinschreibung wird ignoriert, wenn dieser Wert auf True/1 gesetzt ist. In diesem Fall werden die beiden Texte 'A' und 'a' als gleich behandelt. Wenn der Wert auf False/0 gesetzt ist, sind 'A' und 'a' nicht gleich.

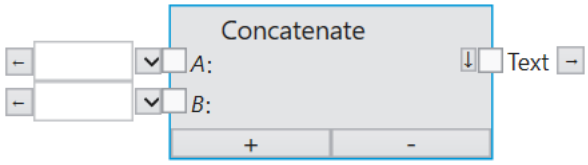
Ausgänge

Equal (Type: `Boolean`)

Das Vergleichsergebnis: True/1 wenn die Texte gleich sind bzw. False/0 wenn die Texte ungleich sind. Zwei leere Zeichenketten werden als gleich betrachtet.

4.5.22 Concatenate

Baut einen Text aus einem oder mehreren Texten zusammen.



An den Eingängen können Texte eingegeben, aus einem Menü ausgewählt (optional) oder von anderen Knoten verbunden werden.

Mit den **+** und **-** Tasten können Eingänge am Ende hinzugefügt oder vom Ende gelöscht werden.

Eingänge

A...Z (Type: `String`)

Die Textbausteine, aus denen ein Gesamttext zusammengebaut wird.

Es gibt drei Möglichkeiten, den Text zu spezifizieren:

1. Der Text kann direkt eingegeben werden.
2. Ein Makro kann aus einer Liste ausgewählt werden (optional).
3. Der Eingang kann mit dem Ausgang eines anderen Knoten verbunden werden. In diesem Fall wird der übertragene Wert automatisch in einen String konvertiert, falls er nicht schon vom Typ String ist.

Es stehen folgende optionale Makros zur Verfügung:

- Acquisition ID
- Job Name
- Job Slot
- Sensor Name
- Sensor Seriennummer
- Jahr
- Monat
- Tag
- Stunde
- Minute
- Sekunde
- Millisekunde
- Zeilenvorschub ()
- Wagenrücklauf ()

- Wagenrücklauf & Zeilenvorschub ()
- Tabulator ()
- Leerzeichen ()

Nicht alle Makros stehen in allen Editoren zur Verfügung. Ein Eintrag kann mit dem **Disconnect** Kommando gelöscht werden.

Es werden folgende Escape Sequenzen verwendet:

- `\\` -> (Backslash)
- `\r` -> (Wagenrücklauf)
- `\n` -> (Zeilenvorschub)
- `\t` -> (Tabulator)
- `\x##` -> ASCII Code ##, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht
- `\u####` -> UTF16 Code ####, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht

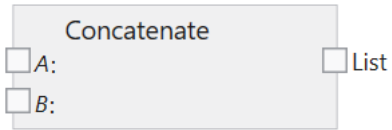
Ausgänge

Text (Type: `String`)

Der zusammengebaute Text.

4.5.23 Concatenate

Baut eine Liste aus zwei Listen zusammen.



Eingänge

A,B (Type: `List<T>`)

Die Listen, aus denen die gesamte Liste zusammengebaut wird. Die Listen müssen denselben Element-Typ haben.

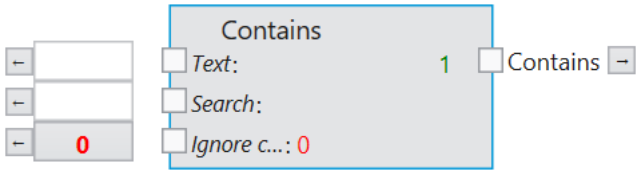
Ausgänge

List (Type: `List<T>`)

Die zusammengebaute Liste.

4.5.24 Contains

Prüft ob ein Text in einem anderen Text enthalten ist.



Es wird geprüft, ob der Such-Text im Text enthalten ist und True/1 bzw. False/0 zurückgeliefert.

Eingänge

Text (Type: `String`)

Der Text in dem der Such-Text gesucht wird.

Search (Type: `String`)

Der Such-Text.

Ignore case (Typ: `Boolean`)

Die Groß- bzw. Kleinschreibung wird ignoriert, wenn dieser Wert auf True/1 gesetzt ist.

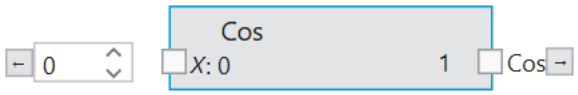
Ausgänge

Contains (Type: `Boolean`)

Das Suchergebnis: True/1 wenn der Such-Text im Text enthalten ist bzw. False/0 wenn er nicht enthalten ist. Ein leerer Suchtext ist in allen Texten (auch einem leeren Text) enthalten.

4.5.25 Cos

Berechnet den Cosinus eines Winkels.



Eingänge

X (Type: `Double`)

Das Argument der Cosinus Funktion, im Bogenmaß.

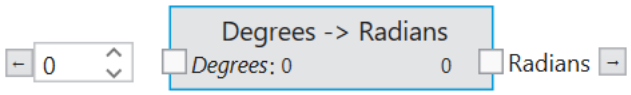
Ausgänge

Cos (Type: `Double`)

Der Cosinus des Winkels.

4.5.26 Degrees -> Radians

Konvertiert einen Winkel von Grad ins Bogenmaß.



Ein Winkel wird so von Grad ins Bogenmaß konvertiert:

$radians = degrees * \frac{\pi}{180}$

Eingänge

Degrees (Type: `Double`)

Der Winkel in Grad.

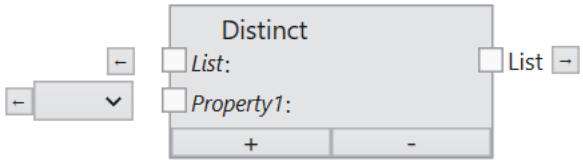
Ausgänge

Radians (Type: `Double`)

Der Winkel im Bogenmaß.

4.5.27 Distinct

Entfernt doppelte Werte in einer Liste.



Mit den **+** und **-** Tasten können Eingänge (`PropertyN`) am Ende hinzugefügt oder vom Ende gelöscht werden.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

Property[1..N] (Type: `String`)

Die Eigenschaft, mittels der Duplikate ermittelt werden. Es können die Properties des Typs `T` gewählt werden, oder `Self` um `T` selbst zu verwenden. Wenn mehrere Eigenschaften gewählt sind, werden alle verwendet um Duplikate zu ermitteln.

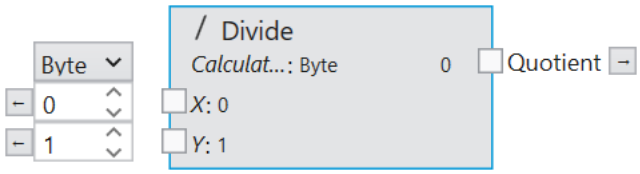
Ausgänge

Sorted (Type: `List<T>`)

Die Liste ohne doppelte Werte.

4.5.28 Divide

Dividiert zwei Zahlen.



Die Division erfolgt mit einem spezifizierten Datentyp.

Eingänge

Calculate as (Type: `String`)

Die Berechnung erfolgt im spezifizierten Datentyp: **Byte**, **Int16**, **Int32**, **Int64**, **Float** oder **Double**.

- Byte erlaubt ganzzahlige Zahlenwerte zwischen 0 und 255.
- Int16 erlaubt ganzzahlige Zahlenwerte zwischen -32.768 und 32.767.
- Int32 erlaubt ganzzahlige Zahlenwerte zwischen -2.147.483.648 und 2.147.483.647.
- Int64 erlaubt ganzzahlige Zahlenwerte zwischen -9.223.327.036.854.775.808 und 9.223.372.036.854.775.807.
- Float erlaubt Fließkommazahlen zwischen +/-1,5 x 10^-45 und +/-3,4 x 10^38 mit einer Präzision von 6-9 Stellen.
- Double erlaubt Fließkommazahlen zwischen +/-5,0 x 10^-324 und +/-1,7 x 10^308 mit einer Präzision von 15-17 Stellen.

X (Type: `Double`)

Der Dividend. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

Y (Type: `Double`)

Der Divisor. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

Ausgänge

Quotient (Type: `Byte|Int16|Int32|Int64|Single|Double`)

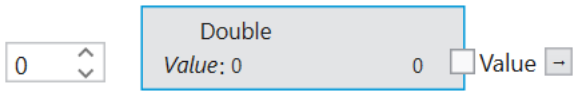
Die Quotient der Zahlen, entsprechend dem spezifizierten Datentyp.

Sobald der Ausgäng verbunden ist, ist der Typ festgelegt und kann nicht mehr geändert werden. Zum Ändern des Typs muss die Verknüpfung des Ausgangs gelöscht werden.

Wenn das Ergebnis im spezifizierten Datentype nicht darstellbar ist, produziert der Knoten kein numerisches Ergebnis.

4.5.29 Double

Definiert einen Fließkommawert mit 64 bit Genauigkeit.



Eingänge

Value (Type: `Double`)

Mit dem Editor am Eingang *Value* kann der Wert auf Fließkommawerte zwischen +/-5,0 x 10^-324 und +/-1,7 x 10^308 mit einer Präzision von 15-17 Stellen gesetzt werden.

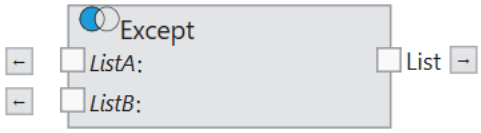
Ausgänge

Value (Type: `Double`)

Der Fließkommawert.

4.5.30 Except

Bildet die mengen-theoretische Differenz aus zwei Listen.



Eingänge

ListA, ListB (Type: `List<T>`)

Die zwei Listen. Der Typ kann beliebig sein, aber alle Elemente beider Listen haben denselben Typ.

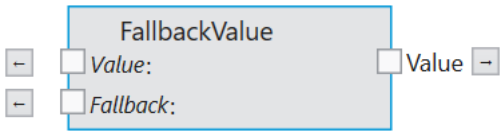
Ausgänge

List (Type: `List<T>`)

Die mengen-theoretische Differenz (ListA - ListB) aus beiden Listen.

4.5.31 FallbackValue

Wählt einen Standardwert, wenn kein Wert eingegeben wird.



Eingänge

Value (Type: `Object`)

Dieser Wert wird an den Ausgang weitergeleitet. Ist der Wert `null`, wird stattdessen der Wert an Port *Fallback* weitergeleitet.

Fallback (Type: `Object`)

Dieser Wert wird an den Ausgang weitergeleitet, wenn der Wert an Port *Value* `null` ist

Ausgänge

Value (Type: `Object`)

Der Wert entweder des Wertes am *Value*-Eingang oder des Wertes am *Fallback*-Eingang, je nach dem Wert am *Value*-Eingang.

Kommentare

Die Typen der Eingänge und der Ausgänge können beliebig sein, d.h. sie müssen mindestens vom Typ `Object` sein. Wenn ein Typ konkreter ist, müssen die anderen Typen vom gleichen Typ sein (oder null, d.h. nicht verbunden).

4.5.32 First

Wählt das erste Element aus einer Liste aus.



Eine Liste besteht aus mehreren, aneinandergereihten Elementen. Dieser Knoten wählt das erste Element aus der Liste aus.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

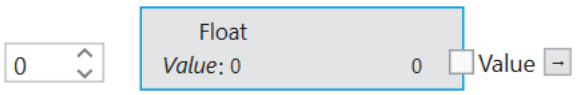
Ausgänge

Item (Type: `T`)

Das erste Element der Liste.

4.5.33 Float

Definiert einen Fließkommawert mit 32 bit Genauigkeit.



Eingänge

Value (Type: `Float`)

Mit dem Editor am Eingang *Value* kann der Wert auf Fließkommawerte zwischen +/-1,5 x 10^-45 und +/-3,4 x 10^38 mit einer Präzision von 6-9 Stellen gesetzt werden.

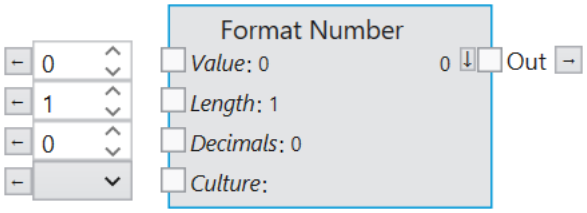
Ausgänge

Value (Type: `Float`)

Der Fließkommawert.

4.5.34 Format Number

Formatiert eine Zahl als Text mit einer angegebenen Länge und Anzahl von Nachkommastellen.



Eingänge

Value (Type: `Double`)

Die Zahl, die formatiert werden soll.

Length (Type: `Int32`)

Die gewünschte Gesamtlänge der Zahl. Bei Bedarf wird links mit Leerzeichen aufgefüllt, bis die gewünschte Länge erreicht ist.

Decimals (Type: `Int32`)

Die gewünschte Anzahl der Nachkommastellen. Die letzte Ziffer wird bei Bedarf entsprechend gerundet.

Culture (Typ: `String`)

Sprachspezifische kulturelle Zahlenformatierung. Wenn diese Option leer bleibt, wird die kulturelle Formatierung des Systems verwendet.

Wenn `Invariant` verwendet wird, wird ein Punkt als Dezimaltrennzeichen verwendet.

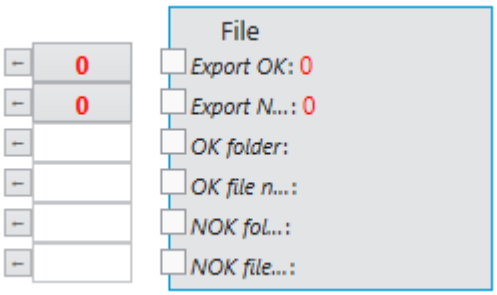
Ausgänge

Out (Type: `String`)

Die formatierte Zahl.

4.5.35 Image Upload

Konfiguriert das Hochladen von Bildern.



Eingänge

ExportOK (Type: `Boolean`)

Konfiguriert, ob IO Bilder exportiert werden sollen.

ExportNOK (Type: `Boolean`)

Konfiguriert, ob NIO Bilder exportiert werden sollen.

FolderOK (Type: `String`)

Der Ordnername, in den IO Bilder exportiert werden sollen.

FileNameOK (Type: `String`)

Der Dateiname für IO Bilder.

FolderNOK (Type: `String`)

Der Ordnername, in den NIO Bilder exportiert werden sollen.

FileNameNOK (Type: `String`)

Der Dateiname für NIO Bilder.

4.5.36 In

Liest Daten von einem digitalen Eingang.



Der Vision Sensor **CS-6x/ID-6x0** verfügt über zwei digitale Eingänge, die verknüpft werden können. Die Eingangsknoten werden auf der Seite Digitale Eingänge des Werkzeugs Akquisition angezeigt. Um sie zum Verlinken zu verwenden, müssen sie benannt werden. Ein Eingang wird benannt durch Hinzufügen eines Link Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von Verknüpfung), Verbinden des Eingabeknoten mit dem Verknüpfung Knoten und Eingabe des entsprechenden Namen in den Verknüpfung Knoten.

Der digitale Eingang wird am Beginn des Inspektionszyklus eingelesen, in etwa zur Zeit des Trigger-Signals.

Ausgänge

Value (Typ: `Boolean`)

Der logische Wert des Eingangs zum Zeitpunkt des Trigger-Signals.

4.5.37 Intersect

Bildet die mengen-theoretische Schnittmenge aus zwei Listen.



Eingänge

ListA, ListB (Type: `List<T>`)

Die zwei Listen. Der Typ kann beliebig sein, aber alle Elemente beider Listen haben denselben Typ.

Ausgänge

List (Type: `List<T>`)

Die mengen-theoretische Schnittmenge aus beiden Listen.

4.5.38 Int16

Definiert einen Int16 Wert.



Eingänge

Value (Type: `Int16`)

Mit dem Editor am Eingang *Value* kann der Wert auf ganzzahlige Werte zwischen `-2.147.483.648` und `2.147.483.647` gesetzt werden.

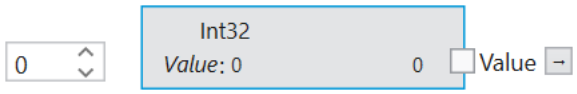
Ausgänge

Value (Type: `Int16`)

Der Int16 Wert.

4.5.39 Int32

Definiert einen Int32 Wert.



Eingänge

Value (Type: `Int32`)

Mit dem Editor am Eingang *Value* kann der Wert auf ganzzahlige Werte zwischen `-2.147.483.648` und `2.147.483.647` gesetzt werden.

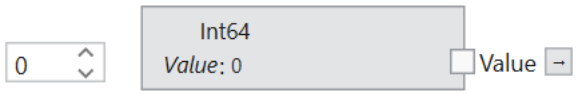
Ausgänge

Value (Type: `Int32`)

Der Int32 Wert.

4.5.40 Int64

Definiert einen Int64 Wert.



Eingänge

Value (Type: `Int64`)

Mit dem Editor am Eingang *Value* kann der Wert auf ganzzahlige Werte zwischen `-9.223.327.036.854.775.808` und `9.223.372.036.854.775.807` gesetzt werden.

Ausgänge

Value (Type: `Int64`)

Der Int64 Wert.

4.5.41 IsNull

Liefert `True`, falls kein Objekt (`Null`) am Eingang ist.



Eingänge

Object (Typ: `Object`)

Wenn dieser Eingang `Null` ist, wird der Wert `True` am Ausgang angezeigt. Ist dieser Eingang ein gültiges Object beliebigen Typs, wird der Wert `False` am Ausgang angezeigt.

Ausgänge

Value (Typ: `Boolean`)

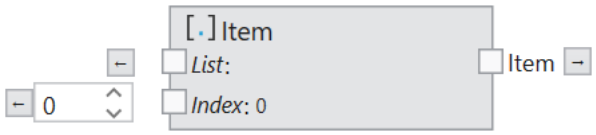
`True` oder `False`, entsprechend des Werts am *Object* Eingang.

Kommentare

Oft wird der **IsNull** Knoten im Zusammenhang mit einem **BinarySelector** verwendet, um beispielsweise einen Default Wert zu bestimmen, für den Fall, dass ein Wert `Null` ist.

4.5.42 Item

Wählt ein Element aus einer Liste aus.



Eine Liste besteht aus mehreren, aneinandergereihten Elementen. Dieser Knoten wählt ein spezielles Element aus der Liste aus, entsprechend seiner Position in der Liste.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

Index (Type: `Int32`)

Der 0-basierte Index des Elements. Liegt der Index außerhalb des zulässigen Bereichs, wird ein Fehler angezeigt.

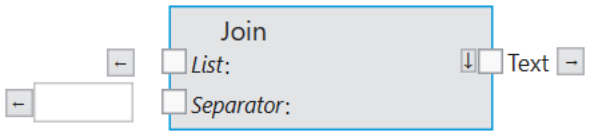
Ausgänge

Item (Type: `T`)

Das spezifische Element der Liste.

4.5.43 Join

Verbindet eine Liste von Texten zu einem String.



Eine Liste von Texten wird in einen String umgewandelt.

Eingänge

Liste (Type: `IList<String>`)

Die Liste von Texten.

Trennzeichen (Type: `String`)

Die Trennzeichen werden optional zwischen den Listenelementen eingefügt.

Es werden folgende Escape Sequenzen verwendet:

- `\\` -> (Backslash)
- `\r` -> (Wagenrücklauf)
- `\n` -> (Zeilenvorschub)
- `\t` -> (Tabulator)
- `\x##` -> ASCII Code ##, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht
- `\u####` -> UTF16 Code ####, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht

Ausgänge

Text (Type: `String`)

Der kombinierte Text.

4.5.44 Last

Wählt das letzte Element aus einer Liste aus.



Eine Liste besteht aus mehreren, aneinandergereihten Elementen. Dieser Knoten wählt das letzte Element aus der Liste aus.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

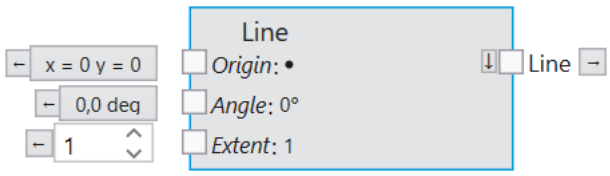
Ausgänge

Item (Type: `T`)

Das letzte Element der Liste.

4.5.45 Line

Definiert ein Liniensegment mit Ursprung, Länge und Winkel.



Eingänge

Origin (Type: `Point`)

Der Ursprung (Startpunkt) der Linie.

Angle (Type: `Angle`)

Der Winkel der Linie.

Extent (Type: `Double`)

Die Länge der Linie.

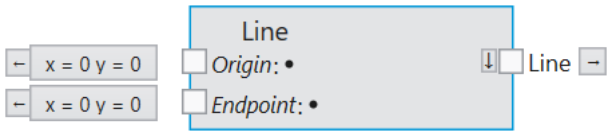
Ausgänge

Line (Type: `LineSegment`)

Das Liniensegment.

4.5.46 Line

Definiert ein Liniensegment zwischen zwei Punkten.



Eingänge

Origin, Endpoint (Type: `Point`)

Der Ursprung (Startpunkt) und Endpunkt (Endpunkt) der Linie.

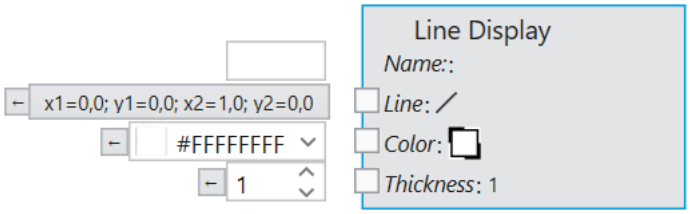
Ausgänge

Line (Type: `LineSegment`)

Das Liniensegment.

4.5.47 Line Display

Erzeugt eine visuelle Darstellung einer Linie.



Die Linie wird auf dem Smart Tool Display und auf der HTML Visualisierung angezeigt.

Eingänge

Name (Type: `String`)

Der Name der Linie, der auf der HTML Visualisierung angezeigt wird, um die Linie ein bzw. auszuschalten.

Position (Type: `LineSegmentDouble`)

Die Position der Linie.

Color (Type: `SolidColorBrushByte`)

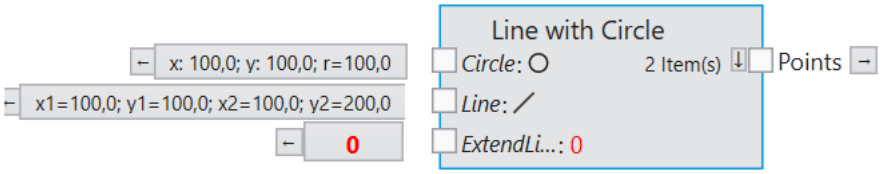
Die Farbe der Linie. Standardmäßig ist die Farbe weiß.

Thickness (Type: `Double`)

Die Dicke der Linie.

4.5.48 Line with Circle

Berechnet die Schnittpunkte eines Linien-Segments und eines Kreises.



Eingänge

Circle (Type: `Circle`)

Der Kreis.

Line (Type: `LineSegmentDouble`)

Das Linien-Segment.

ExtendLines (Type: `Boolean`)

Erweitert das Linien-Segment auf eine unendliche Linie, um die Schnittpunkte zu berechnen.

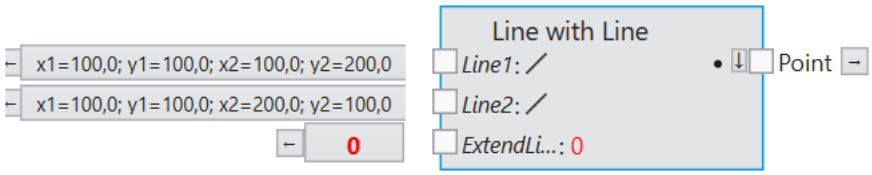
Ausgänge

Points (Type: `PointListPointDouble`)

Die Schnittpunkte.

4.5.49 Line with Line

Berechnet den Schnittpunkt zweier Linien-Segmente.



Eingänge

Line1, Line2 (Type: `LineSegmentDouble`)

Die beiden Linien-Segmente.

ExtendLines (Type: `Boolean`)

Erweitert die Linien-Segmente auf unendliche Linien, um den Schnittpunkt zu berechnen.

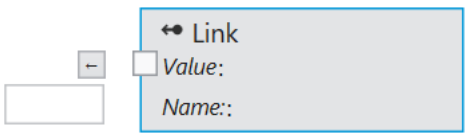
Ausgänge

Point (Type: `PointDouble`)

Der Schnittpunkt.

4.5.50 Link

Erzeugt eine Verknüpfung.



Eine Verknüpfung stellt eine Beziehung zwischen einen Eingangswert und der Verwendung dieses Werts in nachfolgenden Werkzeugen dar. Eingangswerte können durch digitale Eingänge,

Profinet Eingänge oder TCP/IP Variablen Eingänge bereitgestellt werden. Damit sie in nachfolgenden Werkzeugen verwendet werden können, müssen sie benannt werden.

Eingänge

Value (Type: `Object`)

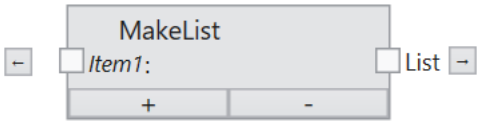
Der Wert der Verknüpfung. Er wird üblicherweise durch Verbinden mit einen digitalen Eingäng, einen Profinet Eingang oder einem TCP/IP Variablen Eingang bereitgestellt.

Name (Type: `Object`)

Der Name der Verknüpfung. Der Name muss eindeutig innerhalb eines Jobs sein. Unter diesem Namen ist der Wert in nachfolgenden Werkzeugen verfügbar.

4.5.51 Make List

Konvertiert ein oder mehrere Elemente in eine Liste mit diesen Elementen.



Mit den **+** und **-** Tasten können Eingänge am Ende hinzugefügt oder vom Ende gelöscht werden.

Eingänge

Item1...ItemN (Type: T)

Ein Element mit beliebigem Typ.

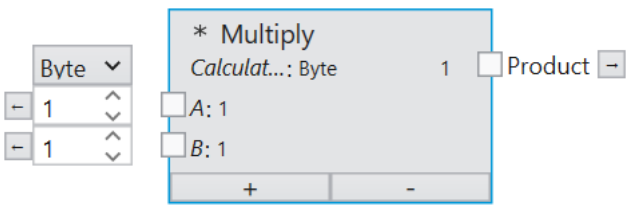
Ausgänge

List (Type: `List<T>`)

Eine Liste mit den Elementen.

4.5.52 Multiply

Multipliziert zwei oder mehr Zahlen.



Die Multiplikation erfolgt mit einem spezifizierten Datentyp.

Mit den **+** und **-** Tasten können Eingänge am Ende hinzugefügt oder vom Ende gelöscht werden.

Eingänge

Calculate as (Type: `String`)

Die Berechnung erfolgt im spezifizierten Datentyp: **Byte**, **Int16**, **Int32**, **Int64**, **Float** oder **Double**.

- Byte erlaubt ganzzahlige Zahlenwerte zwischen 0 und 255.
- Int16 erlaubt ganzzahlige Zahlenwerte zwischen -32.768 und 32.767.
- Int32 erlaubt ganzzahlige Zahlenwerte zwischen -2.147.483.648 und 2.147.483.647.
- Int64 erlaubt ganzzahlige Zahlenwerte zwischen -9.223.327.036.854.775.808 und 9.223.372.036.854.775.807.
- Float erlaubt Fließkommazahlen zwischen +/-1,5 x 10^-45 und +/-3,4 x 10^38 mit einer Präzision von 6-9 Stellen.
- Double erlaubt Fließkommazahlen zwischen +/-5,0 x 10^-324 und +/-1,7 x 10^308 mit einer Präzision von 15-17 Stellen.

A...Z (Type: `Double`)

Die zu multiplizierenden Zahlen. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

Ausgänge

Product (Type: `Byte|Int16|Int32|Int64|Single|Double`)

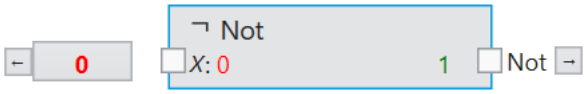
Das Produkt der Zahlen, entsprechend dem spezifizierten Datentyp.

Sobald der Ausgäng verbunden ist, ist der Typ festgelegt und kann nicht mehr geändert werden. Zum Ändern des Typs muss die Verknüpfung des Ausgangs gelöscht werden.

Wenn das Ergebnis im spezifizierten Datentype nicht darstellbar ist, produziert der Knoten kein numerisches Ergebnis.

4.5.53 Not

Logische Negierung eines bool'schen Werts.



Die Wertetabelle für eine logische Nicht Verknüpfung sieht so aus:

Not	X
1	0
0	1

Eingänge

X (Type: `Boolean`)

Der Eingangswert.

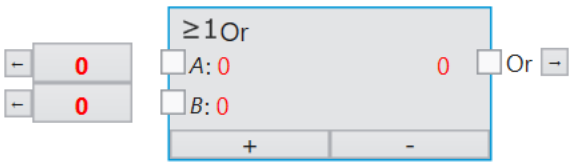
Ausgänge

Not (Type: `Boolean`)

Der negierte Eingang. Das Ergebnis ist True/1 wenn der Eingang False/0 ist, andernfalls ist das Ergebnis False/0.

4.5.54 Or

Logische Oder-Verknüpfung von zwei oder mehr bool'schen Werten.



Die Wertetabelle für eine logische Oder Verknüpfung mit zwei Variablen sieht so aus:

Or	A	B
0	0	0
1	0	1
1	1	0
1	1	1

Mit den **+** und **-** Tasten können weitere Eingänge hinzugefügt bzw. entfernt werden.

Eingänge

A...Z (Type: `Boolean`)

Die zu verknüpfenden bool'schen Eingangswerte.

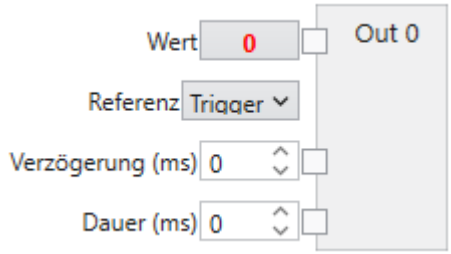
Ausgänge

Oder (Type: `Boolean`)

Die Oder-Verknüpfung der Eingänge. Das Ergebnis ist True/1 wenn mindestens ein Eingang True/1 ist, andernfalls ist das Ergebnis False/0.

4.5.55 Out

Schreibt Daten auf einen digitalen Ausgang.



Der Vision Sensor **CS-6x/ID-6x0** verfügt über vier binäre digitale Ausgänge, die abhängig vom Ergebnis der Inspektion geschaltet werden können.

Wenn digitale Ausgänge für die Signalisierung des Job-Wechsels verwendet werden, können sie nicht gleichzeitig als Inspektionsergebnisse geschaltet werden.

Eingänge

Wert (Typ: `Boolean`)

Der logische Wert der auf den Ausgang geschrieben werden soll.

Referenz (Typ: `String`)

Die Verzögerung des Ausgangssignals kann sich auf den **Trigger** Zeitpunkt beziehen, oder auf das **Inspektionsende**.

Verzögerung (ms) (Typ: `UInt32`)

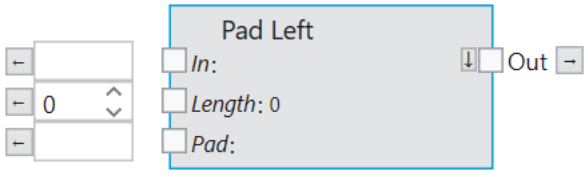
Die Verzögerung des Schaltpunkts in Bezug auf den Zeitpunkt von Trigger oder Inspektionsende. Eine Verzögerung von `0` bedeutet sofortiges Schalten (hier nur nach Inspektionsende, weil erst zu diesem Zeitpunkt der Wert bestimmt werden kann).

Dauer (ms) (Typ: `UInt32`)

Die Dauer des logischen Signals. Eine Dauer von `0` hat ein dauernd anstehendes Signal zur Folge.

4.5.56 Pad Left

Gibt einen Text der angegebenen Länge zurück, in dem der Anfang des aktuellen Texts mit einem angegebenen Zeichen aufgefüllt ist.



Eingänge

In (Type: `String`)

Der Eingangstext.

Length (Type: `Int32`)

Die Länge des Texts.

Character (Type: `String`)

Das erste Zeichen dieser Zeichenkette wird als Auffüllzeichen verwendet. Wird keine Zeichenkette angegeben, wird ein Leerzeichen als Auffüllzeichen verwendet.

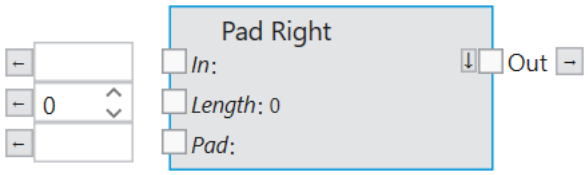
Ausgänge

Out (Typ: `String`)

Der Text, in dem der Anfang des aktuellen Texts mit einem angegebenen Zeichen aufgefüllt ist.

4.5.57 Pad Right

Gibt einen Text der angegebenen Länge zurück, in dem das Ende des aktuellen Texts mit einem angegebenen Zeichen aufgefüllt ist.



Eingänge

In (Type: `String`)

Der Eingangstext.

Length (Type: `Int32`)

Die Länge des Texts.

Character (Type: `String`)

Das erste Zeichen dieser Zeichenkette wird als Auffüllzeichen verwendet. Wird keine Zeichenkette angegeben, wird ein Leerzeichen als Auffüllzeichen verwendet.

Ausgänge

Out (Type: `String`)

Der Text, in dem das Ende des aktuellen Texts mit einem angegebenen Zeichen aufgefüllt ist.

4.5.58 Parse GS1 AI

Parst einen Text, der GS1-Datenbezeichner (Application Identifiers) enthält.

Der Text wird gemäß der GS1-Datenbezeichner-Spezifikation geparst und in eine Liste von Datenbezeichnern umgewandelt.

Die GS1-Datenbezeichner sind im GS1 General Specifications Standard festgelegt. Die unterstützte Version ist Release 25.0. Die Spezifikation ist hier zu finden: https://www.gs1.org/standards/barcodes-epcrfid-id-keys/gs1-general-specifications

Es gibt zwei Versionen dieses Knotens: Die erste nimmt einen einzelnen Text entgegen, die zweite eine Liste von Texten.

Inputs

Code (Typ: `String`)

Der zu parsende Text. Üblicherweise stammt dieser aus einem Barcode, der Datenbezeichner enthält.

Codes (Typ: `IList<string>`)

Die zu parsenden Texte. Üblicherweise stammen diese aus einem Barcode, der Datenbezeichner enthält.

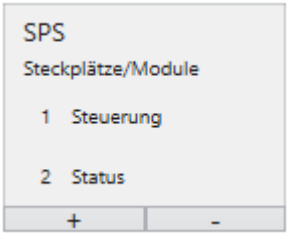
Outputs

AIs (Typ: `GS1ApplicationIdList`)

Eine Liste von Datenbezeichnern.

4.5.59 PLC

Repräsentiert die Konfiguration einer SPS.



Der **PLC** Knoten wird verwendet, um Module in bestimmten Steckplätzen zu definieren. Dieses Setup muss exakt mit dem Setup auf der SPS übereinstimmen, um die Kommunikation zwischen

der SPS und dem **CS-6x/ID-6x0** zu ermöglichen. Der **PLC** Knoten wird sowohl im Akquisition als auch im Logik Werkzeug angezeigt. Innerhalb des **PLC** Knotens werden die Namen Eingang und Ausgang für die Module wie von der SPS aus gesehen verwendet. Ein SPS Ausgang wird von der SPS zum **CS-6x/ID-6x0** übertragen, wo er als Eingang angesehen wird. Ein SPS-Eingang wird vom **CS-6x/ID-6x0** zur SPS übertragen, auf dem **CS-6x/ID-6x0** wird dies als Ausgang gesehen.

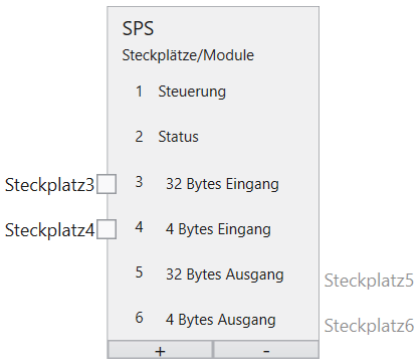
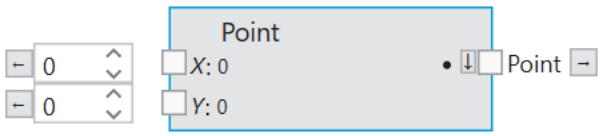


Abb. 170: Der SPS Knoten mit sechs Modulen.

4.5.60 Point

Definiert einen Punkt in kartesischen Koordinaten.



Eingänge

X, Y (Type: `Double`)

Die kartesisichen Koordinaten des Punkts.

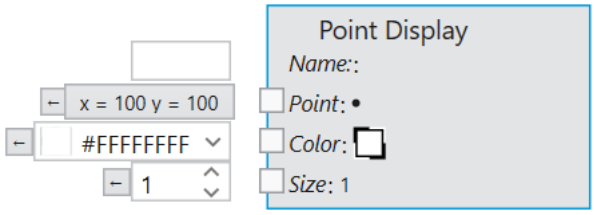
Ausgänge

Point (Type: `PointDouble`)

Der Punkt.

4.5.61 Point Display

Erzeugt eine visuelle Darstellung eines Punktes.



Der Punkt wird auf dem Smart Tool Display und auf der HTML Visualisierung angezeigt.

Eingänge

Name (Type: `String`)

Der Name des Punkts, der auf der HTML Visualisierung angezeigt wird, um den Punkt ein bzw. auszuschalten.

Position (Type: `PointDouble`)

Die Position des Punktes. Standardmäßig liegt der Punkt an Position x=100,y=100.

Color (Type: `SolidColorBrushByte`)

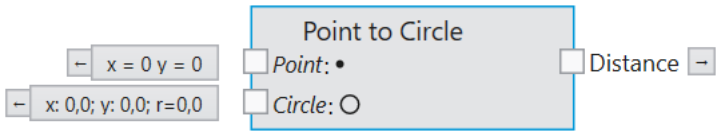
Die Farbe des Punktes. Standardmäßig ist die Farbe weiß.

Size (Type: `Double`)

Die Größe des Punktes.

4.5.62 Point to Circle

Berechnet die Distanz zwischen einem Punkt und einem Kreis.



Eingänge

Point (Type: `PointDouble`)

Der Punkt.

Circle (Type: `Circle`)

Der Kreis.

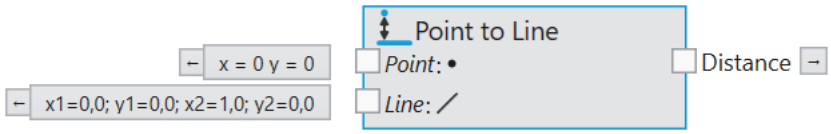
Ausgänge

Distance (Type: `Double`)

Die Distanz zwischen dem Punkt und dem Kreis.

4.5.63 Point to Line

Berechnet die Distanz zwischen einem Punkt und einer Linie.



Eingänge

Point (Type: `PointDouble`)

Der Punkt.

Line (Type: `LineSegmentDouble`)

Die Linie.

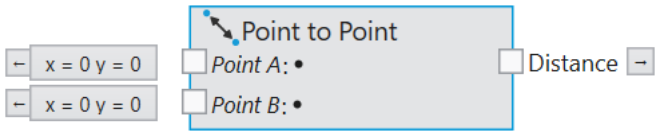
Ausgänge

Distance (Type: `Double`)

Die Distanz zwischen dem Punkt und der Linie.

4.5.64 Point to Point

Berechnet die Distanz zwischen zwei Punkten.



Eingänge

Point A, Point B (Type: `PointDouble`)

Die beiden Punkte.

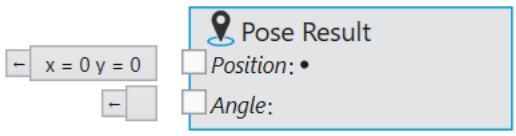
Ausgänge

Distance (Type: `Double`)

Die euklidische Distanz zwischen zwei Punkten.

4.5.65 Pose Result

Erzeugt eine Nachführungspose im Smart Tool.



Das Smart Tool kann verschiedene Resultate erzeugen.

- Das primäre Ergebnis ist ein bool'scher Wert, der anzeigt, ob das Werkzeug Ergebnis in Ordnung ist (`1`) oder ein Fehler auftrat (`0`).
- Sekundäre Ergebnisse können Messwerte sein, die prominent im Werkzeug angezeigt werden.
- Weitere Ergebnisse können Messwerte sein, die im Werkzeug nur angezeigt werden, wenn sie ausgeklappt werden.
- Eine Pose wird benötigt, um eine spezifische Nachführung zu erstellen.

Eingänge

Position (Type: `PointDouble`)

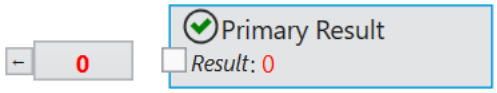
Die Position der Nachführung.

Angle (Type: `Angle`)

Der Winkel der Nachführung (kann in Grad oder im Bogenmaß angegeben werden).

4.5.66 Primary Result

Erzeugt das primäre Resultat für ein Smart Tool.



Das Smart Tool kann verschiedene Resultate erzeugen.

- Das primäre Ergebnis ist ein bool'scher Wert, der anzeigt, ob das Werkzeug Ergebnis in Ordnung ist (`1`) oder ein Fehler auftrat (`0`).
- Sekundäre Ergebnisse können Messwerte sein, die prominent im Werkzeug angezeigt werden.
- Weitere Ergebnisse können Messwerte sein, die im Werkzeug nur angezeigt werden, wenn sie ausgeklappt werden.
- Eine Pose wird benötigt, um eine spezifische Nachführung zu erstellen.

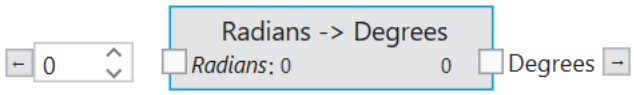
Eingänge

Result (Type: `Boolean`)

Das primäre Werkzeug Ergebnis.

4.5.67 Radians -> Degrees

Konvertiert einen Winkel vom Bogenmaß in Grad.



Ein Winkel wird so vom Bogenmaß in Grad konvertiert:

$degrees = radians * \frac{180}{\pi}$

Eingänge

Radians (Type: `Double`)

Der Winkel im Bogenmaß.

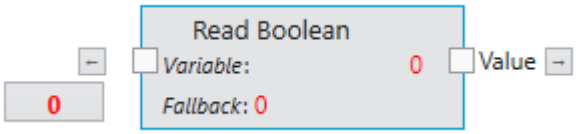
Ausgänge

Degrees (Type: `Double`)

Der Winkel in Grad.

4.5.68 Read Boolean

Liest einen bool'schen Wert aus einer TCP/IP Eingangs-Variablen ein.



Ein bool'scher Eingang wird mit einem TCP/IP Eingang verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

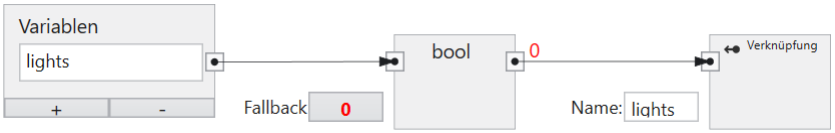


Abb. 171: Ein bool'scher Eingang.

Eingänge

Variable (Type: `TcpIpVariable`)

Verbindet den Knoten mit einer TCP/IP Eingangs-Variablen (TCP/IP -> **CS-6x/ID-6x0**).

Fallback (Type: `Int32`)

Gibt den bool'schen Wert an, der verwendet wird, wenn die TCP/IP-Daten nicht verfügbar sind. TCP/IP-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil der Kommando-Server am Sensor deaktiviert ist, - weil die TCP/IP Eingangs Variable nicht gesetzt wurde.

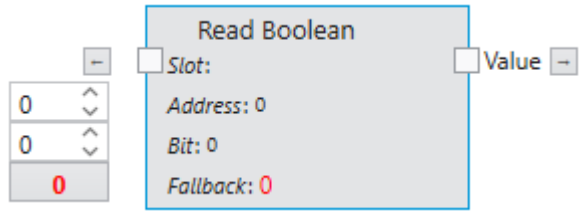
Ausgänge

Value (Type: `Boolean`)

Der über TCP/IP eingelesene bool'sche Wert.

4.5.69 Read Boolean

Liest einen bool'schen Wert aus einer SPS ein.



Ein bool'scher Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

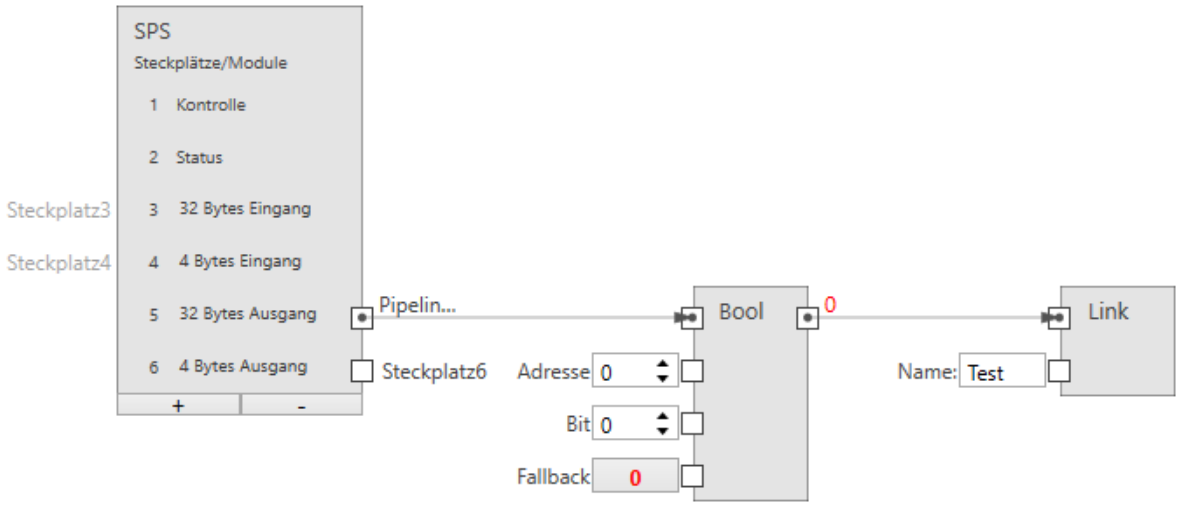


Abb. 172: Ein bool'scher Eingang.

Die Größe eines bool'schen Eingangs beträgt 1 Bit (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Eingänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Bit (Type: `Byte`)

Gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Fallback (Type: `Boolean`)

Gibt den bool'schen Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

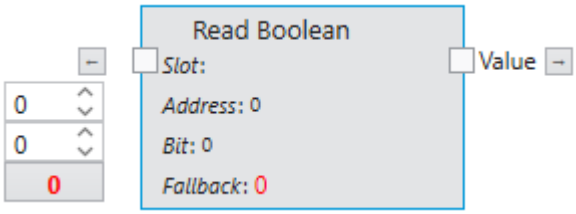
Ausgänge

Value (Type: `Boolean`)

Der über den Feldbus eingelesene bool'sche Wert.

4.5.70 Read Boolean

Liest einen bool'schen Wert aus einer SPS ein.



Ein bool'scher Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

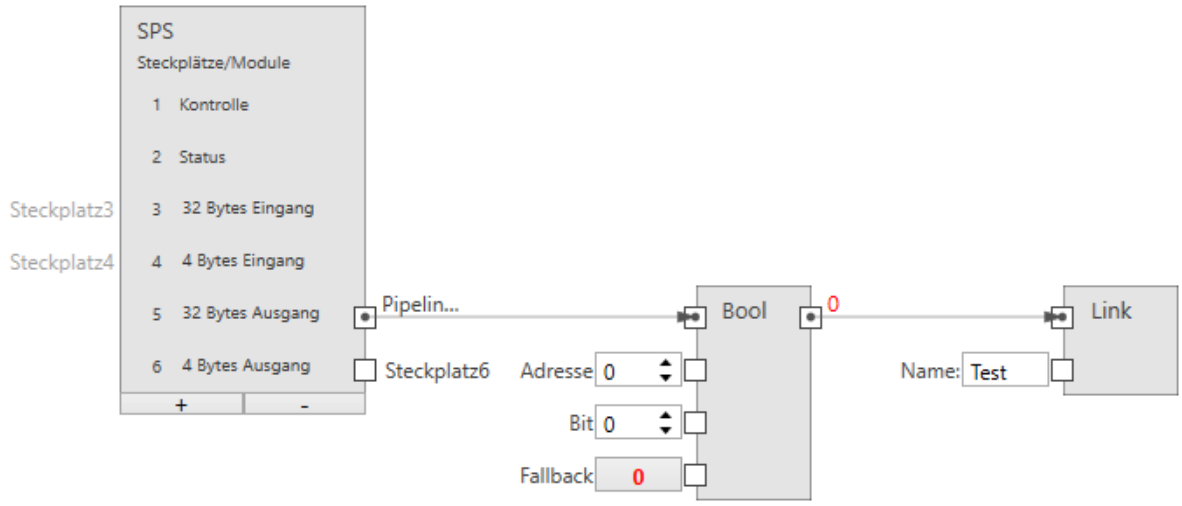


Abb. 173: Ein bool'scher Eingang.

Die Größe eines bool'schen Eingangs beträgt 1 Bit (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Eingänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Bit (Type: `Byte`)

Gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Fallback (Type: `Boolean`)

Gibt den bool'schen Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

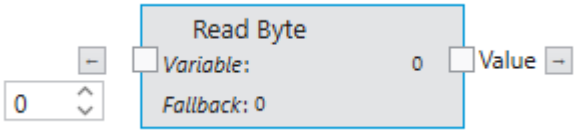
Ausgänge

Value (Type: `Boolean`)

Der über den Feldbus eingelesene bool'sche Wert.

4.5.71 Read Byte

Liest einen Byte Wert aus einer TCP/IP Eingangs-Variablen ein.



Ein Byte Eingang wird mit einem TCP/IP Eingang verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

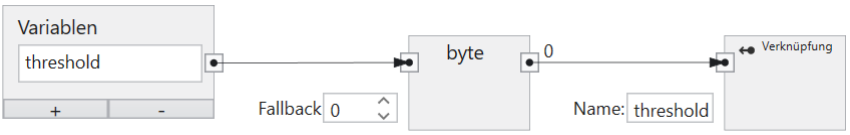


Abb. 174: Ein Byte Eingang.

Eingänge

Variable (Type: `TcpIpVariable`)

Verbindet den Knoten mit einer TCP/IP Eingangs-Variablen (TCP/IP -> **CS-6x/ID-6x0**).

Fallback (Type: `Int32`)

Gibt den Byte Wert an, der verwendet wird, wenn die TCP/IP-Daten nicht verfügbar sind. TCP/IP-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil der Kommando-Server am Sensor deaktiviert ist, - weil die TCP/IP Eingangs Variable nicht gesetzt wurde.

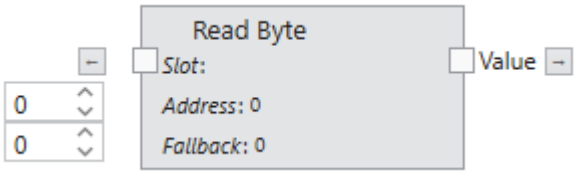
Ausgänge

Value (Type: `Byte`)

Der über TCP/IP eingelesene Byte Wert.

4.5.72 Read Byte

Liest einen Byte Wert aus einer SPS ein.



Ein Byte Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

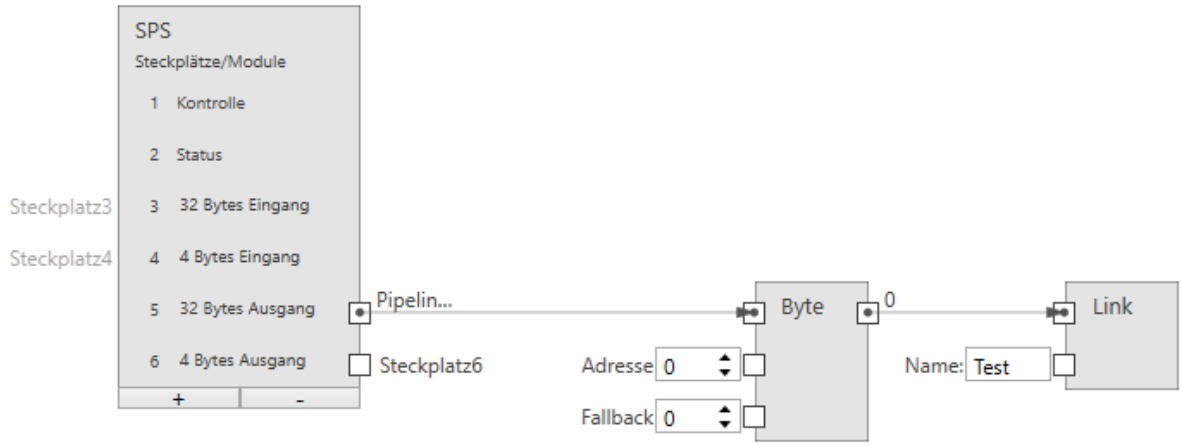


Abb. 175: Ein Byte Eingang.

Eingänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Addrese (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Fallback (Type: `Byte`)

Gibt den Byte Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Ausgänge

Value (Type: `Byte`)

Der über den Feldbus eingelesene Byte Wert.

4.5.73 Read Byte

Liest einen Byte Wert aus einer SPS ein.



Ein Byte Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

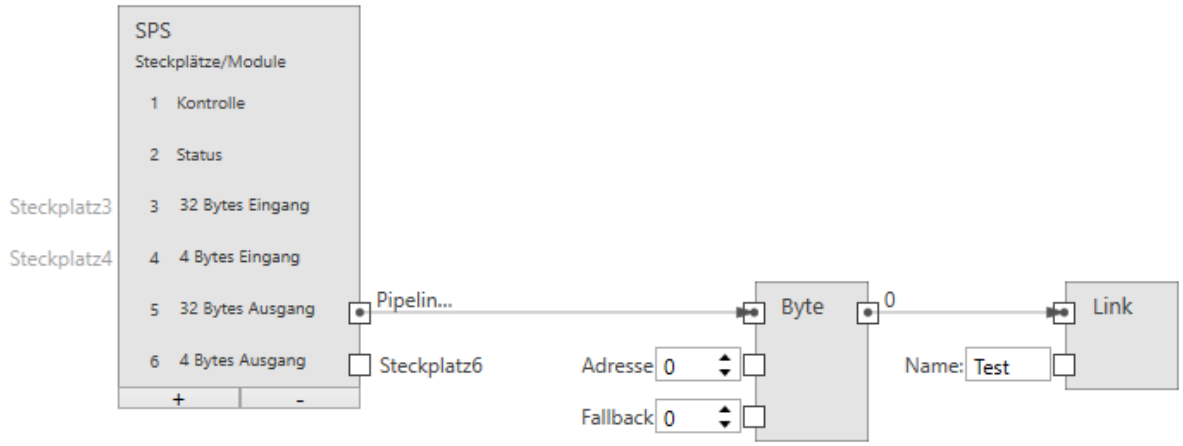


Abb. 176: Ein Byte Eingang.

Eingänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Addres (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Fallback (Type: `Byte`)

Gibt den Byte Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

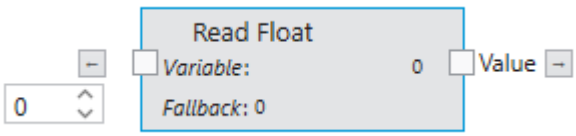
Ausgänge

Value (Type: `Byte`)

Der über den Feldbus eingelesene Byte Wert.

4.5.74 Read Float

Liest einen Fließkomma-Wert aus einer TCP/IP Eingangs-Variablen ein.



Ein Float Eingang wird mit einem TCP/IP Eingang verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

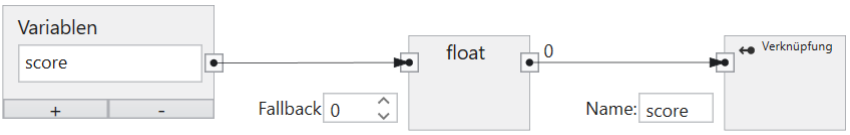


Abb. 177: Ein Fließkomma Eingang.

Eingänge

Variable (Type: `TcpIpVariable`)

Verbindet den Knoten mit einer TCP/IP Eingangs-Variablen (TCP/IP -> **CS-6x/ID-6x0**).

Fallback (Type: `Single`)

Gibt den Float Wert an, der verwendet wird, wenn die TCP/IP-Daten nicht verfügbar sind. TCP/IP-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil der Kommando-Server am Sensor deaktiviert ist, - weil die TCP/IP Eingangs Variable nicht gesetzt wurde.

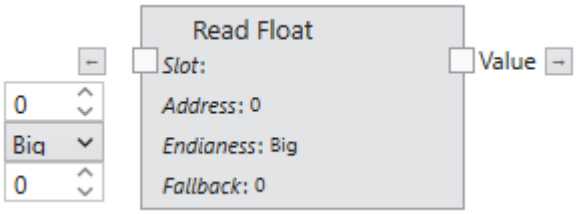
Ausgänge

Value (Type: `Single`)

Der über TCP/IP eingelesene Fließkomma Wert.

4.5.75 Read Float

Liest einen Fließkomma-Wert aus einer SPS ein.



Ein Float Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

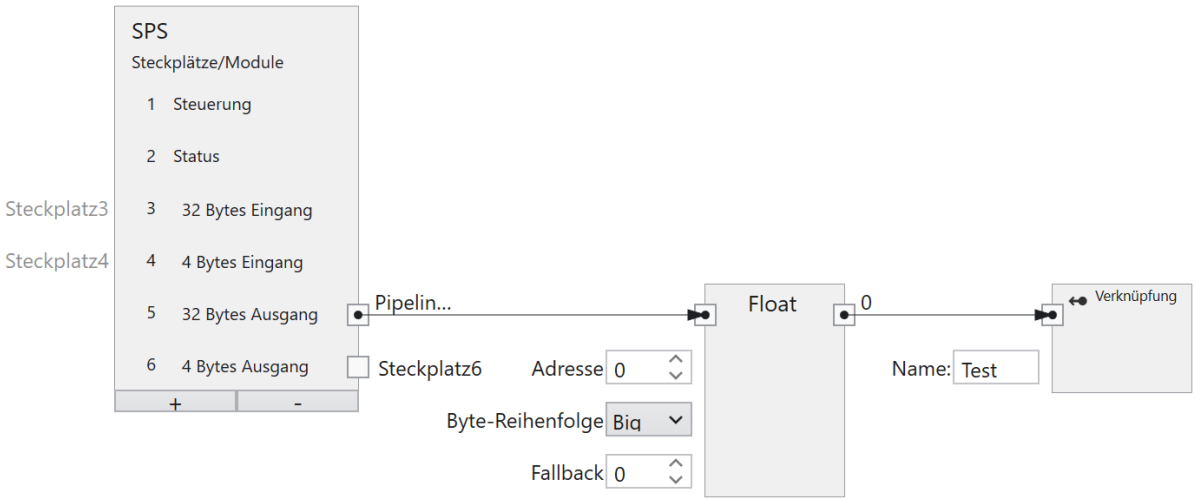


Abb. 178: Ein Fließkomma Eingang.

Die Größe eines Fließkomma Eingangs beträgt 4 Byte.

Eingänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Single`)

Gibt den Float Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

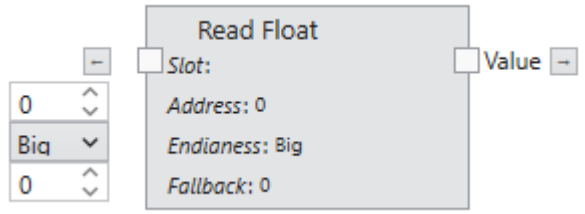
Ausgänge

Value (Type: `Single`)

Der über den Feldbus eingelesene Fließkomma Wert.

4.5.76 Read Float

Liest einen Fließkomma-Wert aus einer SPS ein.



Ein Float Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

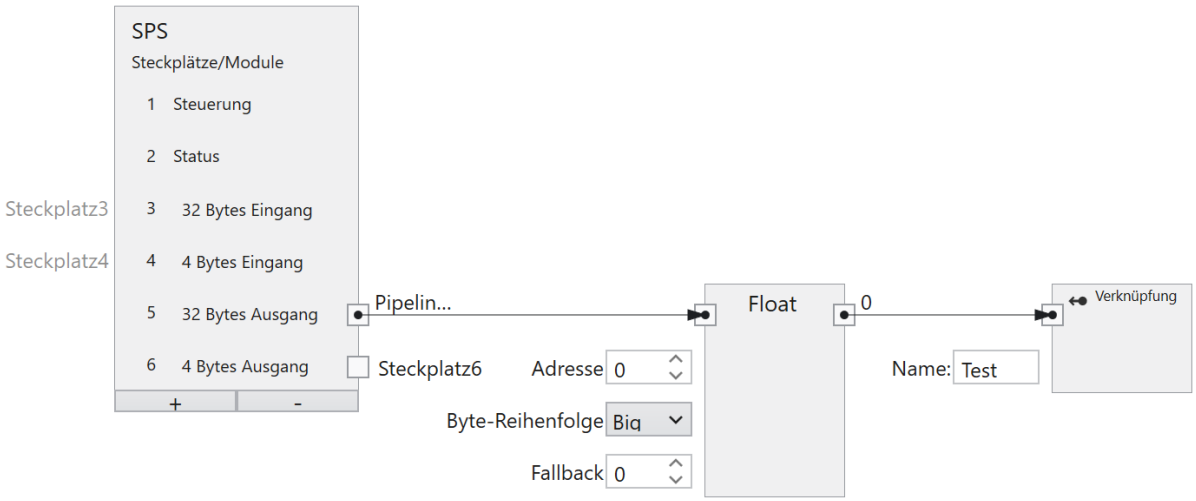


Abb. 179: Ein Fließkomma Eingang.

Die Größe eines Fließkomma Eingangs beträgt 4 Byte.

Eingänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Single`)

Gibt den Float Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Ausgänge

Value (Type: `Single`)

Der über den Feldbus eingelesene Fließkomma Wert.

4.5.77 Read Int32

Liest einen ganzzahligen Wert aus einer TCP/IP Eingangs-Variablen ein.



Ein Int32 Eingang wird mit einem TCP/IP Eingang verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

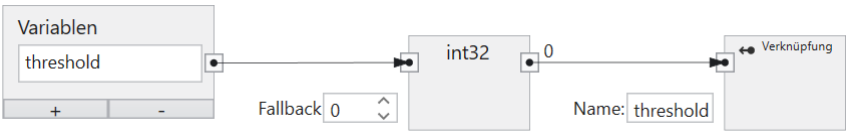


Abb. 180: Ein ganzzahliger Eingang.

Eingänge

Variable (Type: `TcpIpVariable`)

Verbindet den Knoten mit einer TCP/IP Eingangs-Variablen (TCP/IP -> **CS-6x/ID-6x0**).

Fallback (Type: `Int32`)

Gibt den Int32 Wert an, der verwendet wird, wenn die TCP/IP-Daten nicht verfügbar sind. TCP/IP-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil der Kommando-Server am Sensor deaktiviert ist, - weil die TCP/IP Eingangs Variable nicht gesetzt wurde.

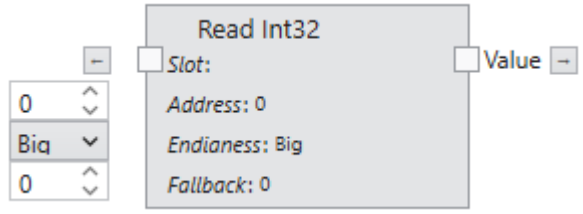
Ausgänge

Value (Type: `Int32`)

Der über TCP/IP eingelesene ganzzahlige Int32 Wert.

4.5.78 Read Int32

Liest einen ganzzahligen Wert aus einer SPS ein.



Ein Int32 Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

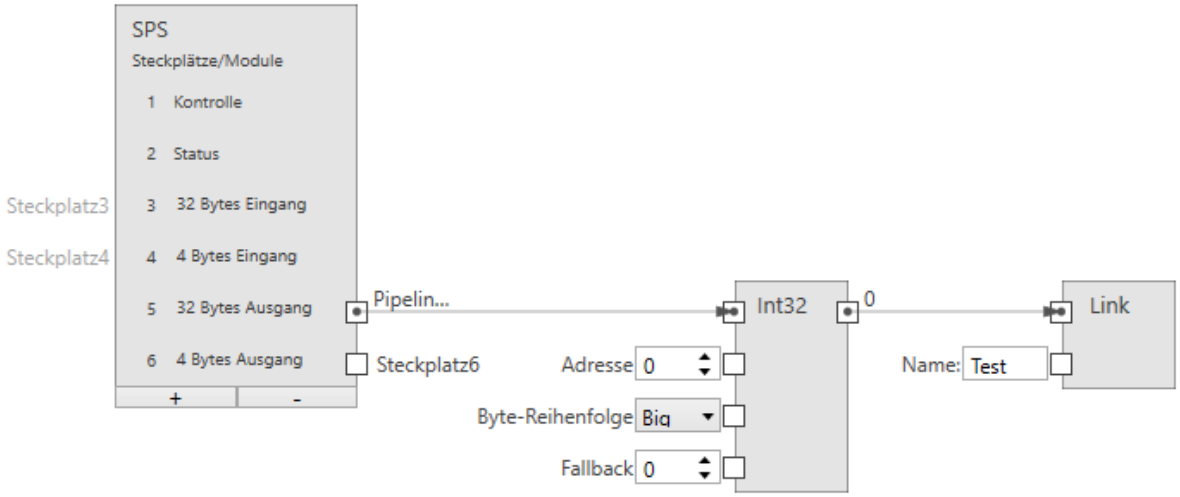


Abb. 181: Ein ganzzahliger Eingang.

Die Größe eines ganzzahligen Eingangs beträgt 4 Byte.

Eingänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Byte-Reihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den Zahlenwert Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

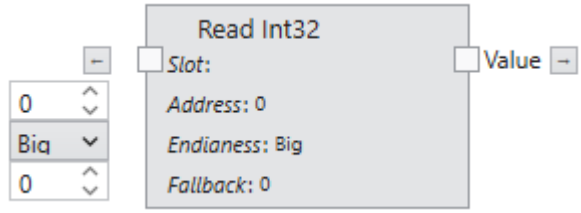
Ausgänge

Value (Type: `Int32`)

Der über den Feldbus eingelesene ganzzahlige Int32 Wert.

4.5.79 Read Int32

Liest einen ganzzahligen Wert aus einer SPS ein.



Ein Int32 Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

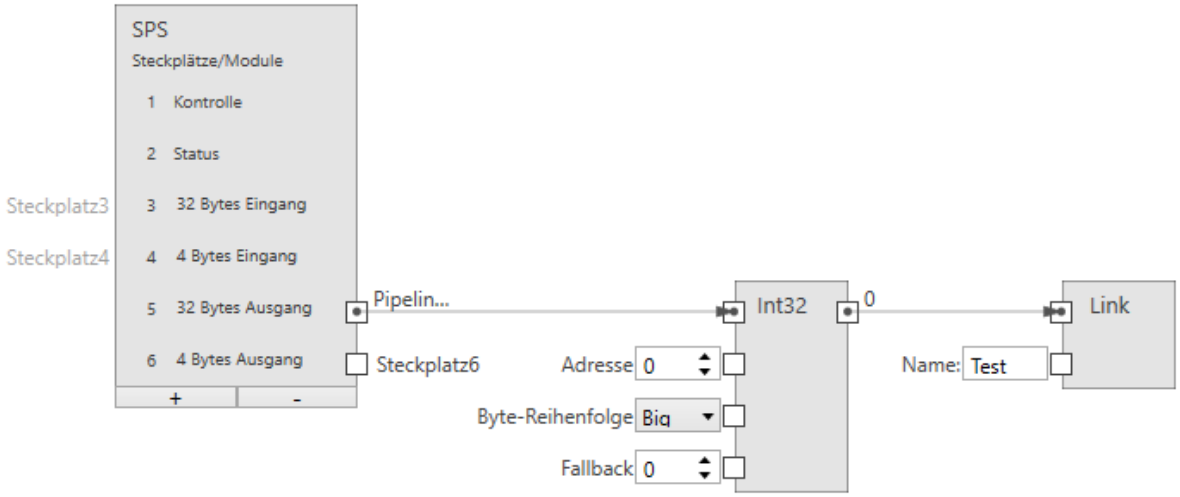


Abb. 182: Ein ganzzahliger Eingang.

Die Größe eines ganzzahligen Eingangs beträgt 4 Byte.

Eingänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Byte-Reihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den Zahlenwert Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

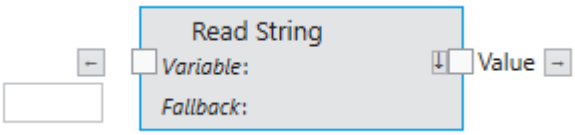
Ausgänge

Value (Type: `Int32`)

Der über den Feldbus eingelesene ganzzahlige Int32 Wert.

4.5.80 Read String

Liest einen Text aus einer TCP/IP Eingangs-Variablen ein.



Ein String Eingang wird mit einem TCP/IP Eingang verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

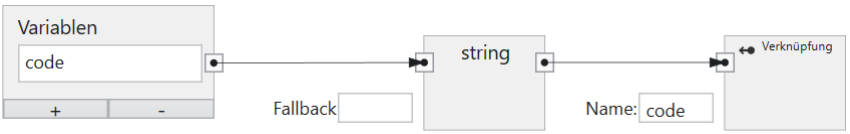


Abb. 183: Ein Text Eingang.

Eingänge

Variable (Type: `TcpIpVariable`)

Verbindet den Knoten mit einer TCP/IP Eingangs-Variablen (TCP/IP -> **CS-6x/ID-6x0**).

Fallback (Type: `String`)

Gibt den String Wert an, der verwendet wird, wenn die TCP/IP-Daten nicht verfügbar sind. TCP/IP-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil der Kommando-Server am Sensor deaktiviert ist, - weil die TCP/IP Eingangs Variable nicht gesetzt wurde.

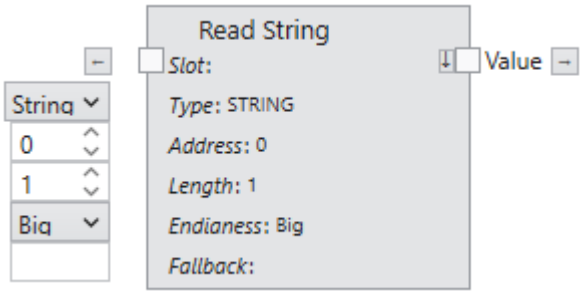
Ausgänge

Value (Type: `String`)

Der über TCP/IP eingelesene Text.

4.5.81 Read String

Liest einen Text aus einer SPS ein.



Ein String Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

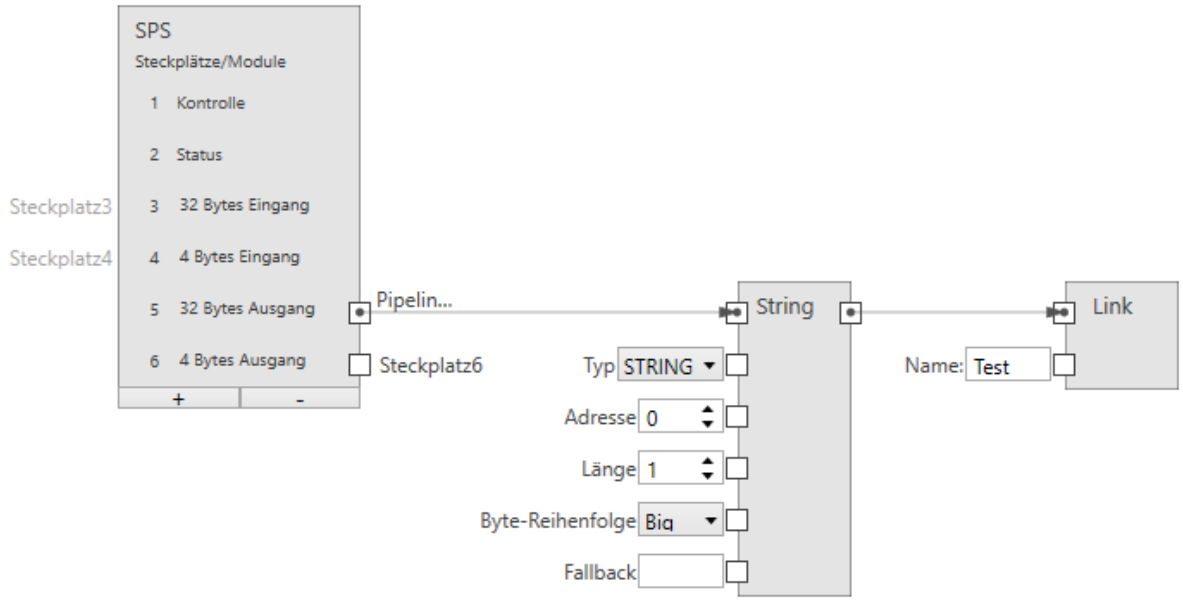


Abb. 184: Ein Text Eingang.

Die Größe eines Texts beträgt (2 + n) * s, wobei n die maximale Anzahl von Zeichen ist und s = 1 (für einen Ansi STRING) oder s = 2 (für einen Unicode-WSTRING). Das erste Byte/Wort (Ansi/Unicode) enthält die maximale Anzahl von Zeichen, welche die Zeichenfolge enthalten kann. Das zweite Byte/Wort (Ansi/Unicode) enthält die tatsächliche Anzahl der Zeichen, welche die Zeichenfolge enthält.

Eingänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Type (Type: `String`)

Gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Length (Type: `UInt32`)

Gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den String Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

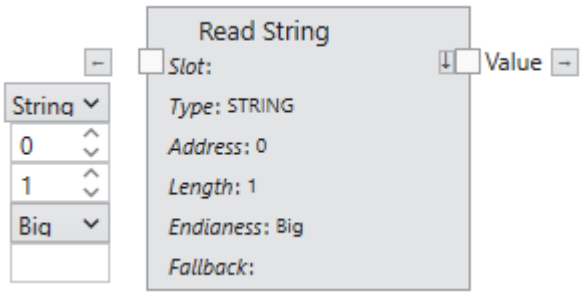
Ausgänge

Value (Type: `String`)

Der über den Feldbus eingelesene Text.

4.5.82 Read String

Liest einen Text aus einer SPS ein.



Ein String Eingang wird mit einem Feldbus Ausgangs-Steckplatz verbunden und anschließend über einen Link zur Verwendung in nachfolgenden Werkzeugen benannt.

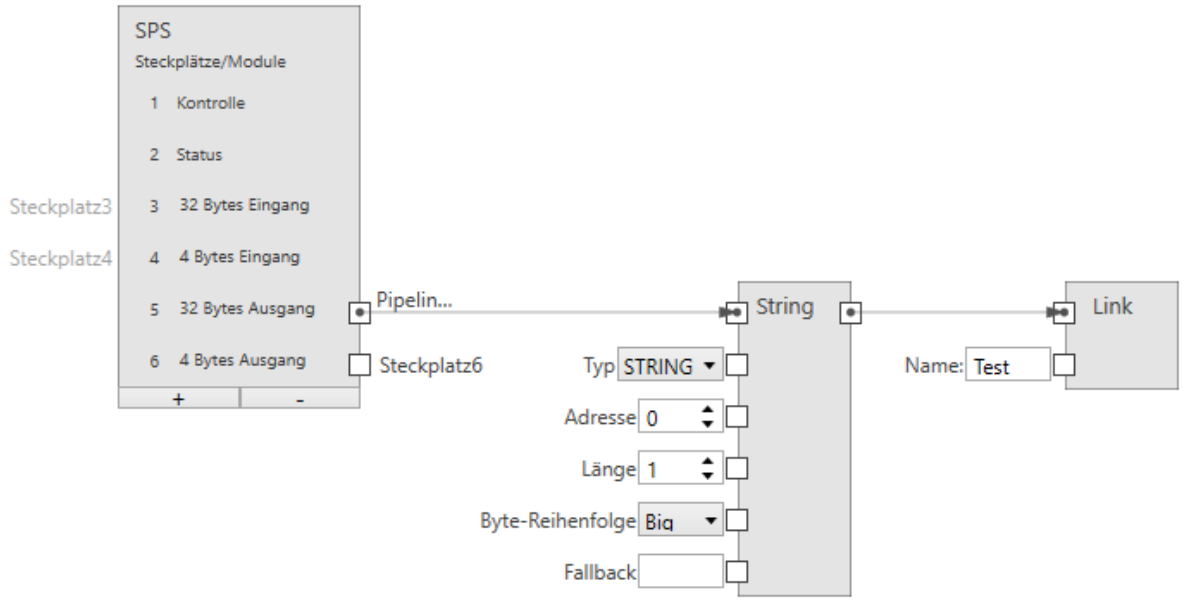


Abb. 185: Ein Text Eingang.

Die Größe eines Texts beträgt (4 + n), wobei n die maximale Anzahl von Zeichen ist. Die ersten 4 Byte werden als Int32 interpretiert und stellen die tatsächliche Anzahl der Zeichen des Strings da.

Eingänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (SPS -> **CS-6x/ID-6x0**).

Type (Type: `String`)

Gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Length (Type: `UInt32`)

Gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den String Wert an, der verwendet wird, wenn keine Feldbus-Daten verfügbar sind. Feldbus-Daten sind möglicherweise aus mehreren Gründen nicht verfügbar: - weil PROFINET oder EtherNet/IP am Sensor deaktiviert ist, - weil der Job im Emulator läuft usw.

Ausgänge

Value (Type: `String`)

Der über den Feldbus eingelesene Text.

4.5.83 Reverse

Kehrt die Reihenfolge der Elemente in einer Liste um.



Eine Liste besteht aus mehreren, aneinandergereihten Elementen. Dieser Knoten kehrt die Reihenfolge der Listenelemente um.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

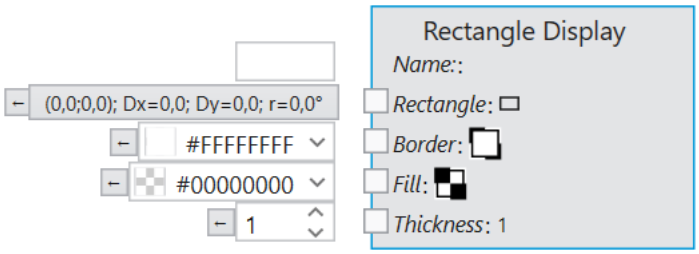
Ausgänge

Reverse (Type: `List<T>`)

Die Liste mit den Elementen in umgekehrter Reihenfolge.

4.5.84 Rectangle Display

Erzeugt ein visuelle Darstellung eines Rechtecks.



Das Rechteck wird auf dem Smart Tool Display und auf der HTML Visualisierung angezeigt.

Eingänge

Name (Type: `String`)

Der Name des Rechtecks, das auf der HTML Visualisierung angezeigt wird, um das Rechteck ein bzw. auszuschalten.

Position (Type: `Rectangle`)

Die Position des Rechtecks.

Border (Type: `SolidColorBrushByte`)

Die Farbe des Umrisses. Standardmäßig ist die Farbe weiß.

Fill (Type: `SolidColorBrushByte`)

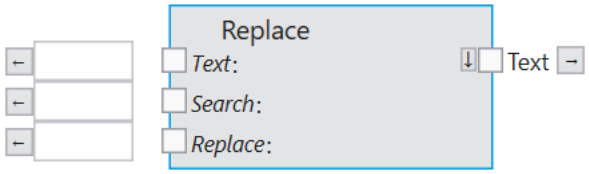
Die Farbe der Füllung. Standardmäßig ist die Farbe durchsichtig.

Thickness (Type: `Double`)

Die Dicke der Umrisslinie.

4.5.85 Replace

Ersetzt einen Such-Text in einem Text.



Der Such-Text wird in einem Text gesucht und durch den Ersetzungs-Text ersetzt.

Eingänge

Text (Type: `String`)

Der Text in dem der Such-Text gesucht wird und die Ersetzung vorgenommen wird.

Search (Type: `String`)

Der Such-Text.

Es werden folgende Escape Sequenzen verwendet:

- `\\` -> (Backslash)
- `\r` -> (Wagenrücklauf)
- `\n` -> (Zeilenvorschub)

- `\t` -> (Tabulator)
- `\x##` -> ASCII Code ##, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht
- `\u####` -> UTF16 Code ####, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht

Replace (Type: `String`)

Der Ersetzungs-Text.

Es werden folgende Escape Sequenzen verwendet:

- `\\` -> (Backslash)
- `\r` -> (Wagenrücklauf)
- `\n` -> (Zeilenvorschub)
- `\t` -> (Tabulator)
- `\x##` -> ASCII Code ##, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht
- `\u####` -> UTF16 Code ####, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht

Ignore case (Type: `Boolean`)

Die Groß- bzw. Kleinschreibung wird ignoriert, wenn dieser Wert auf True/1 gesetzt ist.

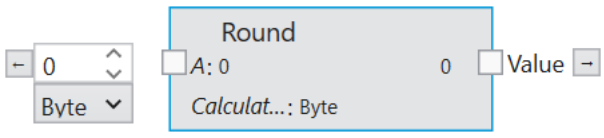
Ausgänge

Text (Type: `String`)

Der Text, in dem der Such-Text durch den Ersetzungs-Text ersetzt worden ist.

4.5.86 Round

Rundet eine Fließkomma-Zahl auf eine ganze Zahl.



Die Rundung erfolgt auf einen spezifizierten Datentyp.

Eingänge

A (Type: `Double`)

Die zu rundende Zahl. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

Calculate as (Type: `String`)

Die Berechnung erfolgt im spezifizierten Datentyp: **Byte**, **Int16**, **Int32**, **Int64**, **Float** oder **Double**.

- Byte erlaubt ganzzahlige Zahlenwerte zwischen 0 und 255.
- Int16 erlaubt ganzzahlige Zahlenwerte zwischen -32.768 und 32.767.
- Int32 erlaubt ganzzahlige Zahlenwerte zwischen -2.147.483.648 und 2.147.483.647.
- Int64 erlaubt ganzzahlige Zahlenwerte zwischen -9.223.327.036.854.775.808 und 9.223.372.036.854.775.807.
- Float erlaubt Fließkommazahlen zwischen +/-1,5 x 10^-45 und +/-3,4 x 10^38 mit einer Präzision von 6-9 Stellen.
- Double erlaubt Fließkommazahlen zwischen +/-5,0 x 10^-324 und +/-1,7 x 10^308 mit einer Präzision von 15-17 Stellen.

Ausgänge

Value (Type: `Byte|Int16|Int32|Int64|Single|Double`)

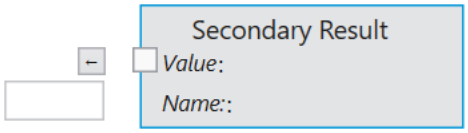
Die gerundete Zahl, entsprechend dem spezifizierten Datentyp.

Sobald der Ausgäng verbunden ist, ist der Typ festgelegt und kann nicht mehr geändert werden. Zum Ändern des Typs muss die Verknüpfung des Ausgangs gelöscht werden.

Wenn das Ergebnis im spezifizierten Datentype nicht darstellbar ist, produziert der Knoten kein numerisches Ergebnis.

4.5.87 Secondary Result

Erzeugt ein sekundäres Resultat für ein Smart Tool.



Das Smart Tool kann verschiedene Resultate erzeugen.

- Das primäre Ergebnis ist ein bool'scher Wert, der anzeigt, ob das Werkzeug Ergebnis in Ordnung ist (`1`) oder ein Fehler auftrat (`0`).
- Sekundäre Ergebnisse können Messwerte sein, die prominent im Werkzeug angezeigt werden.
- Weitere Ergebnisse können Messwerte sein, die im Werkzeug nur angezeigt werden, wenn sie ausgeklappt werden.
- Eine Pose wird benötigt, um eine spezifische Nachführung zu erstellen.

Eingänge

Value (Type: `Object`)

Ein sekundäres Werkzeug Ergebnis.

Name (Type: `String`)

Die Bezeichnung des Ergebnisses.

4.5.88 Select GS1 AI

Wählt einen einzelnen Datenbezeichner (Application Identifier) aus einer Liste aus.

Die GS1-Datenbezeichner sind im GS1 General Specifications Standard festgelegt. Die unterstützte Version ist Release 25.0. Die Spezifikation ist hier zu finden: https://www.gs1.org/standards/barcodes-epcrfid-id-keys/gs1-general-specifications

Inputs

AIs (Typ: `IList<Gs1ApplicationId>`)

Die Liste der Datenbezeichner.

AI/Title (Typ: `string`)

Der auszuwählende Datenbezeichner. Dies kann entweder die numerische Form (aber immer noch als Text, wie "10") oder der Titel (wie "BATCH/LOT") sein. Wenn der Titel verwendet wird, kann er groß- oder kleingeschrieben werden.

Outputs

Text (Typ: `string`)

Der Text des ausgewählten Datenbezeichners.

Value (Typ: `double`)

Der Wert des ausgewählten Datenbezeichners.

Unit (Typ: `string`)

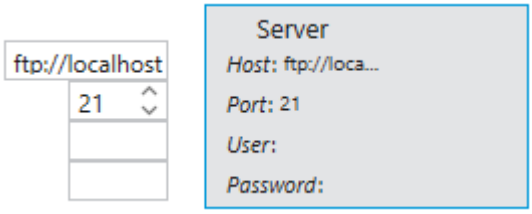
Die Einheit des ausgewählten Datenbezeichners.

Type (Typ: `string`)

Der Typ des ausgewählten Datenbezeichners. Der Typ gibt an, ob der Text oder der Wert vom Datenbezeichner verwendet wird.

4.5.89 Server

Konfiguriert die FTP Server Einstellungen.



Eingänge

Host (Type: `String`)

Der FTP oder SFTP Host, z.B. `ftp://192.168.3.15` oder `sftp://192.168.3.15`. Die IP Adressen müssen entsprechend auf die Gegebenheiten angepasst werden.

Port (Type: `Int32`)

Der FTP oder SFTP Port, z.B. `21` oder `22`.

User (Type: `String`)

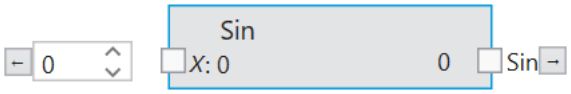
Der Benutzername zum Einloggen in den FTP oder SFTP Server.

Password (Type: `String`)

Das Passwort zum Einloggen in den FTP oder SFTP Server.

4.5.90 Sin

Berechnet den Sinus eines Winkels.



Eingänge

X (Type: `Double`)

Das Argument der Sinus Funktion, im Bogenmaß.

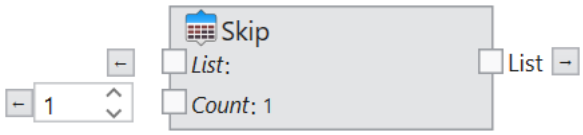
Ausgänge

Sin (Type: `Double`)

Der Sinus des Winkels.

4.5.91 Skip

Überspringt eine festgelegte Anzahl von Elementen in einer Sequenz und gibt dann die übrigen Elemente zurück.



Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

Count (Type: `Int32`)

Die gewünschte Anzahl der Elemente, die übersprungen werden sollen.

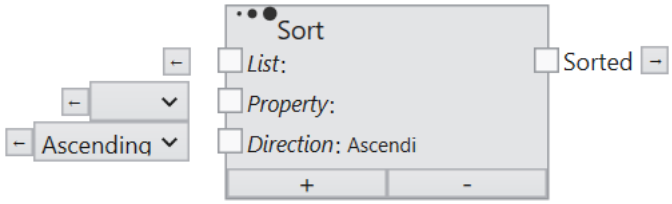
Ausgänge

List (Type: `List<T>`)

Die Liste mit den gewünschten Elementen.

4.5.92 Sort

Sortiert die Elemente in einer Liste.



Mit den **+** und **-** Tasten können paarweise Eingänge (`PropertyN`, `DirectionN`) am Ende hinzugefügt oder vom Ende gelöscht werden.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

Property[1..N] (Type: `String`)

Die Eigenschaft, nach der sortiert werden soll. Es können die Properties des Typs `T` gewählt werden, oder `Self` um nach `T` selbst zu sortieren

Direction[1..N] (Type: `String`)

Ascending sortiert die Werte aufsteigend, **Descending** sortiert absteigend.

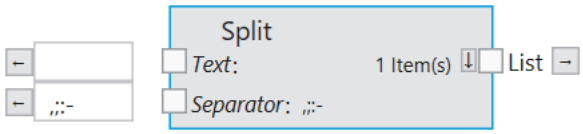
Ausgänge

Sorted (Type: `List<T>`)

Die sortierte Liste.

4.5.93 Split

Trennt einen Text an bestimmten Trennzeichen.



Ein Text wird an bestimmten Trennzeichen getrennt und in eine Liste von Texten umgewandelt.

Eingänge

Text (Type: `String`)

Der aufzutrennende Text.

Separator (Type: `String`)

Enthält die Trennzeichen. An jedem der Trennzeichen wird eine Trennung vorgenommen.

Es werden folgende Escape Sequenzen verwendet:

- `\\` -> (Backslash)
- `\r` -> (Wagenrücklauf)
- `\n` -> (Zeilenvorschub)
- `\t` -> (Tabulator)
- `\x##` -> ASCII Code ##, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht
- `\u####` -> UTF16 Code ####, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht

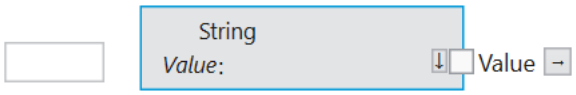
Ausgänge

List (Type: `List<String>`)

Eine Liste von Texten.

4.5.94 String

Definiert einen Text.



Eingänge

Value (Type: `String`)

Mit dem Editor am Eingang *Value* kann der Text eingegeben werden.

Es werden folgende Escape Sequenzen verwendet:

- `\\` -> (Backslash)
- `\r` -> (Wagenrücklauf)
- `\n` -> (Zeilenvorschub)
- `\t` -> (Tabulator)
- `\x##` -> ASCII Code ##, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht
- `\u####` -> UTF16 Code ####, wobei # für Zahlen von 0 bis 9 und Buchstaben von A bis F steht

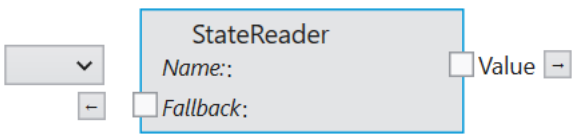
Ausgänge

Value (Type: `String`)

Der Text.

4.5.95 StateReader

Liest einen Zustandswert aus dem Zustandsspeicher.



Eingänge

Name (Type: `String`)

Der Name für das Element im Zustandsspeicher.

Fallback (Type: `Object`)

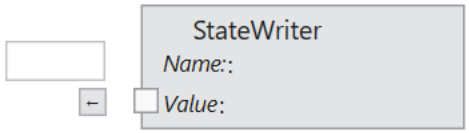
Dieser Wert wird verwendet, falls es im Zustandsspeicher keinen Wert gibt.

Kommentare

Der Zustandsspeicher wird am Ende einer Inspektion geschrieben. Er kann damit erst in darauffolgenden Inspektionen gelesen werden.

4.5.96 StateWriter

Schreibt einen Zustandswert in den Zustandsspeicher.



Der Zustandsspeicher kann in darauffolgenden Inspektionen gelesen werden.

Eingänge

Name (Type: `String`)

Der Name für das Element im Zustandsspeicher.

Value (Type: `Object`)

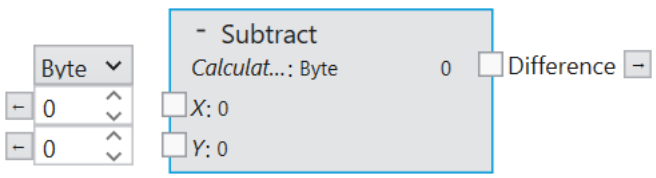
Dieser Wert wird in den Zustandsspeicher geschrieben.

Kommentare

Der Zustandsspeicher wird am Ende einer Inspektion geschrieben. Er kann damit erst in darauffolgenden Inspektionen gelesen werden.

4.5.97 Subtract

Subtrahiert zwei Zahlen.



Die Subtraktion erfolgt mit einem spezifizierten Datentyp.

Eingänge

Calculate as (Type: `String`)

Die Berechnung erfolgt im spezifizierten Datentyp: **Byte**, **Int16**, **Int32**, **Int64**, **Float** oder **Double**.

- Byte erlaubt ganzzahlige Zahlenwerte zwischen 0 und 255.
- Int16 erlaubt ganzzahlige Zahlenwerte zwischen -32.768 und 32.767.
- Int32 erlaubt ganzzahlige Zahlenwerte zwischen -2.147.483.648 und 2.147.483.647.
- Int64 erlaubt ganzzahlige Zahlenwerte zwischen -9.223.327.036.854.775.808 und 9.223.372.036.854.775.807.
- Float erlaubt Fließkommazahlen zwischen +/-1,5 x 10^-45 und +/-3,4 x 10^38 mit einer Präzision von 6-9 Stellen.
- Double erlaubt Fließkommazahlen zwischen +/-5,0 x 10^-324 und +/-1,7 x 10^308 mit einer Präzision von 15-17 Stellen.

X (Type: `Double`)

Der Minuend. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

Y (Type: `Double`)

Der Subtrahend. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

Ausgänge

Difference (Type: `Byte|Int16|Int32|Int64|Single|Double`)

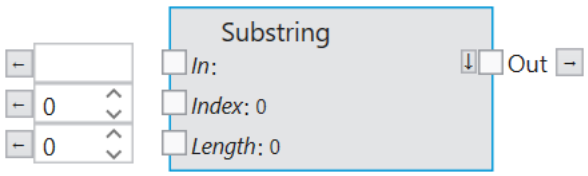
Die Differenz der Zahlen, entsprechend dem spezifizierten Datentyp.

Sobald der Ausgäng verbunden ist, ist der Typ festgelegt und kann nicht mehr geändert werden. Zum Ändern des Typs muss die Verknüpfung des Ausgangs gelöscht werden.

Wenn das Ergebnis im spezifizierten Datentype nicht darstellbar ist, produziert der Knoten kein numerisches Ergebnis.

4.5.98 Substring

Liefert eine Teilzeichenfolge.



Eingänge

In (Type: `String`)

Der Eingabetext.

Index (Type: `Int32`)

Der Beginn der Teilzeichenfolge im Eingabetext, 0-basiert.

Length (Type: `Int32`)

Die Länge der Teilzeichenfolge.

Ausgänge

Out (Type: `Boolean`)

Die Teilzeichenfolge.

4.5.99 Take

Gibt eine angegebene Anzahl von zusammenhängenden Elementen ab dem Anfang einer Liste zurück.



Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

Count (Type: `Int32`)

Die gewünschte Anzahl der Elemente in der Ergebnisliste.

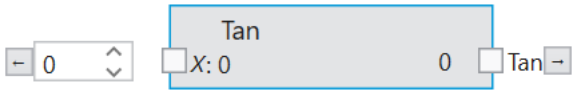
Ausgänge

List (Type: `List<T>`)

Die Liste mit den gewünschten Elementen.

4.5.100 Tan

Berechnet den Tangens eines Winkels.



Eingänge

X (Type: `Double`)

Das Argument der Tangens Funktion, im Bogenmaß.

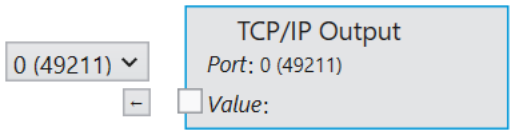
Ausgänge

Tan (Type: `Double`)

Der Tangens des Winkels.

4.5.101 TCP/IP Output

Konfiguration der TCP/IP Ausgänge.



Der **TCP/IP Output** Knoten wird verwendet, um Ausgaben auf einem TCP/IP Kanal zu definieren.

Eingänge

Port (Typ: `String`)

Der für den Ausgangs-Strom verwendete TCP/IP Port. Es können bis zu vier Ports verwendet werden, je nach der vorgewählten Anzahl in den **Protokoll Einstellungen**. Die Port Nummern sind vordefiniert von 49211 bis 49214.

Wert (Typ: `String`)

Der Ausgabewert, der nach jeder Bildauswertung einmal auf dem TCP/IP Port ausgegeben wird.

Werkzeuge können aus der Werkzeugliste gewählt werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die mit dem TCP/IP **TCP/IP Output** Knoten verbunden werden können. Hier ist ein Beispiel:

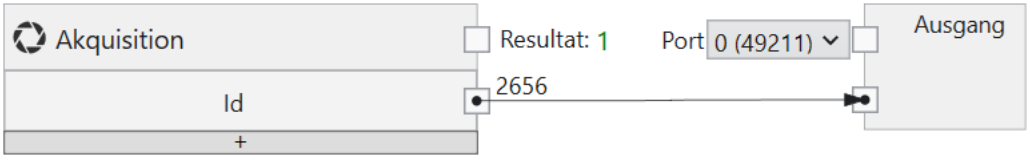


Abb. 186: Ein Werkzeug Knoten für einen TCP/IP Ausgangs-Strom.

Neben logischen und arithmetischen Kombinationen ist auch eine Kombination von Texten möglich, mit dem **Text** Knoten, der durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

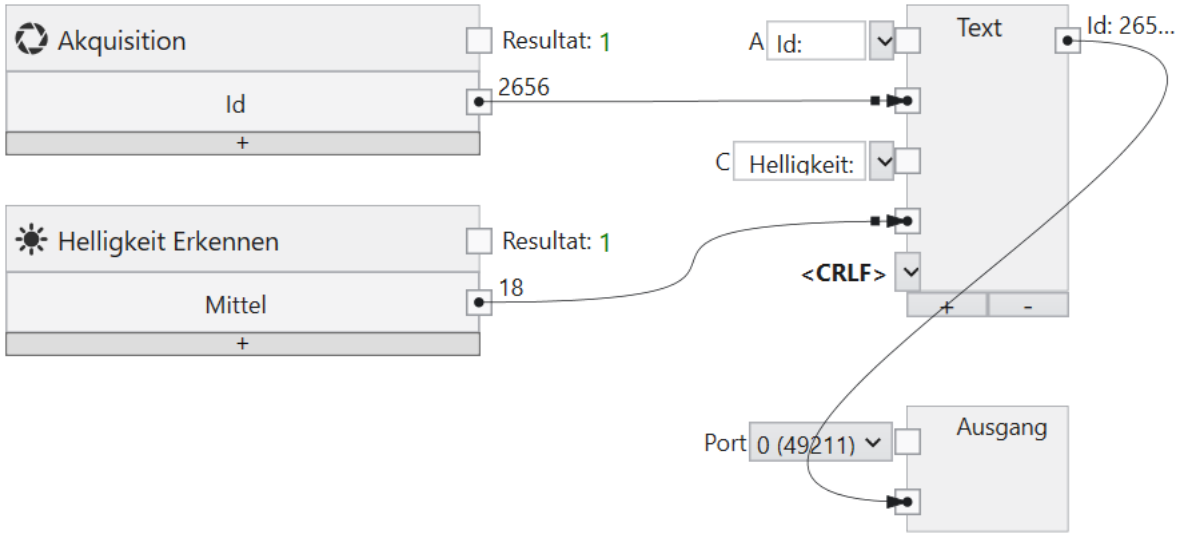


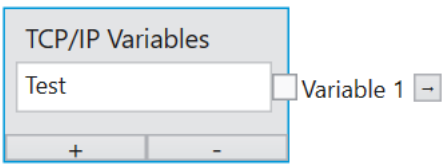
Abb. 187: Text-Kombination für einen TCP/IP Ausgangs-Strom.

Ein Knoten auf der Leinwand oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

Ist der auszugebende Wert nicht verfügbar, wird der Text **UNDEFINED** ausgegeben. Mit dem **Fallback Value**-Knoten kann ein alternatives Verhalten für fehlende Werte definiert werden.

4.5.102 TCP/IP Variables

Repräsentiert die Konfiguration der TCP/IP Kommando Eingänge.



Der **TCP/IP Variables** Knoten wird verwendet, um Variablen zu definieren. Der **TCP/IP Variables** Knoten wird im Akquisition Werkzeug angezeigt. Durch die **+** und **-** Tasten können Variablen hinzugefügt oder gelöscht werden.

Viele Werkzeugparameter können durch Auswahl oder Eingabe von Werten geändert werden. Die meisten dieser Parameter können auch bearbeitet werden durch **Verknüpfen** mit TCP/IP Kommando Eingängen.

Das Verknüpfen eines Werts erfolgt in mehreren Schritten auf der Registerkarte **TCP/IP Variablen Eingänge** des Werkzeugs **Akquisition**:

1. Eine TCP/IP Variable wird mit einem Namen gekennzeichnet und mit einem Lese-Knoten verbunden.
2. Der Ausgang des Lese-Knotens wird mit einem Verknüpfungs-Knoten verbunden.
3. Der Verknüpfungs-Knoten definiert einen Namen, unter dem benannte Wert mit der Taste als Parameter verwendet werden.

Die Namen unter Punkt 1 und 3 können unterschiedlich sein, bezeichnen aber letztlich das selbe Objekt. Unter Punkt 1 wird der Name definiert, mit dem der Wert vom TCP/IP Kommando `set name=value` identifiziert wird. Unter Punkt 3 wird de Name definiert, unter dem der Wert mit den Links im Job identifiziert wird.

Eingabedaten werden immer dann erfasst, wenn ein Bild erfasst wird. Dies geschieht zu Beginn eines Inspektionszyklus.

Verknüpfen von TCP/IP Kommando Eingängen

Der **TCP/IP Variables** Knoten wird verwendet, um Variablen im TCP/IP Namensraum zu benennen.

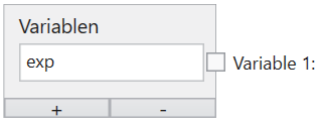


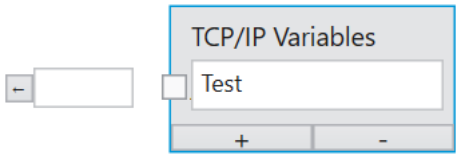
Abb. 188: Der Inputs Knoten mit einer Variablen.

Beispielhaft ist hier eine Variable definiert. Der Name muss mit einem Buchstaben beginnen und kann optional gefolgt sein von einem oder mehreren Buchstaben, Ziffern oder _ (Unterstrich). Dieser Name ist gültig im TCP/IP Namensraum, d.h. bei Verwendung mit den TCP/IP Kommando `set name=value`. Beispielsweise würde das Kommando `set exp=100` den Wert `100` der Variablen `exp` zuweisen.

TCP/IP Kommando Variablen können Eingabewerte verschiedener Typen bereitstellen: Logisch (Bool), Numerisch (Byte, Int32, Float) und Text (String). Um sie zum Verlinken zu verwenden, müssen sie definiert und benannt werden. Ein Eingang wird definiert, indem ein Leseknoten des gewünschten Typs hinzugefügt wird (mit einem Rechtsklick auf Canvas und Auswahl von **Read Boolean**, **Read Byte**, **Read Int32**, **Read Float** oder **Read String**) und mit einem bestimmten Modul in einem Steckplatz verbunden wird, der im Inputs Knoten definiert ist. Ein Eingang wird benannt durch Hinzufügen eines **Verknüpfung** Knotens (Auswahl im Menü links oder mit einem Rechtsklick auf die Zeichenfläche und Auswahl von **Verknüpfung**), Verbinden eines Leseknotens mit dem **Verknüpfung** Knoten und Eingabe des entsprechenden Namen in den **Verknüpfung** Knoten.

4.5.103 TCP/IP Variables

Repräsentiert die Konfiguration der TCP/IP Kommando Variablen Ausgänge.



Der **TCP/IP Variables** Knoten wird verwendet, um Variablen zu definieren. Der **TCP/IP Variables** Knoten wird im Logic Werkzeug angezeigt. Durch die **+** und **-** Tasten können Variablen hinzugefügt oder gelöscht werden.

Werkzeuge können aus der Werkzeugliste gewählt werden, um Werkzeugknoten zu erstellen. Diese stellen die Werte bereit, die mit dem **TCP/IP Variables** Knoten verbunden werden können. Hier ist ein Beispiel:

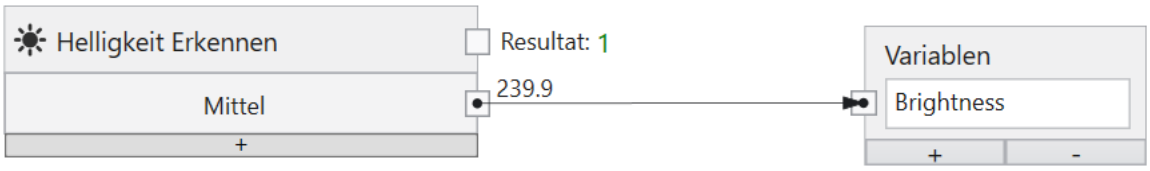


Abb. 189: Ein Werkzeug Knoten für TCP/IP.

Die Ergebnisse mehrerer verschiedener Werkzeuge können mit logischen UND-, ODER- oder NICHT-Knoten kombiniert werden, die durch Klicken mit der rechten Maustaste zur Zeichenfläche hinzugefügt werden.

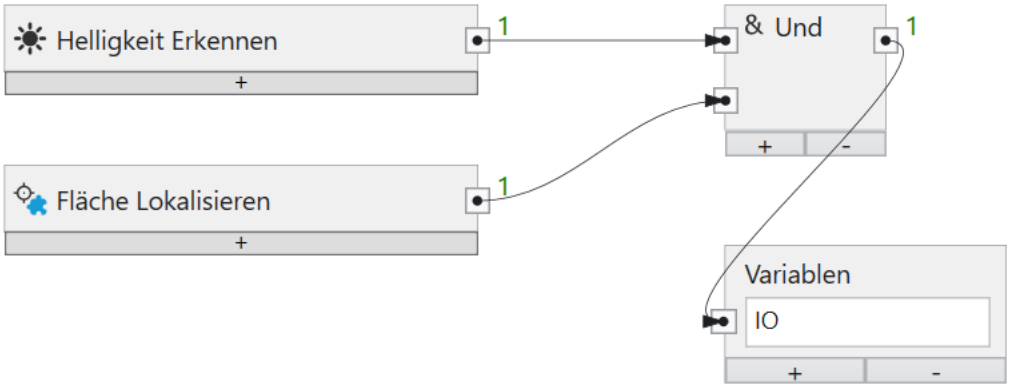


Abb. 190: Logische Kombinationen verschiedener Werkzeuge.

Die logische Verknüpfung kann beliebig komplex sein.

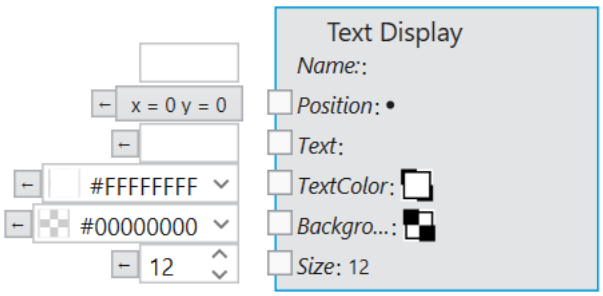
Ein Knoten auf der Leinwand oder eine Verbindung zwischen zwei Knoten kann mit der Taste **Entf** gelöscht werden.

TCP/IP Kommando Variablen können neben logischen Werten auch numerische Werte oder Texte übertragen.

Ist der zu schreibende Wert nicht verfügbar, wird der Text **UNDEFINED** in die Variable geschrieben. Mit dem **Fallback Value**-Knoten kann ein alternatives Verhalten für fehlende Werte definiert werden.

4.5.104 Text Display

Erzeugt eine visuelle Darstellung einese Textes.



Der Text wird auf dem Smart Tool Display und auf der HTML Visualisierung angezeigt.

Eingänge

Name (Type: `String`)

Der Name des Textes, der auf der HTML Visualisierung angezeigt wird, um den Text ein bzw. auszuschalten.

Position (Type: `PointDouble`)

Die Position des Textes.

Text (Type: `String`)

Der Text der angezeigt werden soll.

TextColor (Type: `SolidColorBrushByte`)

Die Farbe des Textes. Standardmäßig ist die Farbe weiß.

Background (Type: `SolidColorBrushByte`)

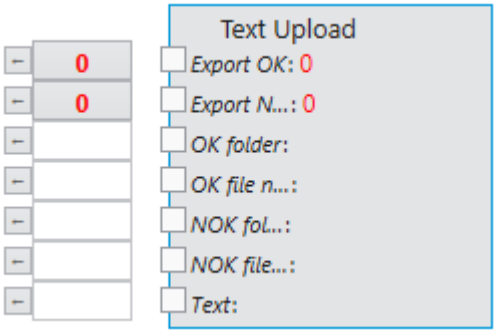
Die Farbe des Hintergrunds. Standardmäßig ist die Farbe transparent.

Size (Type: `Double`)

Die Größe des Textes.

4.5.105 Text Upload

Konfiguriert das Hochladen von Text.



Eingänge

ExportOK (Type: `Boolean`)

Konfiguriert, ob Text zu IO Bildern exportiert werden sollen.

ExportNOK (Type: `Boolean`)

Konfiguriert, ob Texte zu NIO Bildern exportiert werden sollen.

FolderOK (Type: `String`)

Der Ordnername, in den Texte zu IO Bildern exportiert werden sollen.

FileNameOK (Type: `String`)

Der Dateiname für Texte zu IO Bildern.

FolderNOK (Type: `String`)

Der Ordnername, in den Texte zu NIO Bildern exportiert werden sollen.

FileNameNOK (Type: `String`)

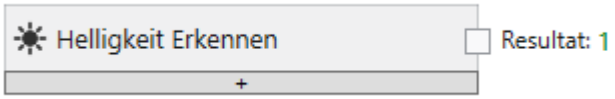
Der Dateiname für Texte zu NIO Bildern.

Text (Type: `String`)

Der zu schreibende Text.

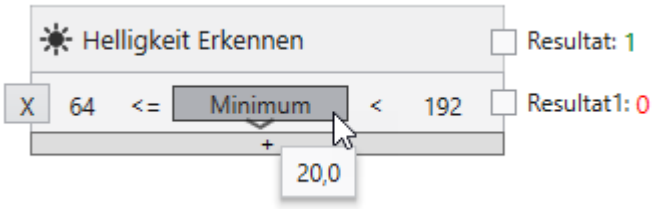
4.5.106 Tool

Liefert Ergebnisse von einzelnen Werkzeugen.



Alle Werkzeuge liefern bool'sche Ergebnisse.

Neben diesen primären Ergebnissen liefern manche Werkzeuge noch sekundäre numerische Ergebnisse, welche über Vergleiche in bool'sche Werte umgewandelt werden können.

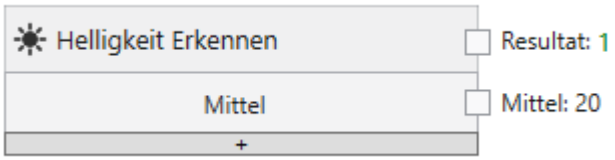


Zusätzliche Zeilen können über die **+** Taste hinzugefügt werden und spezifische Zeilen können über die **X** Taste entfernt werden.

Hier ist beispielhaft gezeigt, wie der Minimalwert des Helligkeits-Werkzeugs ausgewertet wird.

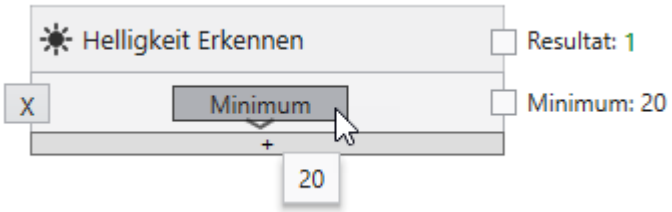
4.5.107 Tool

Liefert Werte von einzelnen Werkzeugen.



Alle Werkzeuge liefern primäre bool'sche Ergebnisse.

Neben diesen primären Ergebnissen liefern manche Werkzeuge noch sekundäre numerische oder textliche Ergebnisse, welche grafisch weiterverarbeitet werden können.

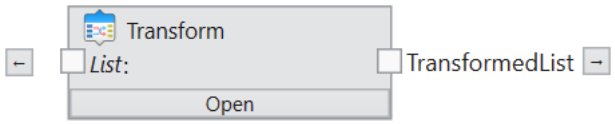


Zusätzliche Zeilen können über die **+** Taste hinzugefügt werden und spezifische Zeilen können über die **X** Taste entfernt werden.

Hier ist beispielhaft gezeigt, wie der Minimalwert des Helligkeits-Werkzeugs ausgewertet wird.

4.5.108 Transform

Transformiert eine Liste in eine zweite Liste.



Die Transformation wird durch ein grafisches Programm erzeugt. Die Taste **Open** öffnet einen Editor, in dem das grafische Programm erstellt werden kann.

Eingänge

List (Type: `List<T>`)

Die Liste der Elemente. Der Typ `T` kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ.

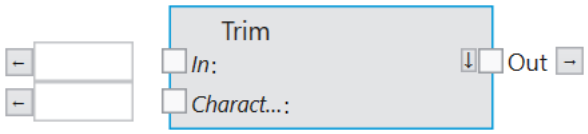
Ausgänge

TransformedList (Type: `List<U>`)

Die Liste der transformierten Elemente. Der Typ `U` kann beliebig sein, aber alle Elemente in der Liste haben denselben Typ. `U` kann gleich sein wie `T`, muss es aber nicht.

4.5.109 Trim

Entfernt alle führenden und nachfolgenden Vorkommen einer in einer anderen Zeichenkette angegebenen Gruppe von Zeichen.



Eingänge

In (Type: `String`)

Der Eingangstext.

Characters (Type: `String`)

Eine Zeichenfolge, die entfernt werden soll, oder eine leere Zeichenfolge. Wenn die Zeichenfolge leer ist, werden Leerzeichen entfernt.

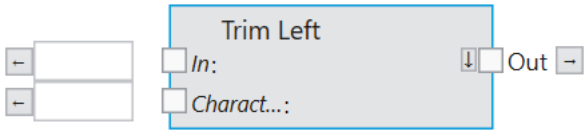
Ausgänge

Out (Typ: `String`)

Die Zeichenfolge, die übrig bleibt, nachdem alle Vorkommen der Zeichen vom Anfang und Ende der Eingabezeichenfolge entfernt wurden.

4.5.110 Trim Left

Entfernt alle führenden Vorkommen einer in einer anderen Zeichenkette angegebenen Gruppe von Zeichen.



Eingänge

In (Type: `String`)

Der Eingangstext.

Characters (Type: `String`)

Eine Zeichenfolge, die entfernt werden soll, oder eine leere Zeichenfolge. Wenn die Zeichenfolge leer ist, werden Leerzeichen entfernt.

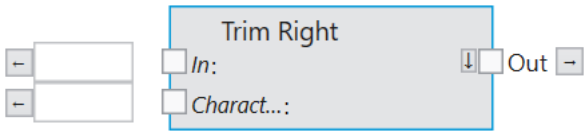
Ausgänge

Out (Typ: `String`)

Die Zeichenfolge, die übrig bleibt, nachdem alle Vorkommen der Zeichen vom Anfang der Eingabezeichenfolge entfernt wurden.

4.5.111 Trim Right

Entfernt alle nachfolgenden Vorkommen einer in einer anderen Zeichenkette angegebenen Gruppe von Zeichen.



Eingänge

In (Type: `String`)

Der Eingangstext.

Characters (Type: `String`)

Eine Zeichenfolge, die entfernt werden soll, oder eine leere Zeichenfolge. Wenn die Zeichenfolge leer ist, werden Leerzeichen entfernt.

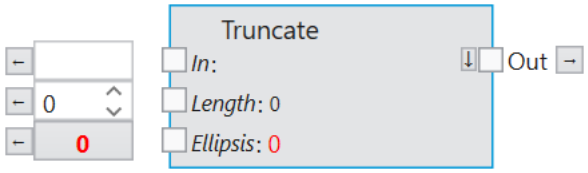
Ausgänge

Out (Typ: `String`)

Die Zeichenfolge, die übrig bleibt, nachdem alle Vorkommen der Zeichen vom Ende der Eingabezeichenfolge entfernt wurden.

4.5.112 Truncate

Kürzt eine Zeichenkette auf eine bestimmte Länge.



Nachfolgende Zeichen der Zeichenkette, die die Länge überschreiten, werden entfernt.

Eingänge

In (Type: `String`)

Der Eingangstext.

Length (Type: `Int32`)

Die Länge des gekürzten Texts.

Ellipsis (Type: `Boolean`)

Bei der Einstellung `true` wird am Ende eine Ellipse `...` eingefügt.

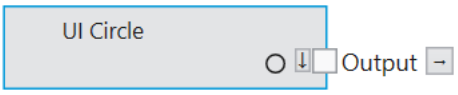
Ausgänge

Out (Typ: `String`)

Die Zeichenfolge, die nach dem Abschneiden der überzähligen Zeichen übrig bleibt.

4.5.113 UI Circle

Definiert einen Kreis durch grafische Eingabe im Benutzerinterface.



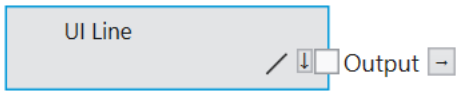
Ausgänge

Output (Type: `Circle`)

Der Kreis.

4.5.114 UI Line

Definiert ein Liniensegment durch grafische Eingabe im Benutzerinterface.



Ausgänge

Output (Type: `LineSegmentDouble`)

Das Liniensegment.

4.5.115 UI Point

Definiert einen Punkt durch grafische Eingabe im Benutzerinterface.



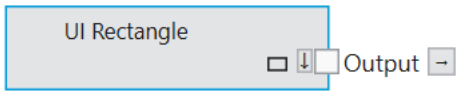
Ausgänge

Output (Type: `PointDouble`)

Der Punkt.

4.5.116 UI Rectangle

Definiert ein Rechteck durch grafische Eingabe im Benutzerinterface.



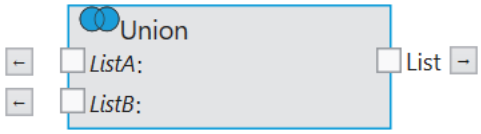
Ausgänge

Output (Type: `Rectangle`)

Das Rechteck.

4.5.117 Union

Bildet die mengen-theoretische Vereinigungsmenge aus zwei Listen.



Eingänge

ListA, ListB (Type: `List<T>`)

Die zwei Listen. Der Typ kann beliebig sein, aber alle Elemente beider Listen haben denselben Typ.

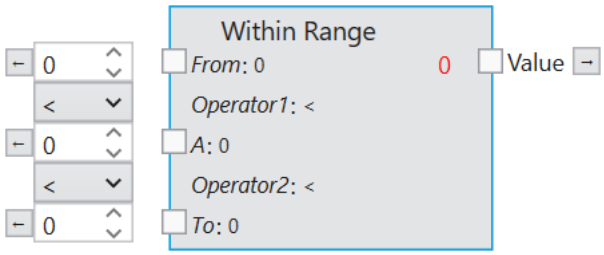
Ausgänge

List (Type: `List<T>`)

Die mengen-theoretische Vereinigungsmenge aus beiden Listen.

4.5.118 Within Range

Überprüft ob sich ein Wert innerhalb eines gegebenen Bereichs befindet.



Die Zahlen werden mit verschiedenen Operatoren verglichen. Wenn die Bedingungen die durch die Operatoren gestellt werden zutreffen wird True/1 bzw. wenn nicht wird False/0 zurückgeliefert.

Eingänge

Operator1, Operator2 (Type: `String`)

Die Vergleichsoperatoren die den Bereich aufspannen. '<' und '<=' stehen zur Verfügung.

From, A, To (Type: `Double`)

Die Zahlen, die verglichen werden sollen. Alle numerischen Datentypen werden nach Bedarf zu Double konvertiert.

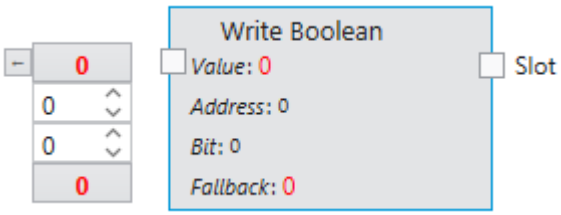
Ausgänge

Value (Type: `Boolean`)

Das Vergleichsergebnis.

4.5.119 Write Boolean

Schreibt einen bool'schen Wert auf eine SPS.



Ein bool'scher Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines bool'schen Eingangs beträgt 1 Bit (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Eingänge

Value (Type: `Boolean`)

Der über den Feldbus geschriebene bool'sche Wert.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Bit (Type: `Byte`)

Gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Fallback (Type: `Boolean`)

Gibt den bool'schen Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

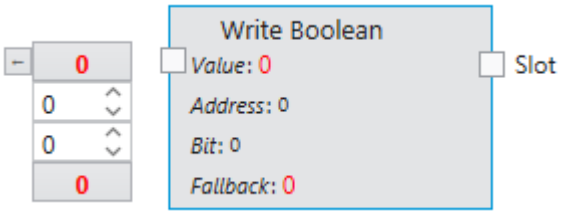
Ausgänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.120 Write Boolean

Schreibt einen bool'schen Wert auf eine SPS.



Ein bool'scher Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines bool'schen Eingangs beträgt 1 Bit (Sie können jedoch bis zu 8 in einem Byte zuweisen).

Eingänge

Value (Type: `Boolean`)

Der über den Feldbus geschriebene bool'sche Wert.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Bit (Type: `Byte`)

Gibt das Bit an, das adressiert wird, wobei 0 das Bit ganz rechts und 7 das Bit ganz links im Byte ist.

Fallback (Type: `Boolean`)

Gibt den bool'schen Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

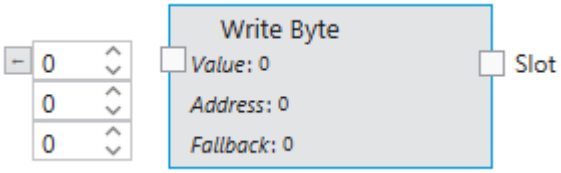
Ausgänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.121 Write Byte

Schreibt einen Byte Wert auf eine SPS.



Ein Byte Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Eingänge

Value (Type: `Byte`)

Der über den Feldbus geschriebene Byte Wert.

Addrese (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Fallback (Type: `Byte`)

Gibt den Byte Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

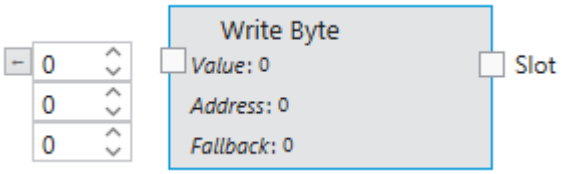
Ausgänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.122 Write Byte

Schreibt einen Byte Wert auf eine SPS.



Ein Byte Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Eingänge

Value (Type: `Byte`)

Der über den Feldbus geschriebene Byte Wert.

Addrese (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Fallback (Type: `Byte`)

Gibt den Byte Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

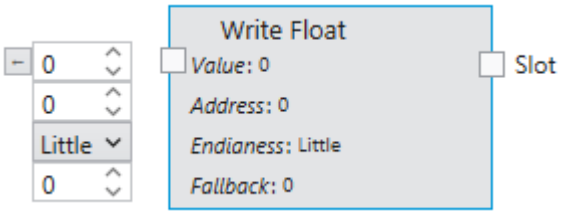
Ausgänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.123 Write Float

Schreibt einen Fließkomma-Wert auf eine SPS.



Ein Float Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines Fließkomma Eingangs beträgt 4 Byte.

Eingänge

Value (Type: `Single`)

Der über den Feldbus geschriebene Fließkomma Wert.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Single`)

Gibt den Float Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

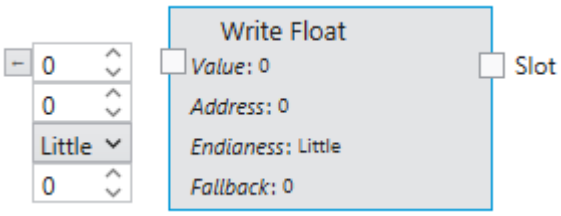
Ausgänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.124 Write Float

Schreibt einen Fließkomma-Wert auf eine SPS.



Ein Float Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines Fließkomma Eingangs beträgt 4 Byte.

Eingänge

Value (Type: `Single`)

Der über den Feldbus geschriebene Fließkomma Wert.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der vier Bytes an, aus denen die 32-Bit-Fließkommazahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Single`)

Gibt den Float Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

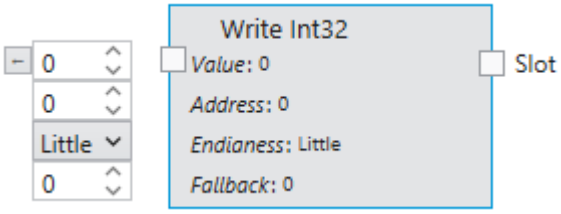
Ausgänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.125 Write Int32

Schreibt einen ganzzahligen Wert auf eine SPS.



Ein Int32 Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines ganzzahligen Eingangs beträgt 4 Byte.

Eingänge

Value (Type: `Int32`)

Der über den Feldbus geschriebene ganzzahlige Int32 Wert.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Byte-Reihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den Zahlenwert Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

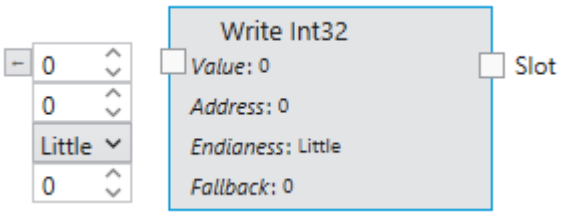
Ausgänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.126 Write Int32

Schreibt einen ganzzahligen Wert auf eine SPS.



Ein Int32 Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines ganzzahligen Eingangs beträgt 4 Byte.

Eingänge

Value (Type: `Int32`)

Der über den Feldbus geschriebene ganzzahlige Int32 Wert.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Endianess (Type: `String`)

Gibt die Byte-Reihenfolge der vier Bytes an, aus denen die 32-Bit-Ganzzahl besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den Zahlenwert Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

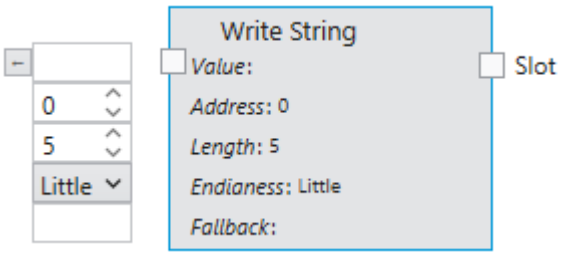
Ausgänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.127 Write String

Schreibt einen Text auf eine SPS.



Ein String Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines Texts beträgt (4 + n), wobei n die maximale Anzahl von Zeichen ist. Die ersten 4 Byte werden als int32 interpretiert und stellt die tatsächliche Anzahl der Zeichen des Strings da.

Eingänge

Value (Type: `String`)

Der über den Feldbus geschriebene Text.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Length (Type: `UInt32`)

Gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den String Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

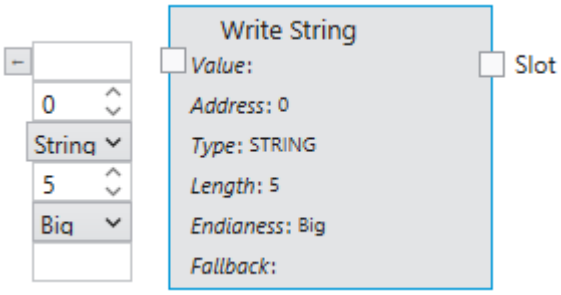
Ausgänge

Slot (Type: `EthernetIpModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.128 Write String

Schreibt einen Text auf eine SPS.



Ein String Eingang wird mit einem Feldbus Eingangs-Steckplatz verbunden wodurch der Wert auf die SPS geschrieben wird.

Die Größe eines Texts beträgt (2 + n) * s, wobei n die maximale Anzahl von Zeichen ist und s = 1 (für einen Ansi STRING) oder s = 2 (für einen Unicode-WSTRING). Das erste Byte/Wort (Ansi/Unicode) enthält die maximale Anzahl von Zeichen, welche die Zeichenfolge enthalten kann. Das zweite Byte/Wort (Ansi/Unicode) enthält die tatsächliche Anzahl der Zeichen, welche die Zeichenfolge enthält.

Eingänge

Value (Type: `String`)

Der über den Feldbus geschriebene Text.

Address (Type: `UInt32`)

Gibt die Byte-Adresse im Modul an.

Type (Type: `String`)

Gibt an, ob die Zeichenfolge Ansi (STRING) oder Unicode (WSTRING) ist.

Length (Type: `UInt32`)

Gibt die Anzahl der Zeichen in der Zeichenfolge an. Eine Länge von 1 gibt ein einzelnes Zeichen an.

Endianess (Type: `String`)

Gibt die Bytereihenfolge der zwei Bytes an, aus denen ein Unicode Zeichen besteht. **Big** bedeutet, dass höherwertige Bytes zuerst gespeichert werden, d. h. an niedrigeren Adressen. **Little** ist das Gegenteil und bedeutet, dass niedrigwertige Bytes zuerst gespeichert werden.

Fallback (Type: `Int32`)

Gibt den String Wert an, der verwendet wird, wenn keine Daten verfügbar sind.

Ausgänge

Slot (Type: `ProfinetModule`)

Verbindet den Knoten mit einem Ausgangs Modul (**CS-6x/ID-6x0** -> SPS).

4.5.129 Xor

Logische Exklusiv-Oder-Verknüpfung von zwei bool'schen Werten.



Die Wertetabelle für eine logische Xor Verknüpfung mit zwei Variablen sieht so aus:

Xor	A	B
0	0	0
1	0	1
1	1	0
0	1	1

Eingänge

X, Y (Type: `Boolean`)

Die zu verknüpfenden bool'schen Eingangswerte.

Ausgänge

Xor (Type: `Boolean`)

Die Exklusiv-Oder-Verknüpfung der Eingänge. Das Ergebnis ist True/1 wenn genau ein Eingang True/1 ist, andernfalls ist das Ergebnis False/0.

4.6 Protokolle

4.6.1 UDP

Die **nVision-i** Software und der **CS-6x/ID-6x0** Vision Sensor kommunizieren mittels UDP (User Datagram Protocol). UDP verwendet ein einfaches verbindungsloses Kommunikationsmodell mit einem Minimum an Protokoll-Mechanismen.

Wenn die **nVision-i** Software läuft, prüft sie das Netzwerk auf vorhandene **CS-6x/ID-6x0** Vision Sensoren. Zu diesem Zweck sendet sie über Broadcast Identifizierungsanforderungen mittels UDP auf Port `1069`. Die Identifizierungsanforderungen werden regelmäßig in Abständen von etwa 4 Sekunden gesendet.

Ein richtig konfigurierter **CS-6x/ID-6x0** Vision Sensor wird auf die Identifizierungsanforderung mit einer UDP Nachricht auf Port `1069` antworten, und sich mit seinem Namen, seiner Seriennummer und seiner IP Adresse identifizieren.

Dieser Mechanismus funktioniert nur, wenn die Firewall (auf dem PC) ausgehende und eingehende UDP Verbindungen auf Port `1069` nicht blockiert. Im Zweifel kontaktieren Sie Ihre IT-Abteilung.

4.6.2 TCP

Nachdem eine Verbindung mittels UDP hergestellt wurde, kommunizieren die **nVision-i** Software und der **CS-6x/ID-6x0** Vision Sensor mittels TCP (Transmission Control Protocol). TCP stellt Kommunikationsdienste zwischen zwei Rechnern zur Verfügung.

Die Kommunikation wird auf Port `50053`' durchgeführt.

Dieser Mechanismus funktioniert nur, wenn die Firewall (auf dem PC) eine TCP Verbindung auf Port `50053` nicht verhindert. Im Zweifel kontaktieren Sie Ihre IT-Abteilung.

4.6.3 SSH

Die **nVision-i** Software und der **CS-6x/ID-6x0** Vision Sensor kommunizieren mittels SSH (Secure Shell). SSH ist ein kryptographisches Netzwerkprotokoll für den sicheren Betrieb von Netzwerkdiensten über ungesicherte Netzwerke.

Bestimmte Funktionen, wie beispielsweise der Firmware-Upgrade erfolgen über SSH. Für SSH wird Port `22` verwendet.

Dieser Mechanismus funktioniert nur, wenn die Firewall (auf dem PC) Verbindungen auf Port `22` nicht blockiert. Im Zweifel kontaktieren Sie Ihre IT-Abteilung.

4.6.4 FTP/SFTP

Der **CS-6x/ID-6x0** Vision-Sensor kann optional Bilder und Texte über FTP (File Transfer Protocol) oder SFTP (Secure File Transfer Protocol) auf einem Server speichern. Die FTP-Option muss aktiviert sein (siehe **Protokoll Einstellungen** des **CS-6x/ID-6x0**) und die Adresse und der Port des FTP-Servers müssen definiert sein.

Das FTP bzw. SFTP Protokoll wird gewählt, indem dem Host ein `ftp://` bzw. `sftp://` vorangestellt wird.

Außerdem müssen die Benutzeranmeldeinformationen (Benutzername und Kennwort) angegeben werden, damit eine Verbindung zum FTP/SFTP-Server hergestellt werden kann. Die Kommunikation zwischen dem eingebauten optionalen FTP/SFTP-Client erfolgt bei FTP standardmäßig über Port `21`, bei SFTP standardmäßig über Port `22`, den Standardports für FTP bzw. SFTP.

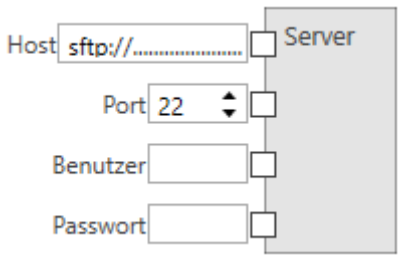


Abb. 191: Die FTP/SFTP Server Einstellungen.

Wenn FTP aktiviert ist, kann ein Job Bilder und Textdateien auf einen FTP/SFTP-Server hochladen. Die Adresse und die Anmeldeinformationen des FTP/SFTP-Servers müssen im Job auf der Registerkarte FTP des Logik-Werkzeugs konfiguriert werden.

Bilder und Texte von Inspektionen können in zwei Verzeichnissen gespeichert werden: **FolderOK** wird für Inspektionen verwendet, die in Ordnung waren, **FolderNOK** wird für fehlgeschlagene Inspektionen verwendet. Das Verzeichnis und/oder der Dateiname der gespeicherten Dateien können auf flexible Weise angegeben werden: Fügen Sie dem Funktionsblockdiagramm einen **Text**-Knoten hinzu und tippen Sie einen Namen oder wählen Sie einen Eintrag aus der Dropdown-Liste, um zwischen **Akquisition-ID**, **Job Name**, **Job Nummer**, **SensorName**, **Sensor Seriennummer**, **Jahr**, **Monat**, **Tag**, **Stunde**, **Minute**, **Sekunde** und **Millisekunde** zu wählen. Eingegebene Namen und vorausgewählte Einträge können frei kombiniert werden, um beliebig komplexe Dateinamen zu erstellen. Wenn Sie einen der vordefinierten Einträge entfernen möchten, wählen Sie das erste Element **Trennen** aus der Dropdown-Liste. Sobald der Name nach Ihren Wünschen erstellt wurde, verbinden Sie die Ausgabe des Knotens **Text** mit den Eingaben **FolderOK**, **FolderNOK**, **FileNameOK** oder **FileNameNOK** des Datei-Knotens.

Wenn der FTP/SFTP-Server nicht zum Hochladen verfügbar ist, werden die Bilder und Texte auf dem **CS-6x/ID-6x0** zwischengespeichert, bis die maximale Anzahl von Dateien erreicht wurde. Wenn die maximale Anzahl erreicht wurde, werden zuerst ältere IO-Bilder und dann ältere NIO-Bilder überschrieben, sodass diese Bilder für den FTP/SFTP-Upload für immer verloren gehen. Dieses Verhalten gibt NIO-Bildern Vorrang.

Alle Dateien, die sich im FTP-Cache befinden und aus irgendeinem Grund nicht hochgeladen werden konnten, können mit nVision-i auf dem PC gespeichert werden.

4.6.5 Telnet

Speicherprogrammierbare Steuerungen beziehungsweise PCs oder auch andere Geräte können über ein textbasiertes TCP/IP Protokoll über den Telnet Port `23` mit einem **CS-6x/ID-6x0** Vision Sensor kommunizieren.

Damit können Befehle zum **CS-6x/ID-6x0** Vision Sensor gesendet werden und entsprechende Resultate zurückgelesen werden.

Weitere Informationen finden sich unter dem Kapitel **TCP/IP Kommando Server**.

4.6.6 TCP/IP Kommando-Server

Der **CS-6x/ID-6x0** Sensor kann über TCP/IP (Telnet) Kommandos von externen Geräten angesprochen und gesteuert werden.

Um den **CS-6x/ID-6x0** über TCP/IP anzusprechen, muss das Kommandoprotokoll aktiviert werden. Dies kann von der nVision-i Startseite erfolgen. (Erweitern Sie die Gerätekachel und klicken Sie auf **Protokoll Einstellungen** und **Kommando-Server aktivieren**). Der **CS-6x/ID-6x0** wird neu gestartet, wenn der Kommando-Server aktiviert oder deaktiviert wird.

Verbindung

Wenn der Kommando-Server aktiviert ist, wartet der **CS-6x/ID-6x0** auf Verbindungen auf Port `23`.

Eine Verbindung kann beispielsweise mit dem Windows 10 Terminal mit diesem Befehl erfolgen:

```
telnet 192.168.3.15
```

Sobald eine Verbindung aufgebaut wurde ist, sendet der **CS-6x/ID-6x0** diesen Text:

```
command server
```

Ab diesem Zeitpunkt können Kommandos über die Schnittstelle gesandt werden. Der **CS-6x/ID-6x0** führt diese Kommandos aus und antwortet entsprechend. Das API ist asynchron, d.h. die entsprechende Antwort des **CS-6x/ID-6x0** muss abgewartet werden, bevor das nächste Kommando abgesetzt werden kann. Das ist typischerweise kein Problem bei einer Texteingabe über Telnet, muss aber bei der Verwendung einer SPS oder eines Steuerungs-PCs beachtet werden.

Kommandos

ActiveJob

setzt den aktiven Job

→ `activejob slot<CRLF>`

Aktiviert den Job in einem bestimmten numerischen Slot. Wenn der aktive Job gesetzt werden konnte, antwortet der **CS-6x/ID-6x0** mit `Ok!`. Andernfalls antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(Ok!|Error!)<CRLF>`

ActiveJob?

frägt den aktiven Job ab

→ `activejob?<CRLF>`

Der numerische Steckplatz des aktiven Jobs wird zurückgegeben. Wenn kein Job aktiv ist, antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(slot|Error!)<CRLF>`

Blink

blitzt die LEDs drei mal

→ `blink<CRLF>`

Dieser Befehl dient dazu, ein **CS-6x/ID-6x0**-Gerät visuell zu lokalisieren. Dies kann einen Inspektionszyklus stören und sollte daher nicht während einer laufenden Inspektion ausgegeben werden. Der **CS-6x/ID-6x0** antwortet mit `Ok!`.

← `Ok!<CRLF>`

BootJob

setzt den Bootjob

→ `bootjob slot<CRLF>`

Macht den Job in einem bestimmten numerischen Slot zum Bootjob. Wenn der Bootjob festgelegt werden konnte, antwortet der **CS-6x/ID-6x0** mit `Ok!`, andernfalls antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(Ok!|Error!)<CRLF>`

BootJob?

frägt den Bootjob ab

→ `bootjob?<CRLF>`

Der numerische Steckplatz des Bootjobs wird zurückgegeben. Wenn kein Job als Bootjob festgelegt ist, antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(slot|Error!)<CRLF>`

Clock

stellt die Sensor Uhr ein

→ `clock datetime<CRLF>`

Stellt Datum und Uhrzeit der Sensoruhr ein. Datum und Uhrzeit sollten in folgender Form angegeben werden: `15.01.2021 08:30:00 AM`. Falls die Uhr gestellt werden konnte, antwortet der **CS-6x/ID-6x0** mit `Ok!`, andernfalls antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(Ok!|Error!)<CRLF>`

Clock?

liest die Sensor Uhr aus

→ `clock?<CRLF>`

Ruft Datum und Uhrzeit von der Sensoruhr ab.

← `1/1/2021 8:00:00 AM<CRLF>`

CloseInspectionGate

das Inspektions-Tor für eine ständige Inspektion schließen

→ `closeinspectiongate<CRLF>`

Der Erfassungsmodus im Job muss auf Gated eingestellt sein, andernfalls wird dieser Befehl ignoriert. Wenn das Tor geschlossen werden konnte, antwortet der **CS-6x/ID-6x0** mit `Ok!`, andernfalls antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(Ok!|Error!)<CRLF>`

Cls

löscht den Schirm

→ `cls<CRLF>`

Löscht den Bildschirm durch Senden von VT100-Terminal-Escape-Sequenzen: `Esc [1J` löscht den Bildschirm vom Cursor nach oben. `Esc [H` bewegt den Cursor in die obere linke Ecke.

← `Esc[1JEsc[H`

Exit

verlässt die Sitzung

→ `exit<CRLF>`

Meldet sich vom Kommando-Server ab und schließt die Verbindung.

← `Ok!<CRLF>`

Get

liest den Wert eines Symbols

→ `get[?] name<CRLF>`

Ruft den Wert des Symbols `name` ab. Das Fragezeichen ist optional. Der Symbolname muss mit einem Groß- oder Kleinbuchstaben [A-Za-z] beginnen und kann von Groß- oder Kleinbuchstaben, Zahlen oder einem Unterstrich gefolgt werden. Der zugehörige Wert (möglicherweise mehrere Zeilen) wird zurückgegeben, wenn das Symbol vorhanden ist, andernfalls wird `Error!` zurückgegeben.

← `(...value...|Error!)<CRLF>`

Help

zeigt die Kommando Hilfe

→ `help[?][ command]<CRLF>`

Das Fragezeichen und command sind optional. Wenn kein Kommando (oder ein falsches Kommando) angegeben wird, zeigt die Hilfe eine Liste der Befehle an. Wenn ein Kommando gegeben wird, zeigt die Hilfe die spezifische Hilfe des Kommandos an.

← `...multiline text...` ← `Ok!<CRLF>`

Jobs?

zählt die Jobs auf

→ `jobs[?]<CRLF>`

Das Fragezeichen ist optional. Die Jobs werden mit ihrem Slot gefolgt von ihrem Namen aufgelistet, getrennt durch ein Leerzeichen. Wenn keine Jobs geladen sind, antwortet der **CS-6x/ID-6x0** mit `Error!`

← `...multiline job list...` ← `Ok!<CRLF>`

oder

← `Error!<CRLF>`

License?

liest die Lizenz zurück

→ `license[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die Lizenz zurück. Mögliche Lizenzzeichenfolgen sind `Basic`, `Basic & Measure`, `Basic & Barcode` oder `Basic, Measure & Barcode`.

← `...license description...<CRLF>`

MacAddress?

liest die MAC-Adresse zurück

→ `macadress[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die MAC-Adresse zurück. Eine MAC-Adresse sieht so aus `8C:0F:A0:00:10:69`.

← `...MAC address...<CRLF>`

Model?

liest das Modell zurück

→ `model[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die Modellbeschreibung zurück.

← `...model description...<CRLF>`

Name?

liest den Namen des **CS-6x/ID-6x0** zurück

→ `name[?]<CRLF>`

Das Fragezeichen ist optional. Gibt den Namen zurück.

← `...name...<CRLF>`

Offline

setzt den **CS-6x/ID-6x0** in den Offline Status

→ `offline[CRLF]`

Versucht, den **CS-6x/ID-6x0** offline zu schalten. Wenn der **CS-6x/ID-6x0** offline gehen könnte, lautet die Antwort `Ok!`, andernfalls `Error!`. Es kann einige Zeit dauern, bis der **CS-6x/ID-6x0** offline geschaltet ist, da jeder aktuelle Ausführungszyklus beendet werden muss.

← `(Ok!|Error!)<CRLF>`

Online

setzt den **CS-6x/ID-6x0** in den Online Status

→ `online<CRLF>`

Versucht, den **CS-6x/ID-6x0** online zu stellen. Wenn der **CS-6x/ID-6x0** online gehen könnte, lautet die Antwort `Ok!`, andernfalls `Error!`. Es kann einige Zeit dauern, bis der **CS-6x/ID-6x0** online ist. Falls auf dem **CS-6x/ID-6x0** kein Job verfügbar ist, kann der **CS-6x/ID-6x0** nicht online gehen.

← `(Ok!|Error!)<CRLF>`

Online?

liest den Online-Status zurück

→ `online?<CRLF>`

Fragt den Online-Status ab. Die Antwort lautet `Ok!`, wenn der **CS-6x/ID-6x0** online ist, und `Error!`, wenn der **CS-6x/ID-6x0** offline ist.

← `(Ok!|Error!)<CRLF>`

OpenInspectionGate

das Inspektions-Tor für eine ständige Inspektion öffnen

→ `openinspectiongate<CRLF>`

Der Erfassungsmodus im Job muss auf Gated eingestellt sein, andernfalls wird dieser Befehl ignoriert. Wenn das Tor geöffnet werden konnte, antwortet der **CS-6x/ID-6x0** mit `Ok!`, andernfalls antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `(Ok!|Error!)<CRLF>`

Prompt

setzt den interaktiven Prompt

→ `prompt[ ...characters...]<CRLF>`

Legt die interaktive Eingabeaufforderung fest. Eine interaktive Eingabeaufforderung (z. B. >) ist in interaktiven Terminalsitzungen sehr nützlich. In SPS-Sitzungen kann die Eingabeaufforderung ein Problem darstellen, daher wird standardmäßig eine leere Zeichenfolge festgelegt.

← `Ok!<CRLF>` ← `...characters...<CRLF>`

Push

überträgt den Wert einer Variablen nach jeder Inspektion

→ `push[ variable]<CRLF>`

Nach jeder Inspektion wird der Wert der angegebenen Variable übertragen, push ohne Variable entfernt die Anfrage.

← `Ok!<CRLF>`

Ready?

liest den Ready-Status zurück

→ `ready?<CRLF>`

Das Fragezeichen ist optional. Der Bereitschaftsstatus des **CS-6x/ID-6x0** bleibt erhalten. Der **CS-6x/ID-6x0** ist bereit, wenn er einen Inspektionszyklus beendet hat. Der **CS-6x/ID-6x0** ist nicht bereit, wenn er sich mitten in einem Inspektionszyklus befindet. Wenn der **CS-6x/ID-6x0** offline ist, ist er ebenfalls nicht bereit. Der **CS-6x/ID-6x0** antwortet mit `Ok!` falls er bereit ist, andernfalls mit `Error!`.

← `(Ok!|Error!)<CRLF>`

Resolution?

liest die Kamera-Auflösung zurück

→ `resolution[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die Auflösung der **CS-6x/ID-6x0** Kamera zurück.

← `...resolution...<CRLF>`

Results?

listet die Inspektions-Ergebnisse auf

→ `results[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die Ergebnisse der letzten Inspektion zurück. Wenn keine Ergebnisse verfügbar sind, antwortet der **CS-6x/ID-6x0** mit `Error!`.

← `...multiline results...` ← `Ok!<CRLF>`

oder

← `Error!<CRLF>`

Serial?

liest die Seriennummer zurück

→ `serial[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die **CS-6x/ID-6x0**-Seriennummer zurück. Eine Seriennummer sieht beispielsweise aus wie `10030`.

← `...serial number...<CRLF>`

Set

setzt den Wert eines Symbols

→ `set name=value<CRLF>`

Setzt das Symbol `name` auf den Wert `value`. Der Symbolname muss mit einem Groß- oder Kleinbuchstaben [A-Za-z] beginnen und kann von Groß- oder Kleinbuchstaben, Zahlen oder einem Unterstrich gefolgt werden. Der **CS-6x/ID-6x0** antwortet mit `Ok!`, wenn er den Wert einstellen könnte, andernfalls mit `Error!`.

← `(Ok!|Error!)<CRLF>`

Trigger

löst die Bildaufnahme aus

→ `trigger<CRLF>`

Startet die Bildaufnahme und -prüfung. Die Antwort lautet `Ok!`, wenn der **CS-6x/ID-6x0** den Trigger akzeptiert hat, und `Error!` wenn der **CS-6x/ID-6x0** nicht bereit war und den Trigger nicht akzeptierte. Der Trigger Modus im Akqusitions-Werkezeug des Jobs muss auf `Virtuell` eingestellt sein, ansonsten wird das Trigger-Kommando nicht angenommen und mit `Error!` quittiert.

← `(Ok!|Error!)<CRLF>`

Variables?

zählt die Variablen des Jobs auf

→ `variables[?]<CRLF>`

Das Fragezeichen ist optional. Alle Jobvariablen werden zusammen mit ihren tatsächlichen Werten aufgelistet.

← `...multiline variables...` ← `Ok!<CRLF>`

oder

← `Error!<CRLF>`

Version?

liest die Firmware Version zurück

→ `version[?]<CRLF>`

Das Fragezeichen ist optional. Gibt die **CS-6x/ID-6x0** Firmware-Version zurück. Eine Firmware-Version sieht aus wie `20.4.1205`, wobei 20 das Jahr ist, 4 eine Ordnungszahl innerhalb des Jahres ist und 1205 eine numerische Build-Nummer ist.

← `...firmware version...<CRLF>`

Beispiele

Jobwechsel

Ein Jobwechsel wird ausgeführt, indem der **CS-6x/ID-6x0** offline geschaltet, der aktive Job auf einen neuen Steckplatz gesetzt und der **CS-6x/ID-6x0** wieder online geschaltet wird.

→ `offline<CRLF>`
← `Ok!<CRLF>`
→ `activejob 2<CRLF>`
← `Ok!<CRLF>`
→ `online<CRLF>`
← `Ok!<CRLF>`

Bildaufnahme

Um ein Bild aufzunehmen, sollten Sie das Flag `ready?` lesen, bevor Sie einen Trigger auslösen.

→ `ready?<CRLF>`
← `Ok!<CRLF>`
→ `trigger<CRLF>`
← `Ok!<CRLF>`

4.6.7 TCP/IP-Streams

Bis zu vier Ports können für TCP/IP Streams verwendet werden, je nach der vorgewählten Anzahl in den **Protokoll Einstellungen**. Die Port Nummern sind vordefiniert von 49211 bis 49214.

Auf der Seite **TCP/IP Variablen Ausgänge** kann definiert werden, was auf den jeweiligen TCP/IP Streams ausgegeben wird. Der Ausgabewert wird nach jeder Bildauswertung einmal auf dem TCP/IP Port ausgegeben. Hier ist ein Beispiel:

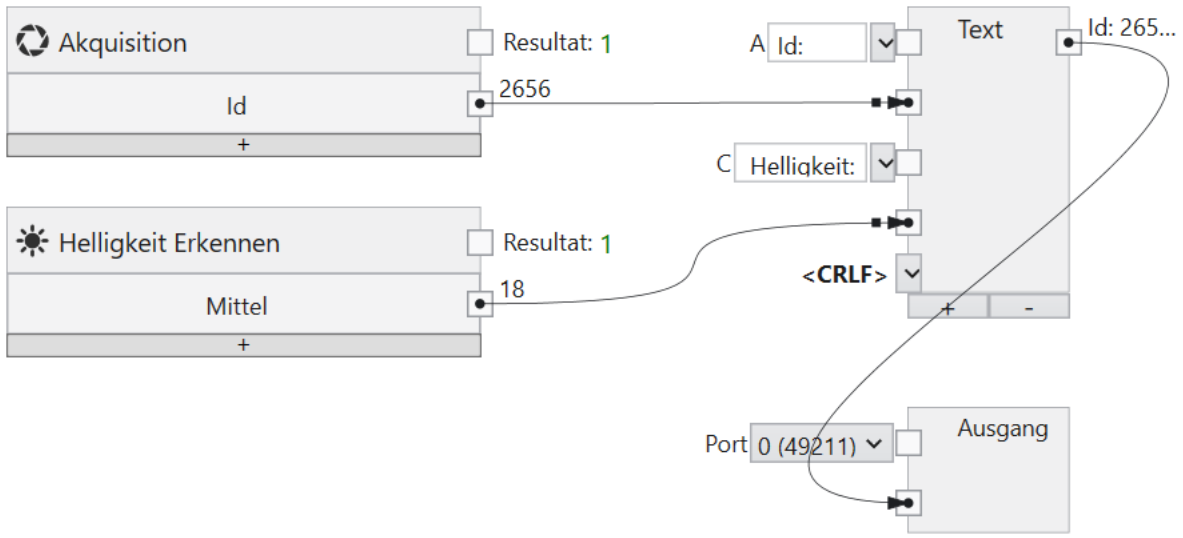


Abb. 192: Text-Kombination für einen TCP/IP Ausgangs-Stream.

4.6.8 PROFINET

Der **CS-6x/ID-6x0** Sensor kann als ein PROFINET Gerät in einer PROFINET Umgebung verwendet werden.

Um den **CS-6x/ID-6x0** als PROFINET Gerät zu verwenden, muss das PROFINET Kommunikationsprotokoll aktiviert werden. Dies kann von der nVision-i Startseite erfolgen. (Erweitern Sie die Gerätekachel und klicken Sie auf **Protokoll Einstellungen** und **Profinet aktivieren**). Der **CS-6x/ID-6x0** wird neu gestartet, wenn PROFINET aktiviert oder deaktiviert wird. Wenn PROFINET aktiviert ist, startet der **CS-6x/ID-6x0** im Offline-Modus.

Die PROFINET Gsd Datei ist im Unterverzeichnis `Profinet` im nVision-i Installationsverzeichnis (`%PROGRAMME%\di-soric\nVision-i\`) abgelegt.

PROFINET Identität

Ein PROFINET Gerät wird durch eine **Vendor Id** und eine **Device Id** eindeutig identifiziert. Die di-soric Vendor Id ist 0x0221.

Die Tabelle zeigt die Device Id der verschiedenen Modelle:

Modell	Profinet Device Id
CS60	0x0001
CS62	0x0002

Verbindung

Ein PROFINET Master und ein PROFINET Gerät tauschen Daten in einem zyklischen Nachrichtenprotokoll aus. Die Zykluszeit des Vision Sensors beträgt 8 ms.

Der ausgetauschte Datenblock beträgt maximal 1024 Byte sowohl für Input als auch Output. Die aktuelle Größe des ausgetauschten Blocks wird von der Einstellung der SPS gesteuert. Die SPS

kann Module in verschiedenen Größen (1 Byte, 2 Byte, etc.) in bis zu 256 Slots einlesen oder ausgeben. Die aktuelle Slot-Konfiguration bestimmt, wieviele Byte zwischen der SPS und dem **CS-6x/ID-6x0** ausgetauscht werden.

Vorbereitung der SPS

Integrieren Sie den **CS-6x/ID-6x0** in die SPS Umgebung mit der TIA Portal Software. Suchen Sie die **CS-6x/ID-6x0** GSD Datei (`GSDML-V2.42-di-soric-CS-6x/ID-6x0-...`) im `Profinet` Unterverzeichnis des nVision-i Installations-Verzeichnis (`%PROGRAMME%\di-soric\nVision-i\`) und verwenden Sie das Kommando `Extras -> Gerätebeschreibungsdateien (GSD) verwalten` um die GSD Datei zur SPS hinzuzufügen.

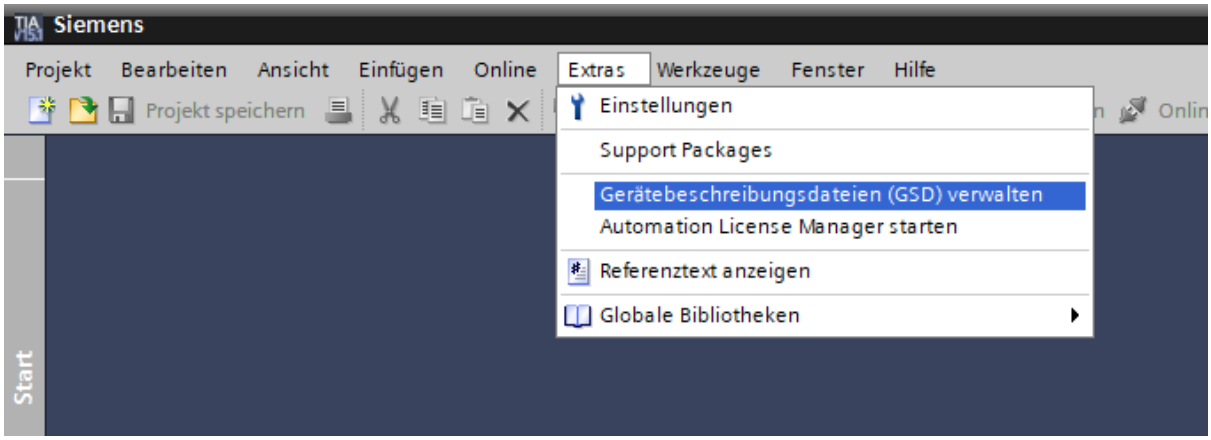


Abb. 193: Finden der GSD Datei des **CS-6x/ID-6x0**

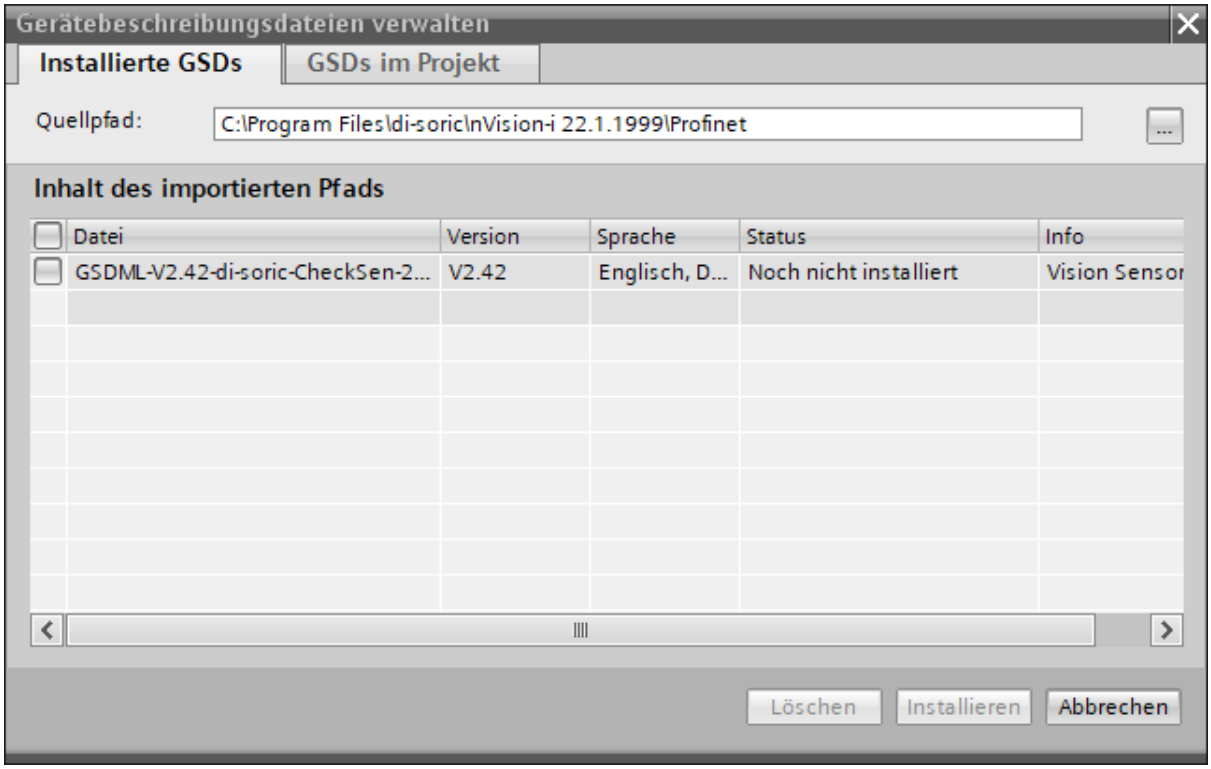


Abb. 194: Hinzufügen der GSD Datei des **CS-6x/ID-6x0** zu TIA

Nachdem Sie die SPS zum TIA Projekt hinzugefügt haben, fügen Sie auch den **CS-6x/ID-6x0** zum Projekt hinzu. Der **CS-6x/ID-6x0** ist im Hardware Katalog, unter `Weitere Feldgeräte -> PROFINET IO -> General -> di-soric -> Vision Sensor -> CS-6x/ID-6x0` aufgelistet.

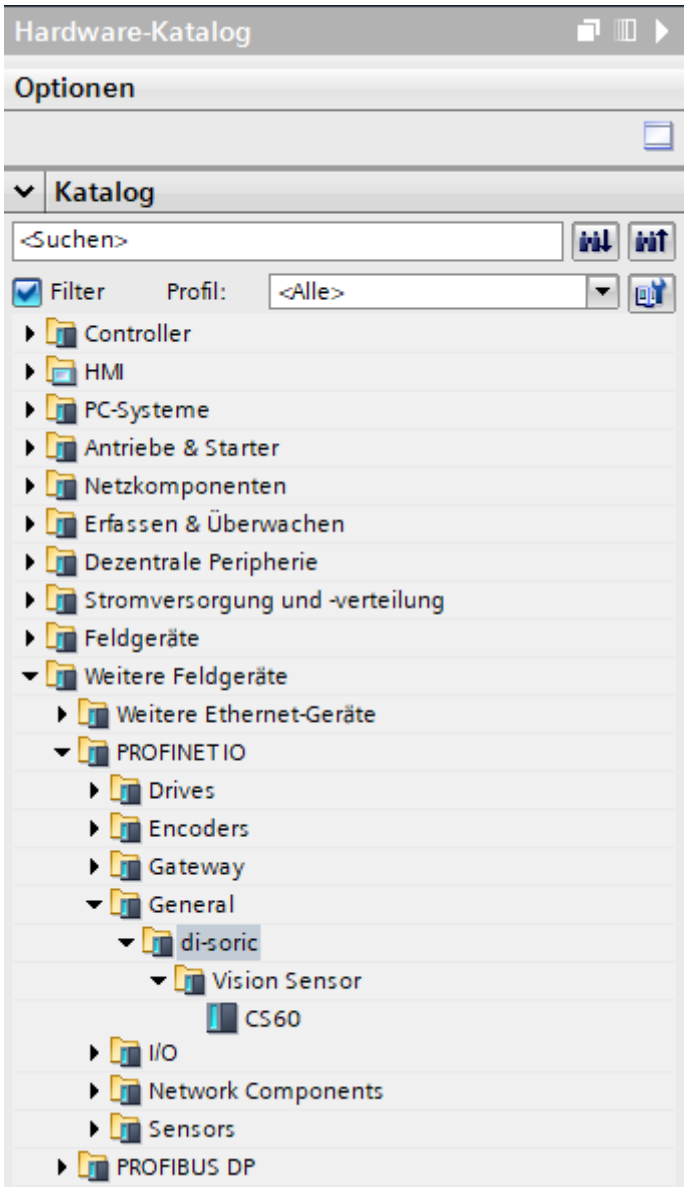


Abb. 195: Der TIA Hardware Katalog

Nachdem der **CS-6x/ID-6x0** hinzugefügt wurde, muss er mit der SPS verbunden werden.

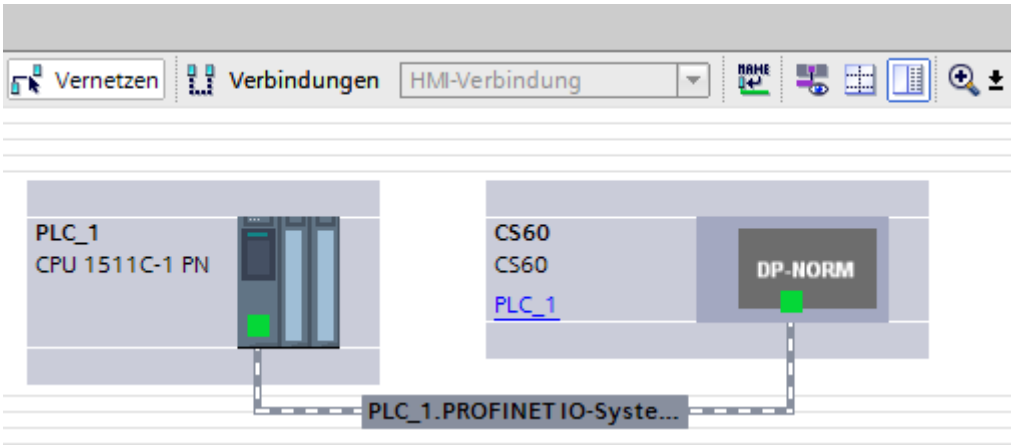


Abb. 196: Verbindung der SPS mit dem **CS-6x/ID-6x0**

Danach müssen Sie die Input und Output Module in die Slots des **CS-6x/ID-6x0** konfigurieren. Hier ist ein Beispiel:

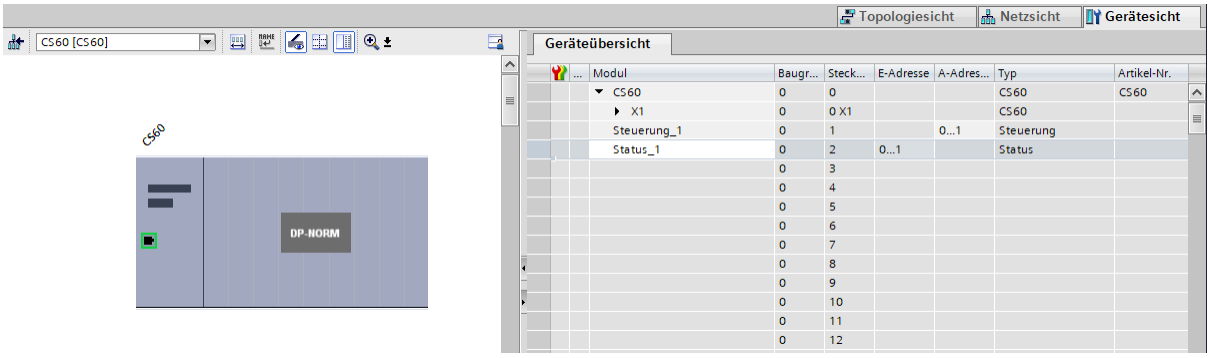


Abb. 197: Module zu den **CS-6x/ID-6x0** Slots hinzufügen

Es müssen jeweils mindestens 2 Byte Eingänge und Ausgänge für die Steuerungs- und Status-Signale konfiguriert werden.

Die Einstellungen müssen final kompiliert werden und auf die SPS geladen werden.

4.6.9 EtherNet/IP

Der **CS-6x/ID-6x0** Sensor kann als ein EtherNet/IP Gerät in einer EtherNet/IP Umgebung verwendet werden.

Um den **CS-6x/ID-6x0** als EtherNet/IP Gerät zu verwenden, muss das EtherNet/IP Kommunikationsprotokoll aktiviert werden. Dies kann von der nVision-i Startseite erfolgen. (Erweitern Sie die Gerätekachel und klicken Sie auf **Protokoll Einstellungen** und **EtherNet/IP aktivieren**). Der **CS-6x/ID-6x0** wird neu gestartet, wenn EtherNet/IP aktiviert oder deaktiviert wird. Wenn EtherNet/IP aktiviert ist, startet der **CS-6x/ID-6x0** im Offline-Modus.

Die EtherNet/IP EDS Datei ist im Unterverzeichnis `EthernetIP` im nVision-i Installationsverzeichnis (`%PROGRAMME%\di-soric\nVision-i\`) abgelegt.

EtherNet/IP Identität

Ein EtherNet/IP Gerät wird durch eine **Vendor Id** und einem **Product Code** eindeutig identifiziert. Die di-soric Vendor Id ist 1663.

Die Tabelle zeigt die Product Codes der verschiedenen Modelle:

Modell	EtherNet/IP Product Code
CS60	1
ID600	2
CS62	3
ID620	4

Verbindung

Ein EtherNet/IP Scanner und ein EtherNet/IP Adapter tauschen Daten in einem zyklischen Nachrichtenprotokoll aus.

Der ausgetauschte Datenblock beträgt maximal 400 Byte sowohl für Input als auch Output.

Der 8 Byte große Config Block wird derzeit nicht benutzt und sollte nicht angetastet werden.

Vorbereitung der PLC (Allen-Bradley CompactLogix 5380)

Integrieren Sie den **CS-6x/ID-6x0** in die PLC Umgebung mit dem Logix Designer Studio 5000. Suchen Sie die **CS-6x/ID-6x0** EDS Datei im `EthernetIP` Unterverzeichnis des nVision-i Installations-Verzeichnis (`%PROGRAMME%\di-soric\nVision-i\`) und verwenden Sie das Kommando `Tools -> EDS Hardware Installation Tool` um die EDS Datei in der PLC zu registrieren.

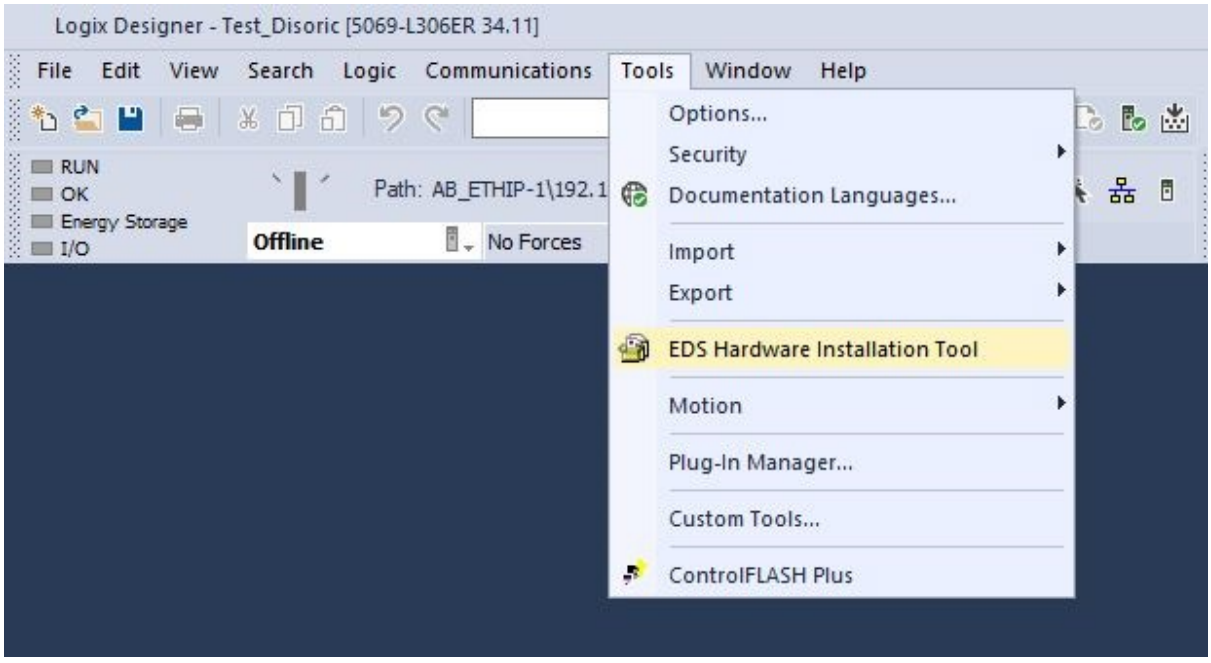


Abb. 198: Finden der EDS Datei des **CS-6x/ID-6x0**

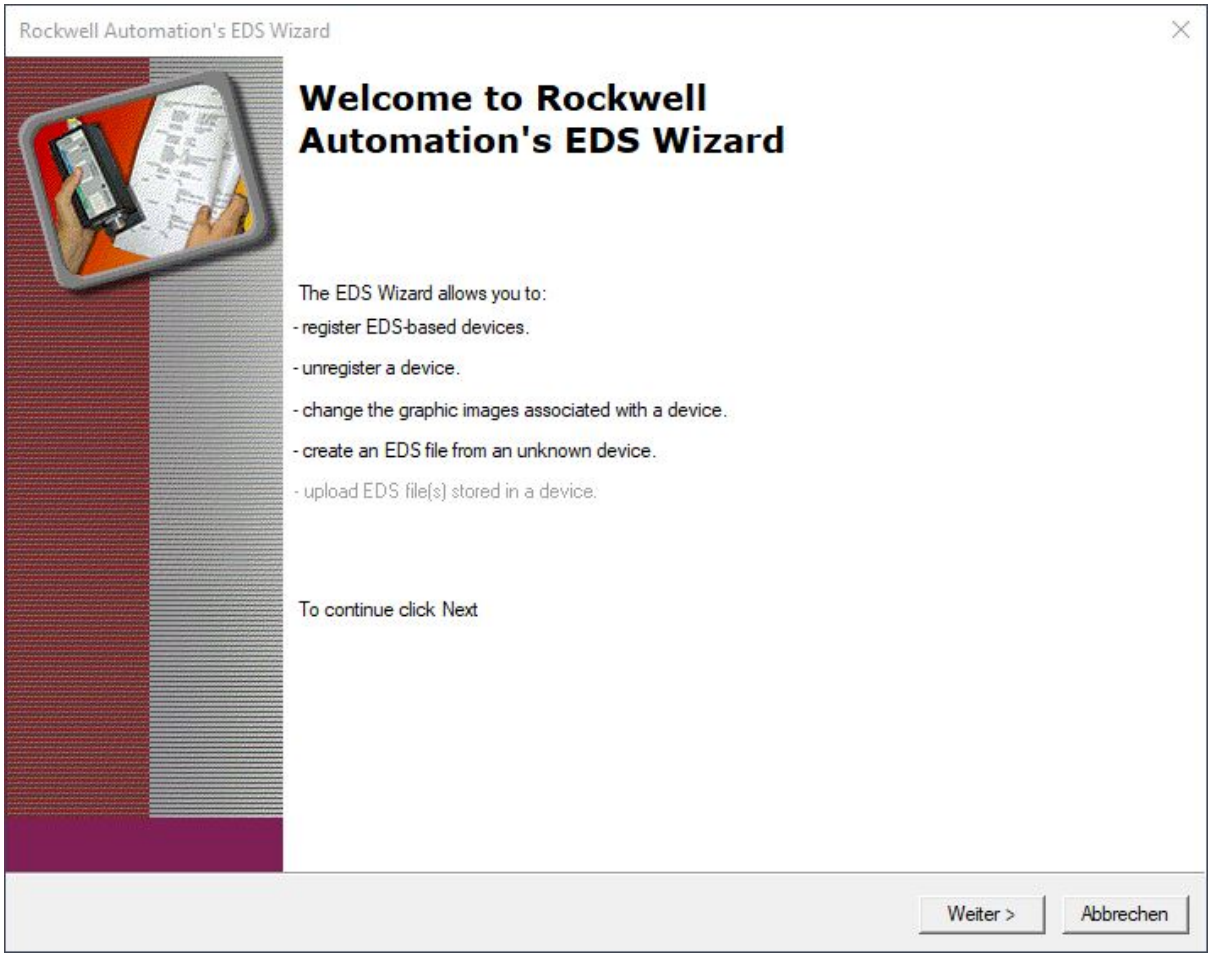


Abb. 199: Hinzufügen der EDS Datei des **CS-6x/ID-6x0** zu Studio 5000

Nachdem Sie ein neues Projekt erstellt haben, fügen Sie auch den **CS-6x/ID-6x0** zum Projekt hinzu. Der **CS-6x/ID-6x0** ist im Katalog, unter CS-6x/ID-6x0 zu finden.

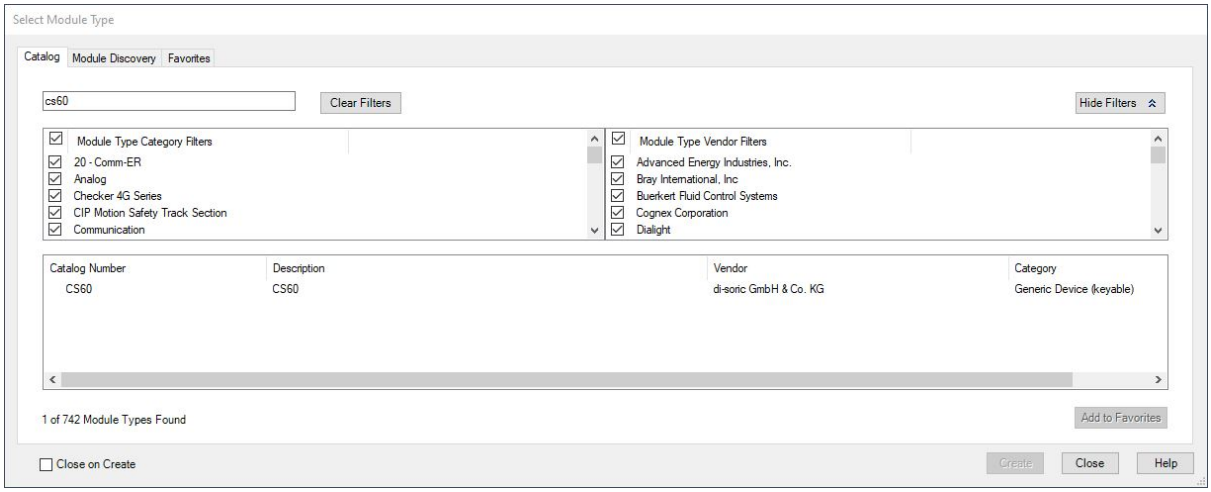


Abb. 200: Hinzufügen des CS-6x/ID-6x0 zum Projekt

Nachdem der **CS-6x/ID-6x0** hinzugefügt wurde, muss er mit einem Namen und IP-Addresse konfiguriert werden.

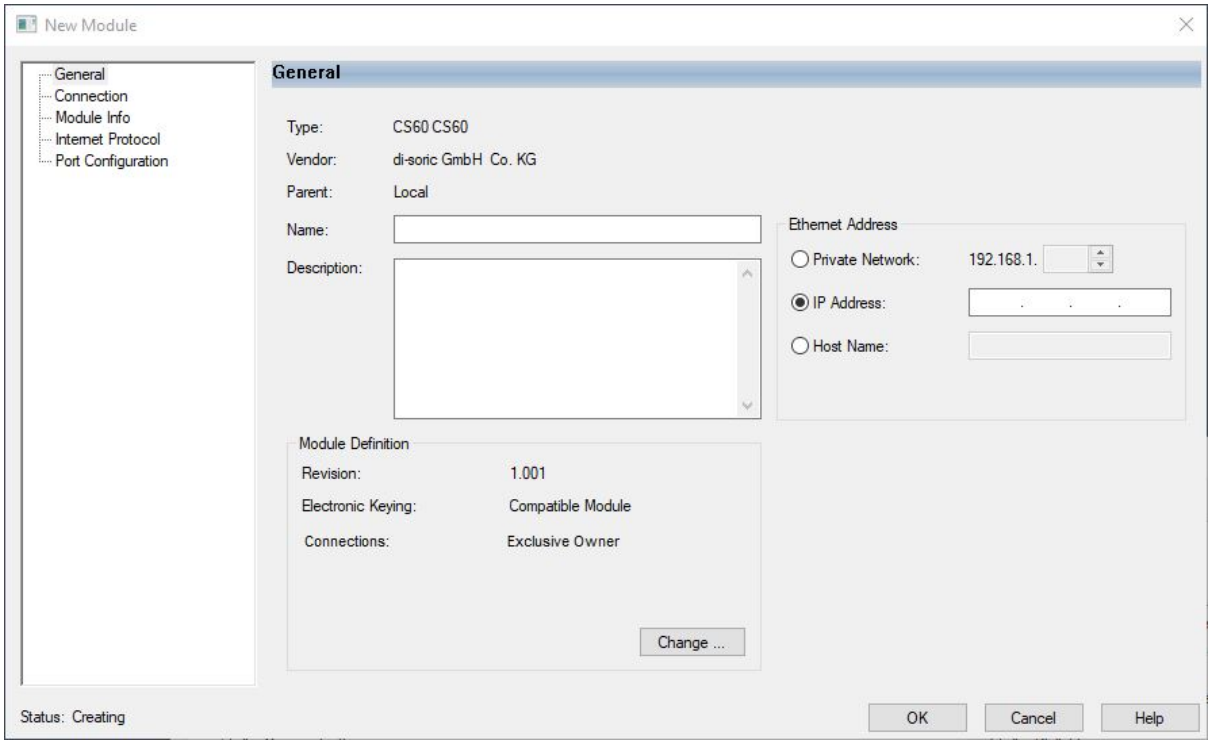


Abb. 201: Konfiguration des **CS-6x/ID-6x0**

4.6.10 Standard Status- und Kontroll Signale für PROFINET oder EtherNet/IP

CS-6x/ID-6x0 Statusinformationen werden vom **CS-6x/ID-6x0** zur PLC gesendet. Standardmäßig sieht die Zuordnung so aus:

Byte Offset	Bits	Data	Beschreibung
0	0	Online	Wahr falls der **CS-6x/ID-6x0** Online ist, falsch falls der **CS-6x/ID-6x0** Offline ist.
0	1	Ack-SetOnline	Umschaltung des `SetOnline` Kontrollbits quittieren.
0	2		
0	3	Ack-JobLoad	Umschaltung des `JobLoad` Kontrollbits quittieren.
0	4	JobLoadOk	Wahr falls ein Job geladen wurde, falsch falls das Laden fehlschlug.
0	5	Ready	Wahr falls der **CS-6x/ID-6x0** einen Triggerimpuls akzeptieren kann, falsch falls er beschäftigt ist.
0	6	AckTrigger	Trigger quittieren. Wahr nach der steigenden Flanke, falsch nach der fallenden Flanke von Trigger.
0	7	Acquiring	Bit ist während der Bildaufnahme wahr, danach falsch.
1	0-7	JobId	Die Id des gegenwärtigen Jobs (0 .. 255).

Acquiring wird zusammen mit AckTrigger(wahr) gesetzt. Abhängig vom zeitlichen Ablauf kann beim Setzen von AckTrigger(wahr) die Aufnahme schon beendet sein und Acquiring bleibt falsch.

CS-6x/ID-6x0 Kontrollinformationen werden von der PLC zum **CS-6x/ID-6x0** gesendet. Standardmäßig sieht die Zuordnung so aus:

Byte Offset	Bits	Data	Beschreibung
0	0	SetOnline	Setzen, um den **CS-6x/ID-6x0** Online zu schalten, löschen, um ihn Offline zu schalten.
0	1	JobLoad	Das Job Load Kommando ausführen.
0	2	BootJob	Wahr, um den gewählten Job zum Boot Job zu machen, andernfalls falsch.
0	3	Trigger	Eine Bildaufnahme triggern.
1	0-7	JobLoadId	Die Id des Jobs, der geladen werden soll (0 .. 255).

Der Vision Sensor startet im Offline Modus, wenn PROFINET oder EtherNet/IP aktiviert ist.

Online Status

Der Online oder Offline Status des Vision Sensor wird im **Online** bit (Status Byte 0, Bit 0) angezeigt. Wenn dieses Bit gesetzt ist, ist der Vision Sensor Online, andernfalls ist er Offline.

Im Online Modus wird ein Job ausgeführt.

Im Offline Modus wird kein Job ausgeführt, aber der Vision Sensor hört auf Kommandos, z.B. zum Job-Wechsel.

Zum Wechseln von Offline nach Online, oder andersherum, setzen oder löschen Sie das **SetOnline** Bit (Kontroll Byte 0, Bit 0).

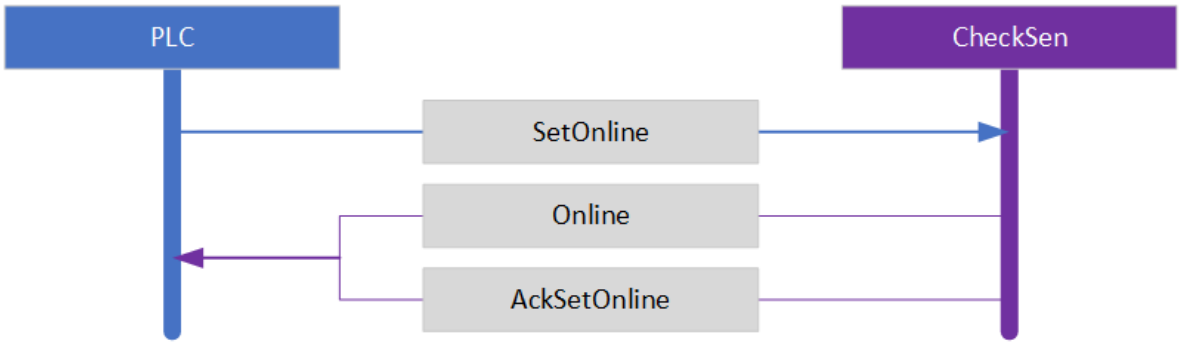


Abb. 202: Die SetOnline Sequenz

Online Setzen

Wenn der Vision Sensor Offline ist, kann er in den Online Modus versetzt werden. Um den Vision Sensor Online zu setzen, setzen Sie das **SetOnline** Bit (Kontroll Byte 0 Bit 0).

Der Vision Sensor kann nur Online gehen, wenn vorher ein Job geladen wurde.

Um zu überprüfen, ob der Vision Sensor Online gehen konnte, warten Sie bis das **AckSetOnline** Bit (Status Byte 0, Bit 1) gesetzt ist und prüfen Sie das **Online** Bit (Status Byte 0, Bit 0).

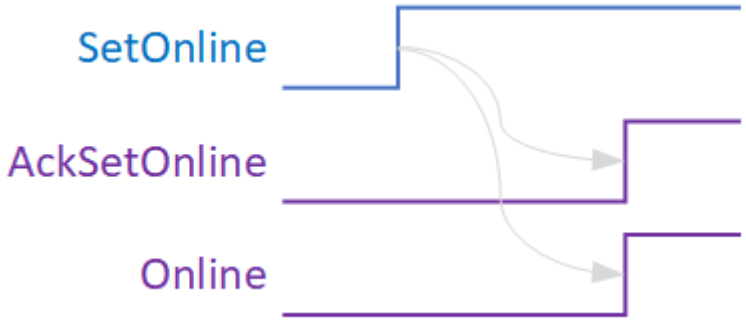


Abb. 203: Das SetOnline Zeitverhalten.

Offline Setzen

Wenn der Vision Sensor Online ist, kann er in den Offline Modus versetzt werden. Um den Vision Sensor Offline zu setzen, löschen Sie das **SetOnline** Bit (Kontroll Byte 0 Bit 0).

Damit der Vision Sensor Offline gehen kann, wird der laufende Zyklus des ausgeführten Jobs beendet und das weitere Ausführen des Jobs wird angehalten. Der Job bleibt geladen. Danach geht der Vision Sensor Offline.

Um zu überprüfen, ob der Vision Sensor Offline gehen konnte, warten Sie bis das **AckSetOnline** Bit (Status Byte 0, Bit 1) gelöscht wurde. Das **Online** Bit (Status Byte 0, Bit 0) wird ebenfalls gelöscht.

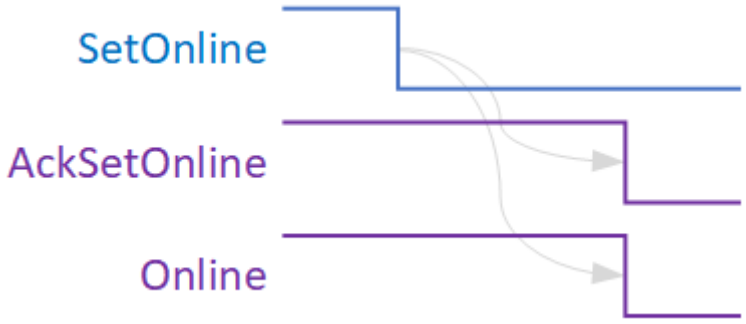


Abb. 204: Das SetOffline Zeitverhalten.

Job Laden

Um einen Job via Profinet oder EtherNet/IP zu laden, muss der Vision Sensor Offline gesetzt werden, die **JobLoadId** Nummer (Kontroll Byte 1, Bits 0..7) muss auf die gewünschte Job-Nummer gesetzt werden und das **JobLoad** Bit (Kontroll Byte 0, Bit 1) muss gesetzt werden. Setzen Sie das **BootJob** Bit (Kontroll Byte 0, Bit 2) wenn Sie den neuen Job zum BootJob machen wollen, andernfalls löschen Sie das **BootJob** Bit. Wenn das Laden beendet ist, wird das **AckJobLoad** Bit (Status Byte 0 Bit 3) gesetzt. Das **JobLoadOk** Bit (Status Byte 0 Bit 4) wird gesetzt, falls der Job geladen werden konnte und gelöscht falls der Job nicht geladen werden konnte. Weiterhin wird die **JobId** Nummer (Status Byte 1, Bits 0..7) auf die neue Job-Id gesetzt. Die SPS sollte dann das **JobLoad** Bit (Kontroll Byte 0, Bit 1) löschen und den Vision Sensor Online setzen.

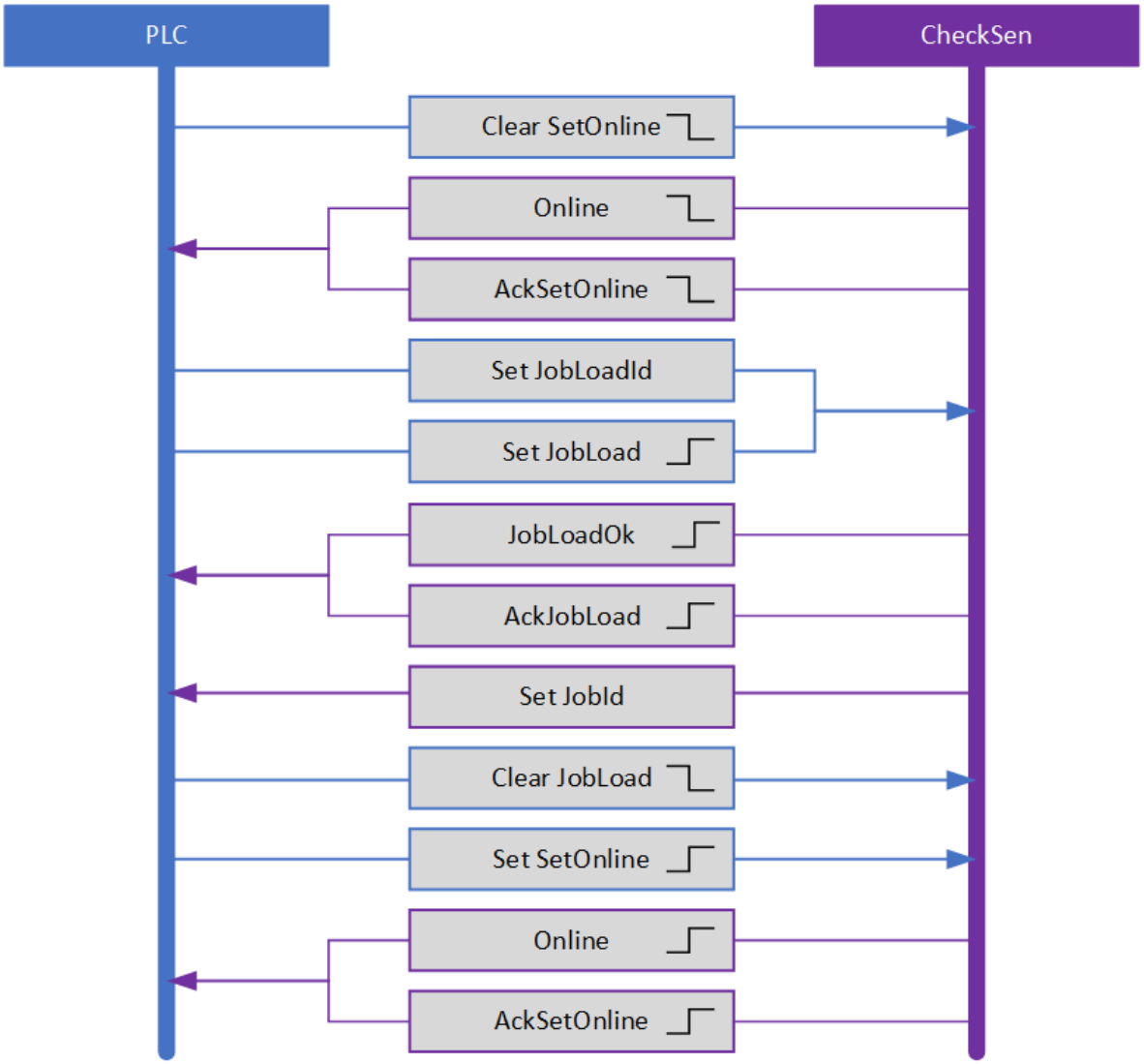


Abb. 205: Die JobLoad Sequenz

Trigger

Mit einem **Trigger** (Kontroll Byte 0, Bit 3) kann die Bildaufnahme über PROFINET oder EtherNet/IP gestartet werden. Profinet oder EtherNet/IP Trigger werden nur ausgeführt, wenn - **Ready** (Status Byte 0, Bit 5) gesetzt ist, und - der Trigger Typ im Akquisition Werkzeug auf `Virtual` steht.

4.6.11 HTML Webserver

Der **CS-6x/ID-6x0**-Sensor hat einen integrierten optionalen Webserver. Der Webserver zeigt eine Webseite und bietet eine ReST API. Die Kommunikation zwischen dem Browser und dem eingebauten Webserver wird auf Port `80` ausgeführt, dem Standardport für HTTP.

Webseite

Auf der **CS-6x/ID-6x0**-Webseite werden Informationen zum Vision-Sensor sowie zu den geladenen Jobs angezeigt. Damit kann der aktive Jobs gewechselt werden und der Bootjob eingestellt werden. Die Webseite kann durch Eingabe der IP-Adresse des Sensors in die Adressleiste des Browsers geöffnet werden, damit wird automatisch die **Info** Seite geöffnet.

Info

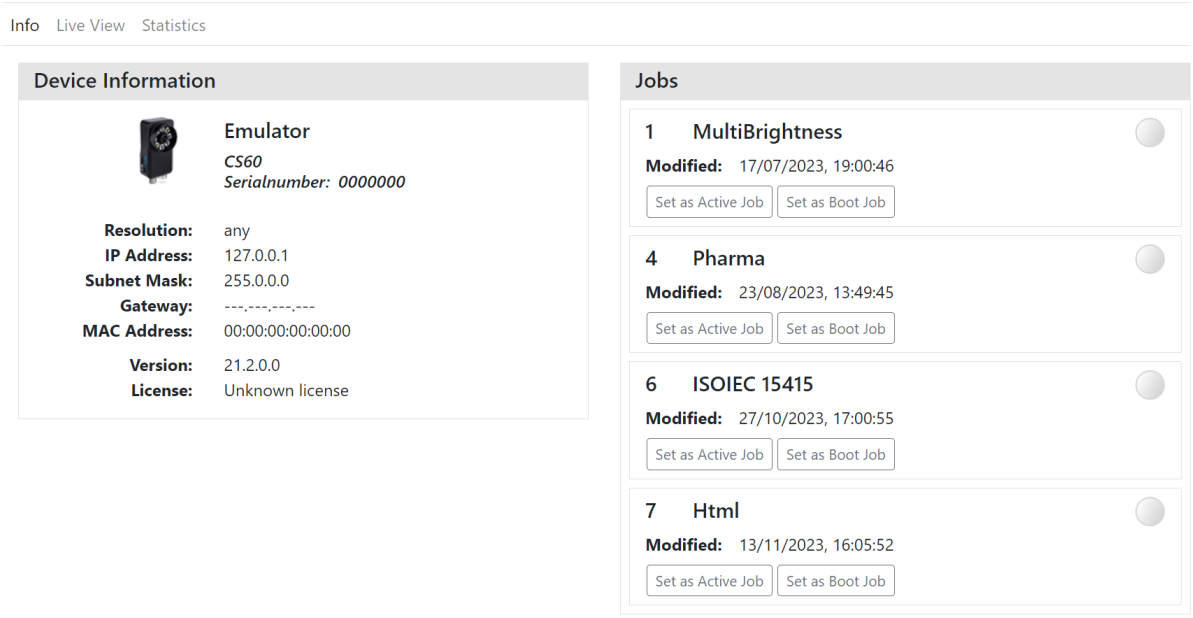


Abb. 206: Die in **CS-6x/ID-6x0 Info** Seite.

Sie können Jobs mit der Schaltfläche **Set Active Job** wechseln und den Bootjob mit der Schaltfläche **Set Boot Job** festlegen.

Live View

Zusätzlich zur **Info** Seite können Sie auch auf **Live View** schalten, um die Inspektionsresultate und das aufgenommene Bild zu betrachten.

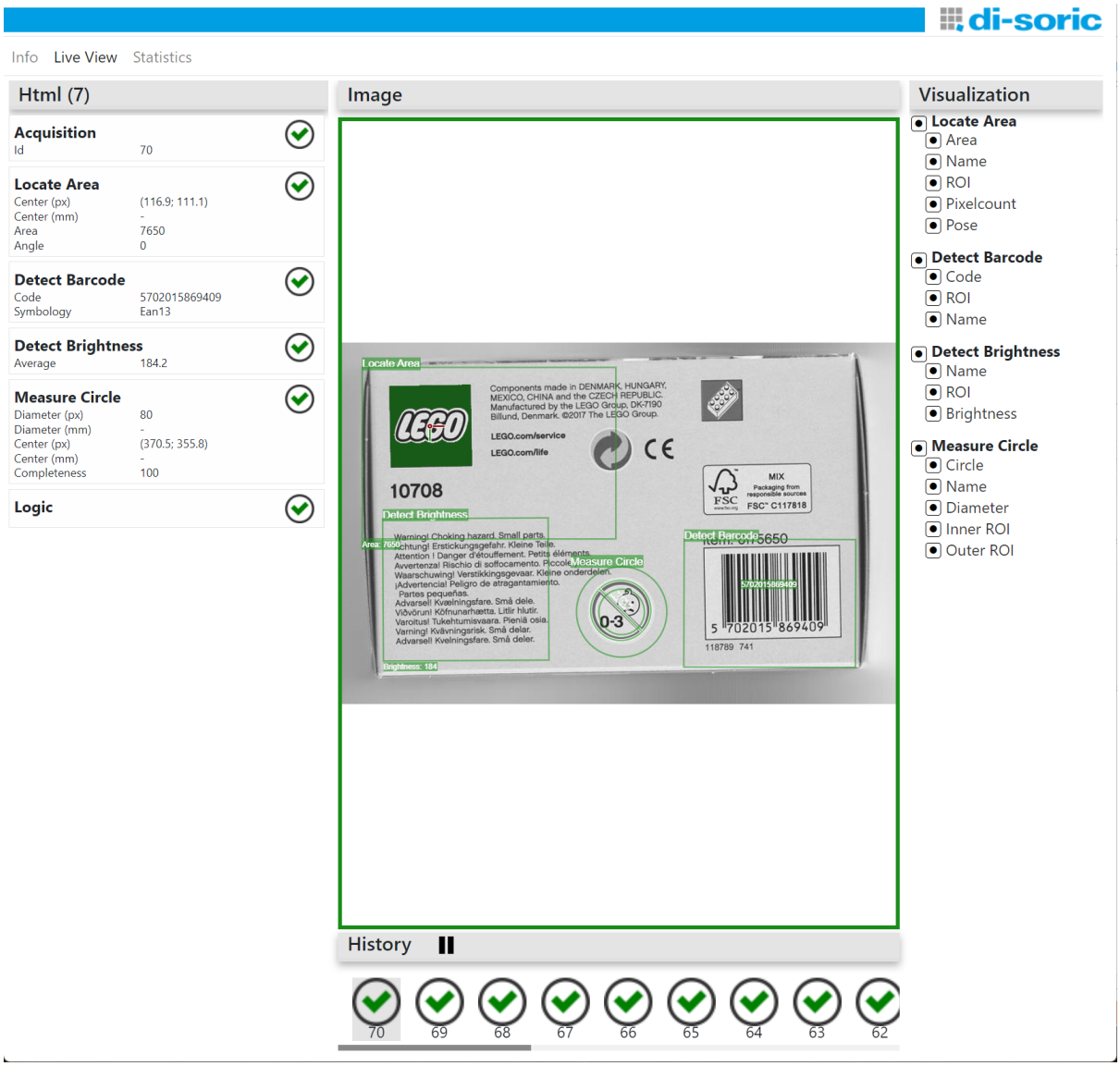


Abb. 207: Die **CS-6x/ID-6x0 Live View** Seite.

Links sehen Sie eine Auflistung der Werkzeuge im Job und deren wichtigste Berechnungsergebnisse.

In der Mitte sehen Sie das Bild mit einer grafischen Visualisierung der Werkzeuge. Sie können die gewünschte Visualisierung anpassen, indem Sie die entsprechenden Elemente auf der rechten Seite an- oder abschalten.

Am unteren Rand sehen Sie die Historie der letzten Inspektionen. Sie können die Historie pausieren und wieder starten, und außerdem auf die einzelnen Inspektionen klicken, um die Visualisierung und die Werte in Ruhe zu studieren.

Statistics

Außerdem können Sie mit der **Statistics** Seite das Zeitverhalten des Jobs und der einzelnen Werkzeuge messen.

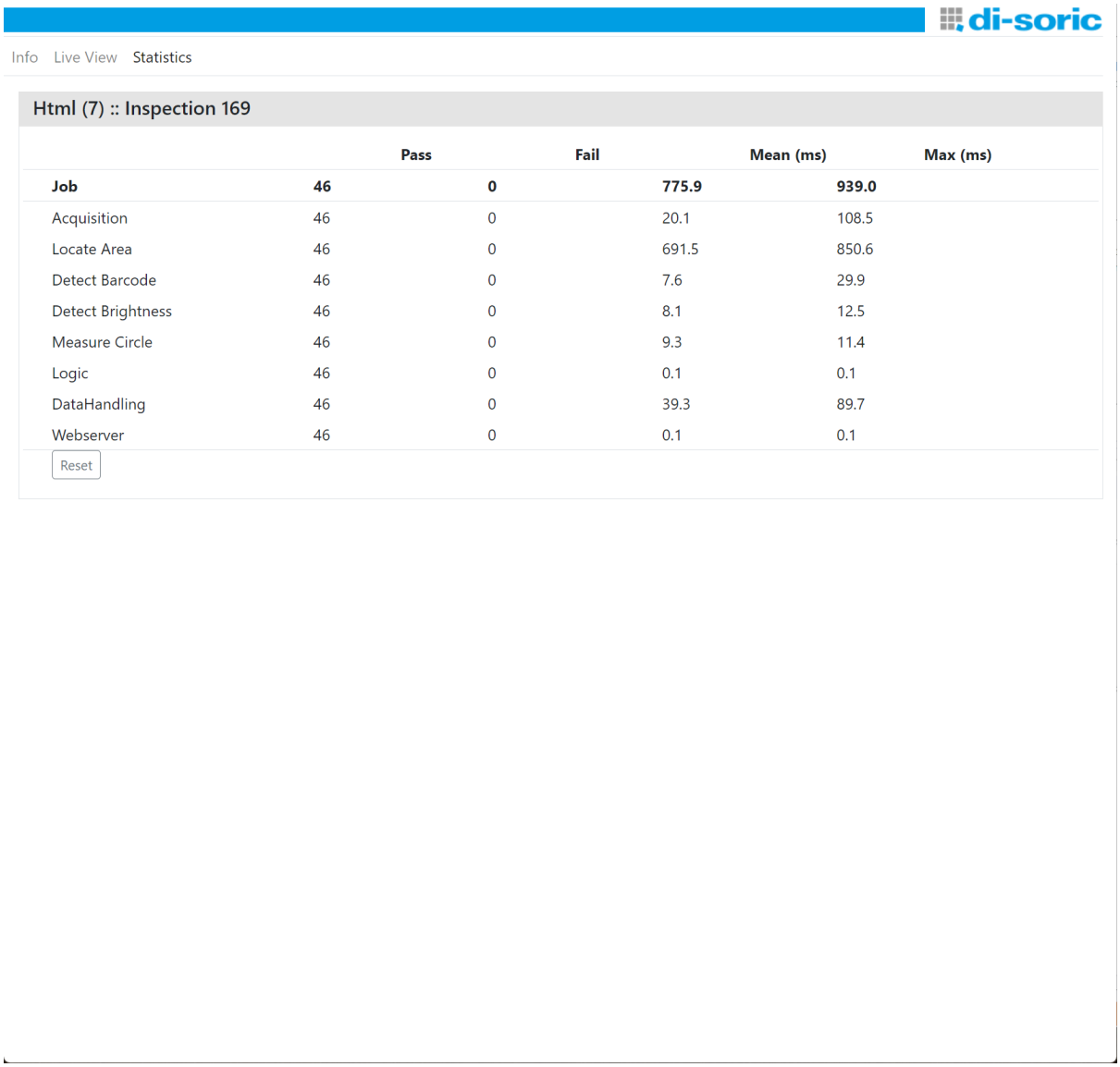


Abb. 208: Die **CS-6x/ID-6x0 Statistics** Seite.

Mit der **Reset** Taste können Sie die Statistic auf Null setzen.

4.6.12 ReST API

Die ReST-API bietet verschiedene Funktionen, um Informationen zum Sensor bereitzustellen, Jobs aufzulisten und Jobs zu wechseln.

Endpoint: `/api/device` HTTP-method: `GET`

Gibt Informationen zum Gerät zurück.

Hier ist eine Beispielantwort im 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`

Listet die Jobs auf dem Gerät auf.

Hier ist eine Beispielantwort:


```
[
    {
        "slot":1,
        "name":"dauertest",
        "lastModified":"2020-01-31T13:59:49",
        "created":"2020-01-31T13:59:49",
        "isBoot":true,
        "isActive":true
    },
    {
        "slot":2,
        "name":"new_job",
        "lastModified":"2020-01-31T13:57:33",
        "created":"2020-01-31T13:57:33",
        "isBoot":false,
        "isActive":false
    }
]
```


Endpoint: `/api/jobs/active` HTTP-method: `GET`

Gibt den aktuell aktiven Job zurück.

Hier ist eine Beispielantwort:

```
{
    "slot":1,
    "name":"dauertest",
    "lastModified":"2020-01-31T13:59:49",
    "created":"2020-01-31T13:59:49",
    "isBoot":true,
    "isActive":true
}
```

Endpoint: `/api/jobs/boot` HTTP-method: `GET`

Gibt den aktuell gesetzten Boot-Job zurück

Beispielantwort: siehe '/api/jobs/active'

Endpoint: `/api/jobs/active?slot=` HTTP-method: `POST`

Legt den aktiven Job fest. Der Job mit dem angegebenen Slot wird aktiviert.

Endpoint: `/api/jobs/boot?slot` HTTP-method: `POST`

Legt den Boot Job fest. Der Job mit dem angegebenen Slot wird als Boot Job festgelegt.

Endpoint: `/api/results/latest` HTTP-method: `GET`

Gibt statistische Informationen für das aktuellste Inspektionsergebnis zurück.

Hier ist eine Beispielantwort:

```
{
"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,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
    "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`

Gibt das aufgenommene Bild zurück.

4.6.13 Job-Wechsel mit den digitalen Eingängen

Die digitalen Eingänge des **CS-6x/ID-6x0** können entweder für den Job-Wechsel oder für Input Linking verwendet werden.

Die beiden Eingänge werden dabei als **Gate** und **Counter** verwendet. Der Eingang 0 (braun) ist das **Gate**, der Eingang 1 (blau) ist der **Counter**.

Wird das **Gate** auf den Pegel 1 (High) gesetzt, so werden am **Counter** die steigenden Flanken gezählt.

Jede steigende Flanke am **Gate** setzt den Zähler zurück. Jede fallende Flanke am **Gate** übernimmt die bis dahin gezählten steigenden Flanken des **Counter** und lädt den entsprechenden Job.

Der **Counter** für den Job-Wechsel darf mit maximal 250 Hz geschaltet werden.

4.7 Hardware

4.7.1 Kabel Pinbelegung

Der **CS-6x/ID-6x0** kommuniziert mit der Außenwelt über zwei Kabel: ein X-kodiertes LAN-Kabel und ein A-kodiertes IO-Kabel.

A-kodierter Stecker CS-6x / ID-6x0

Signal	Kontakt	Farbe	Beschreibung
+UB	7	schwa	
GND	8	grau	
Digital Out 0	11	grau/p	Digitalausgang 0 (vom Logikwerkzeug gesteuert).
Digital Out 1	12	rot/bl	Digitalausgang 1 (vom Logikwerkzeug gesteuert).
Digital Out 2	3	weiß	Digitalausgang 2 (vom Logikwerkzeug gesteuert).
Digital Out 3	4	grün	Digitalausgang 3 (vom Logikwerkzeug gesteuert).
Trigger In	10	violett	Digitaler Trigger (nur zulässig wenn Ready - zwischen der steigenden Flanke von Ready 100 µs warten, bis der nächste Trigger ausgeführt wird).
Digital In 0	1	braun	Digitaleingang 0 (für Job-Wechsel oder für Verknüpfung).
Digital In 1	2	blau	Digitaleingang 1 (für Job-Wechsel oder für Verknüpfung).
Common Interface	6	gelb	
Ready	5	pink	Ready-Ausgang.

A-kodierter Stecker CS-5x / ID-5x0

Signal	Kontakt	Farbe	Beschreibung
+UB	7	schwa	
GND	8	grau	
Digital Out 0	11	grau/p	Digitalausgang 0 (vom Logikwerkzeug gesteuert).
Digital Out 1	12	rot/bla	Digitalausgang 1 (vom Logikwerkzeug gesteuert).
Digital Out 2	3	weiß	Digitalausgang 2 (vom Logikwerkzeug gesteuert).
NC	4	grün	
Trigger In	10	violett	Digitaler Trigger (nur zulässig wenn Ready - zwischen der steigenden Flanke von Ready 100 µs warten, bis der nächste Trigger ausgeführt wird).
NC	1	braun	
Digital In 1	2	blau	Digitaleingang 1 (für Verknüpfung).
NC	6	gelb	
Ready	5	pink	Ready-Ausgang.

4.7.2 LED

Der **CS-6x/ID-6x0** Sensor besitzt drei LEDs auf der Seite.



Abb. 209: Die **CS-6x/ID-6x0** LEDs.

Die LEDs zeigen folgende Information:

Farbe	Beschreibung
grün	Betriebsspannung liegt an.
gelb	Blinkt, um LAN Kommunikation anzuzeigen.
gelb	Zeigt einen externen Trigger.

KAPITEL 5

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

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 Linux kernel

https://bitbucket.org/impuls-imaging/visionsensorpublic

5.20 MessagePack-CSharp

https://github.com/neuecc/MessagePack-CSharp

License

5.21 NetworkView

https://www.codeproject.com/Articles/182683/NetworkView-A-WPF-custom-control-for-visualizing-a

License

5.22 NLog

http://nlog-project.org/ https://github.com/NLog/NLog/

License

5.23 OxyPlot

http://www.oxyplot.org/ https://github.com/oxyplot/oxyplot

License

5.24 Microsoft.Diagnostics.Tracing.TraceEvent

https://github.com/Microsoft/perfview

License

5.25 Prism

https://github.com/PrismLibrary/Prism

License

5.26 signalr.min

https://raw.githubusercontent.com/stefanpenner/es6-promise/master/LICENSE

License

5.27 System.Commandline

https://github.com/dotnet/command-line-api

License

5.28 Sphinx RTD Theme

https://sphinx-rtd-theme.readthedocs.io/en/latest/index.html

License

5.29 SSH.NET

https://github.com/sshnet/SSH.NET

License

5.30 The Helper Trinity

https://github.com/kentcb/TheHelperTrinity/blob/master/LICENSE

License

5.31 Threading Building Blocks

https://www.threadingbuildingblocks.org/

License

5.32 VTK

http://www.vtk.org

License

5.33 WpfLocalizeExtension

https://github.com/XAMLMarkupExtensions/WPFLocalizationExtension/

License

5.34 WPF Task Dialog Wrapper

https://github.com/yadyn/WPF-Task-Dialog

License

5.35 zlib

http://www.zlib.net/

License

nVision-i Endkunden- Lizenzvereinbarung (nVision-i ELV)

Die vorliegenden Bedingungen nehmen Sie zur Kenntnis im Zusammenhang mit dem Angebot der Nutzung einer oder mehrerer Komponenten der Software nVision-i (gesamt und jede Komponente für sich im Folgenden bezeichnet als „Software“). Mit dem Herunterladen, Installieren, Kopieren oder jeder sonstigen Form der Verwendung der Software erklären Sie Ihr Einverständnis mit der Geltung der in dieser Endkunden-Lizenzvereinbarung enthaltenen Bedingungen.

Alle Komponenten der Software nVision-i werden Ihnen von der di-soric Imaging GmbH & Co. KG, Schlingener Strasse 4, DE-86842 Türkheim, Deutschland (im Folgenden: Hersteller) im Vertriebsweg über die di-soric GmbH & Co. KG, Steinbeisstraße 6, DE-73660 Urbach, Deutschland, oder über eine der Auslandsgesellschaften der di-soric -Firmengruppe oder über einen der weltweiten di-soric Vertriebspartner zur Verfügung gestellt, ihrem jeweils zuständigen Vertragspartner (im Folgenden: Individueller Vertragspartner), auf der Basis dessen mit Ihnen vereinbarter individueller nVision-i Nutzungslizenz- Vereinbarung und gegebenenfalls zusammen mit dazugehörigen Medien, einer Produktdokumentation (insbesondere dem nVision-i Benutzerhandbuch) und einer Hardware wie z.B. einem Vision Sensor. Diese individuellen Lizenzbedingungen ihres Individuellen Vertragspartners regeln den lizenzierten Nutzungsumfang und gelten insofern ergänzend neben der vorliegenden Lizenzvereinbarung.

1. Geistiges Eigentum

Die Software ist gesetzlich und vertraglich geschützt als geistiges Eigentum, insbesondere auf Basis des deutschen Urheberrechts, europäischer und internationaler gesetzlichen Bestimmungen und Abkommen. Alle Rechte liegen bei der di-soric Imaging GmbH & Co. KG, Schlingener Strasse 4, DE-86842 Türkheim, Deutschland, Telefon: +49 (0) 82 45 774 96 00, imaging@di-soric.com. Hinsichtlich OpenSource-Software- Bestandteilen wird verwiesen auf die im Zusammenhang mit der Installation eingebundene –Textdatei, das OpenSource Verzeichnis und das nVision-i Benutzerhandbuch; die dort zitierten und enthaltenen anwendbaren Lizenzbedingungen gelten zusätzlich und im Rahmen ihres Anwendungsbereichs vorrangig vor dieser Lizenzvereinbarung.

2. Lizenz

Ihnen wird hiermit das nicht ausschließliche Recht eingeräumt zur bestimmungsgemäßen Verwendung der Software gemäß dem nVision-i Benutzerhandbuch, also zur Evaluation des Einsatzes von di-soric Vision Sensoren in einer automatisierten Produktion und zum Zweck der Kalibrierung und Steuerung von di-soric Vision Sensoren. Ihnen wird die Software in maschinenlesbarer Form (object code) zu Verfügung gestellt. Abgesehen von der Erstellung einer Sicherungskopie und von Kopien für weitere Verwendungen der vorbezeichneten Art sind anderweitige Nutzungen ausgeschlossen. Insbesondere ist jede Änderung, Zurückentwicklung, Übersetzung oder Herauslösung von Teilen der Software unzulässig. Dies gilt auch für jede Form der Dekompilierung mit Ausnahme einer notwendigen Herstellung der Interoperabilität (§ 69i Deutsches UrhG). Ihnen wird auch das Recht zur Übertragung der Software erteilt zum Zweck der vorbezeichneten Verwendungen und vorbehaltlich eines Widerrufs wegen eines wichtigen Grundes. Jede Vervielfältigung oder Übertragung der Software hat die vorliegende Lizenzvereinbarung und Hinweise zum Geistigen Eigentum zu umfassen und deren Befolgung sicherzustellen.

3. Gewährleistung und Software- Pflege

Der Funktionsumfang, die Systemvoraussetzungen und die technischen Spezifikationen der Software bestimmen sich nach dem nVision-i Benutzerhandbuch, welches Ihnen separat angeboten wird und welches Ihnen im Rahmen der Installation zusammen mit der Software zu Verfügung gestellt wird. Der Hersteller bietet im Rahmen der gesetzlichen Bestimmungen Updates an. Gewährleistungsrechte bestimmen sich ausschließlich nach den vertraglichen Regelungen mit ihrem Individuellen Vertragspartner, ebenso wie eine etwaige Softwarepflege.

4. Laufzeit und Rechtsbehelfe

Diese Lizenz gilt für unbestimmte Zeit. Der Hersteller hat das Recht, diese Lizenz zu kündigen mit sofortiger Wirkung bei einem Verstoß gegen diese Lizenzbedingungen. Im Fall einer Kündigung des Herstellers haben Sie jede weitere Nutzung der Software unmittelbar einzustellen und sämtliche Kopien der Software an den Hersteller herauszugeben bzw. stattdessen, auf Verlangen des Herstellers, alle Installationen und Kopien endgültig zu löschen und dies dem Hersteller schriftlich zu bestätigen.

5. Ausfuhrbestimmungen; keine militärische Verwendung

Die Einhaltung sämtlicher für Sie geltender Exportbestimmungen liegt in Ihrer alleinigen Verantwortung, zu deren Beachtung Sie sich hiermit verpflichten. Eine Verwendung zu militärischen Zwecken ist ausgeschlossen. Sie sichern ferner zu, dass Sie keiner Personen- oder Firmengruppe angehören, der die Software nach den deutschen oder den für ihren individuellen Vertragspartner geltenden Export- oder Embargobestimmungen nicht zur Verfügung gestellt werden darf.

6. Schlussbestimmungen

Sollten eine oder mehrere Bestimmungen dieser Lizenzvereinbarung unwirksam sein oder werden, so bleibt die Wirksamkeit der übrigen Bestimmungen hiervon unberührt. Das Verhältnis zwischen den Parteien und alle Streitigkeiten, die sich aus oder im Zusammenhang mit diesem Vertragsverhältnis ergeben, unterliegen dem Recht der Bundesrepublik Deutschland (BRD). Soweit Sie Kaufmann, juristische Person des öffentlichen Rechts oder öffentlich-rechtliches Sondervermögen sind oder keinen allgemeinen Gerichtsstand in der BRD haben oder nach Vertragsschluss ihren Wohnsitz oder gewöhnlichen Aufenthalt aus der BRD verlegen oder ihr Wohnsitz oder gewöhnlicher Aufenthalt im Zeitpunkt einer Klageerhebung nicht bekannt ist, ist ausschließlicher Gerichtsstand für sämtliche Streitigkeiten aus und in Verbindung mit dieser Lizenzvereinbarung der Geschäftssitz des Herstellers; dem Hersteller steht es frei, auch an ihrem Hauptgeschäftssitz Klagen einzureichen und insbesondere auch einstweiligen Rechtsschutz zu beantragen.

Allgemeine Verkaufs- und Lieferbedingungen (AVLB) 07/2017 der di-soric GmbH & Co. KG, DE-73660 Urbach, Deutschland

1 Allgemeines

1.1 Für unsere Lieferungen und Leistungen gelten ausschließlich diese Allgemeinen Verkaufs- und Lieferbedingungen (AVLB) in der jeweils gültigen Fassung. Abweichende, entgegenstehende oder ergänzende allgemeine Geschäftsbedingungen des Bestellers werden – es sei denn, ihre Geltung wird ausdrücklich schriftlich anerkannt – nicht Vertragsbestandteil, selbst wenn wir in Kenntnis solcher Bedingungen Lieferungen oder Leistungen erbringen.

1.2 Im Rahmen einer laufenden Geschäftsbeziehung gelten unsere AVLB bezüglich künftiger Lieferungen und Leistungen auch dann, wenn sie nicht jeweils ausdrücklich vereinbart werden. Unsere AVLB gelten auch für die Anbahnung von Vertragsbeziehungen.

1.3 Wir behalten uns unsere Eigentums- und urheberrechtlichen Verwertungsrechte an von uns zu Verfügung gestellten Kostenvoranschlägen, Zeichnungen und anderen Unterlagen (z.B. Applikationslösungen) uneingeschränkt vor. Derartige Unterlagen dürfen nur mit unserer vorherigen schriftlichen Zustimmung Dritten zugänglich gemacht werden und sind uns zurückzugeben, wird der betreffende Auftrag nicht erteilt. Software wird dem Besteller lediglich zur nicht-ausschließlichen Nutzung in unveränderter Form auf der vereinbarten Hardware mit den vorausgesetzten Parametern und zu den angebotenen Lizenzbedingungen überlassen.

1.4 Der Besteller erklärt sich mit der Speicherung und Auswertung von Bestell- und Bestellerdaten nach den Vorschriften des Bundesdatenschutzgesetzes (BDSG) einverstanden.

2 Angebot

2.1 Unsere Angebote sind freibleibend und lediglich als Aufforderung zur Abgabe einer Bestellung zu verstehen.

2.2 Technische Angaben, auf die direkt oder indirekt in unseren Angeboten Bezug genommen wird, wie insb. Maße, Gewichte, Leistungen/Leistungsmerkmale oder physikalische bzw. stoffliche Beschaffenheit und ähnliche anwendungsbezogene Angaben sind, soweit nicht im Angebot ausdrücklich schriftlich als verbindlich bestätigt, im Rahmen der handelsüblichen Toleranzen nur annähernd maßgeblich.

2.3 Von Angeboten und Prospekten abweichende technische und konstruktive, handelsübliche Änderungen der Liefergegenstände bleiben vorbehalten, soweit sie dem Besteller zumutbar sind und sie die Gebrauchsfähigkeit der Liefergegenstände nicht berühren.

2.4 Wir sind berechtigt, die Leistung entsprechend dem neusten Stand der Technik zu erbringen, auch wenn dieser sich nach Vertragsschluss ändert.

3 Vertragsschluss, Abrufauftrag, Muster

3.1 Der Besteller ist an seine Bestellung vierzehn Tage – gerechnet vom Tage ihres Zugangs – gebunden.

3.2 Der Vertrag kommt durch unsere Auftragsbestätigung oder durch seine Ausführung zustande je nachdem, welches Ereignis früher liegt. Der Inhalt des Vertrages bestimmt sich nach der Auftragsbestätigung, sollte eine solche fehlen, nach unserem Angebot.

3.3 Bei Abrufaufträgen über eine bestimmte Warenmenge ist diese vom Besteller innerhalb der vereinbarten Zeit – fehlt eine solche Vereinbarung, innerhalb von 12 Monaten ab Vertragsschluss – in einer oder mehreren Chargen abzurufen. Wird innerhalb der Abrufzeit nicht die gesamte vereinbarte Liefermenge abgerufen, sind wir von der Vorleistungspflicht befreit und berechtigt, den Kaufpreis für die nicht abgerufene Menge Zug um Zug gegen Leistung zu verlangen. Etwaige Rücktrittsrechte und Schadensersatzansprüche unsererseits bleiben unberührt.

3.4 Muster werden im Inland auf Probe übersandt, wenn nicht ausdrücklich etwas anderes vereinbart ist. Im Export ist der Versand von Mustern als Verkauf zu Sonderpreisen möglich. Im Inland gelieferte Mustergeräte sind innerhalb von zwei Wochen ab Versendungsdatum an uns auf Kosten des Bestellers in unbeschädigtem Zustand und originalverpackt zurückzusenden, falls nicht der Besteller die dem Muster beigefügte Rechnung innerhalb dieser Frist bezahlt. Eine Rücksendung aus dem Ausland ist ausgeschlossen. Wir behalten uns das Eigentum am Muster vor bis zur Zahlung der Rechnung und dem entsprechenden Geldeingang bei uns.

4 Preise

4.1 Unsere Preise verstehen sich „ab Werk" (EXW Incoterms 2010) einschließlich Geräteverpackung, sofern nicht ausdrücklich abweichend vereinbart. Transportverpackung, Transportkosten und -versicherung, Zoll und Entladung werden gesondert pauschal berechnet. Die am Tag der Fälligkeit gültige, gesetzlich vorgeschriebene Mehrwertsteuer wird gesondert in Rechnung gestellt.

4.2 Lieferungen, für die keine Preise vereinbart sind, werden zu den am Tage der Lieferung gültigen Katalogpreisen berechnet.

4.3 Der Mindestbestellwert pro Auftrag beträgt bei Bestellungen aus Deutschland und aus dem Ausland 50,00 €. Bei Bestellungen unter diesem Wert erheben wir eine Bearbeitungsgebühr von 25,00 €.

4.4 Liegt der Liefertermin später als vier Monate nach Vertragsabschluss, so sind beide Parteien berechtigt, die Preise anzupassen entsprechend einer mittlerweile eingetretenen Veränderung der Material- oder Lohnkosten seit Vertragsabschluss um mehr als 10 %; ausgenommen hiervon sind Abrufaufträge gemäß Ziffer 3.3 mit einer Festpreisgarantie.

4.5 Bei Teillieferungen, auch im Rahmen von Abrufaufträgen gemäß Ziffer 3.3, wird jede Lieferung gesondert in Rechnung gestellt.

5 Zahlung, Zahlungsverzug, Aufrechnungsverbot

5.1 Unsere Rechnungen sind – vorbehaltlich Ziffer 3.4 – innerhalb von 30 Tagen ohne Abzug, jeweils ab Rechnungsdatum frei angegebener Zahlstelle zu begleichen.

5.2 Alle Zahlungen haben in Euro zu erfolgen.

5.3 Andere Zahlungsmittel als Bargeld oder Überweisungen werden nur nach vorhergehender Vereinbarung und auch dann nur erfüllungshalber angenommen. Bei Zahlungen im Scheck-Wechsel-Verfahren führt erst die Einlösung des letzten Wechsels zur Erfüllung. Mit der Hereinnahme eines Wechsels ist eine Stundung nur verbunden, wenn dies ausdrücklich und schriftlich vereinbart wird; Zinsen und alle (auch Bank-) Kosten gehen zu Lasten des Bestellers.

5.4 Rabatte oder Skonti können nur abgezogen werden, wenn dies ausdrücklich schriftlich vereinbart wurde. Ein Skontoabzug setzt voraus, dass sämtliche Zahlungen aus dem Auftrag innerhalb der Skontofrist bei uns eingehen.

5.5 Bei Zahlungsverzug des Bestellers können wir Verzugszinsen in Höhe von 8 Prozentpunkten über dem Basiszinssatz berechnen. Unsere Ansprüche auf Ersatz eines tatsächlichen höheren Schadens bleiben unberührt.

5.6 Eine Aufrechnung ist nur mit rechtskräftig festgestellten, unbestrittenen, von uns anerkannten oder sonst fälligen und einredefreien Gegenansprüchen zulässig.

5.7 Ist der Besteller länger als vier Wochen mit seiner Zahlungsverpflichtung in Verzug oder im Falle von Scheck- o. Wechselprotesten oder Vollstreckungsmaßnahmen gegen denselben, so sind wir berechtigt, ausstehende Lieferungen an den Besteller aus anderen Aufträgen oder aus Abrufaufträgen nur noch gegen Vorauskasse auszuführen.

Eine wesentliche Verschlechterung der Vermögensverhältnisse des Bestellers oder sonstige Umstände, die seine Kreditwürdigkeit deutlich beeinträchtigen, führen zur sofortigen Fälligkeit aller unserer Forderungen, selbst wenn uns Akkreditive zur Verfügung gestellt wurden. Dies gilt insbesondere, wenn

- der Besteller länger als vier Wochen mit beträchtlichen Forderungen im Zahlungsverzug ist,
- der Wert der für diesen Vertrag gestellten Sicherheiten nicht unerheblich abnimmt,
- die Vermögensverhältnisse seines persönlich haftenden Gesellschafters sich wesentlich verschlechtern oder erheblich gefährdet sind oder
- dieser stirbt oder wechselt.

In einem solchen Fall sind wir berechtigt, weitere Belieferung von der Zahlung der offenen Rechnungen, Vorkasse oder Sicherheitsleistung abhängig zu machen. Verweigert der Besteller dies, sind wir zum Rücktritt berechtigt.

5.8 Zur Ausübung eines Zurückbehaltungsrechtes ist der Besteller nur befugt, wenn sich sein Gegenanspruch aus derselben (Einzel-) Bestellung bzw. bei Abrufaufträgen aus dem betreffenden (Einzel-) Abruf ergibt. Im Übrigen gelten die Anforderungen an die Gegenansprüche aus Ziffer 5.6 entsprechend.

6 Lieferzeit, Unmöglichkeit

6.1 Der Lauf von vereinbarten Lieferzeiten (Lieferfristen oder Liefertermine) setzt voraus, dass der Besteller alle für die Herstellung und Lieferung der bestellten Ware erforderlichen Informationen, Freigaben, Genehmigungen von Plänen usw. erteilt und die vereinbarten Konstruktionsunterlagen, beizustellenden Materialien oder Prüfvorrichtungen zur Verfügung gestellt hat (im Folgenden: Mitwirkungshandlungen) in einer Form, die es ermöglicht, dass der damit verbundene Zweck von uns erreicht werden kann (im Folgenden: Mitwirkungserfolg), und dass eine etwa vereinbarte Vorauszahlung eingegangen ist. Anderenfalls verlängert bzw. verschiebt sich die Lieferzeit entsprechend des Zeitraums bis zum Eintritt der letzten fehlenden Voraussetzung, es sei denn, wir haben die Verzögerung zu vertreten.

Die Lieferfrist beginnt mit Zugang der Auftragsbestätigung.

Verbindliche Liefertermine bedürfen einer schriftlichen Vereinbarung. Lieferfristen sind eingehalten, wenn bis zu ihrem Ablauf vereinbarungsgemäß entweder die Ware unser Werk verlassen hat, oder die Versandbereitschaft mitgeteilt wurde.

6.2 Richtige und rechtzeitige Selbstbelieferung bleibt vorbehalten.

6.3 Vereinbarte Lieferzeiten verlängern bzw. verschieben sich angemessen bei Maßnahmen im Rahmen von Arbeitskämpfen, insbesondere Streik und Aussperrung und bei Vorliegen von außergewöhnlichen Umständen, die außerhalb unserer Kontrolle liegen und die uns die Vertragserfüllung deutlich erschweren, wie insbesondere

(i) Maschinenbruch, anderweitige Beschädigung von Maschinen oder Maschinenteilen,

(ii) der Eintritt höherer Gewalt, insbesondere Bürgerkrieg, Mobilmachung, Blockaden, Kriegsakte, öffentlicher Aufruhr, Sabotage, Wirbelsturm, Orkan, Hoch-/Niedrigwasser, Erdbeben, Flutwellen und sonstige Naturereignisse, Feuer, Explosion,

(iii) Folgen einer volkswirtschaftlichen Energiekrise, Rohstoff-, Brennstoff-, Hilfsstoff- oder Energiemangel,

(iv) Beschlagnahmen, Export- oder Importverbote, Embargos oder sonstige behördlichen Maßnahmen, die uns oder unsere Zulieferer betreffen.

Entsprechendes gilt, wenn solche Erschwernisse bei Unterlieferanten eintreten. Diese Erschwernisse haben wir auch dann nicht zu vertreten, wenn wir in Lieferverzug sind. Wir teilen dem Besteller den Eintritt solcher Erschwernisse unverzüglich mit, ferner deren Ende nebst voraussichtlichem Liefertermin. Ist uns aus einem der vorgenannten Umstände die Ausführung des Auftrages nicht mehr zuzumuten, können wir vom Vertrag zurücktreten.

Dauert ein solches Erschwernis länger als sechs Monate, wird der Vertrag angemessen angepasst. Lediglich wenn eine Anpassung wirtschaftlich unzumutbar ist, kann die betreffende Partei vom Vertrag zurückzutreten. Wollen wir hiervon Gebrauch machen, informieren wir den Besteller unverzüglich, und zwar selbst dann, wenn bereits eine Verlängerung der Lieferfrist vereinbart wurde.

6.4 Die Beachtung und Durchführung aller relevanten außenwirtschaftlichen Bestimmungen (z.B. Importlizenzen, Devisentransfergenehmigungen usw.) und sonstiger außerhalb Deutschland geltender Vorschriften obliegt dem Besteller, soweit nicht ausdrücklich abweichend vereinbart.

6.5 Im Falle des Lieferverzuges kann der Besteller nach fruchtlos abgelaufener, angemessener Nachfrist vom Vertrag zurückzutreten; im Falle der Unmöglichkeit steht dem Besteller dieses Recht auch ohne Setzung einer Nachfrist zu. Schadensersatzansprüche wegen Verzugs (einschl. etwaiger Folgeschäden) und Ansprüche auf Aufwendungsersatz nach § 284 BGB bestimmen sich ausschließlich nach Ziffer 9 (Haftung).

6.6 Die Begrenzung der Verzugshaftung nach vorstehender Ziffer 6.5 gilt nicht bei einem kaufmännischen Fixhandelskauf nach § 376 des Handelsgesetzbuches (HGB).

7 Versand und Gefahrübergang

7.1 Teillieferungen sind in zumutbarem Umfang zulässig.

7.2 Wir liefern „ab Werk" (EXW Incoterms 2010), sofern nicht ausdrücklich anders vereinbart. Sind wir zur Versendung verpflichtet, wählen wir den Versandweg, die Versandart sowie den Frachtführer, sofern nicht ausdrücklich anders vereinbart. Eine Gewähr für eine vorgeschriebene Verfrachtung übernehmen wir nicht. Lediglich auf Wunsch des Bestellers wird auf seine Kosten die Lieferung versichert gegen Diebstahl, Bruch-, Feuer- und Wasserschaden sowie andere versicherbare Risiken.

7.3 Die Gefahr des zufälligen Untergangs und der zufälligen Verschlechterung geht – auch bei Teillieferungen – spätestens mit Abholung durch den Besteller bzw. Übergabe an das Transportunternehmen auf den Besteller über, auch wenn wir weitere Leistungen wie etwa die Versendungskosten oder die Anlieferung übernommen haben.

7.4 Die Gefahr des zufälligen Unterganges und der zufälligen Beschädigung geht auf den Besteller auch über bei Annahmeverzug bzw. 10 Tage nach Mitteilung der Versandbereitschaft. Auf schriftliche Bestellung und Kosten des Bestellers wird die Ware danach aber versichert.

7.5 Hat der Besteller den verzögerten Versand zu vertreten, können wir diesem die anfallenden Lagerkosten, bei Lagerung im Hause mindestens 0,5 % des Nettowertes der eingelagerten Ware für jeden angefangenen Monat berechnen. Wir sind in einem solchen Fall auch berechtigt, nach Setzung einer angemessenen Nachfrist die Ware anderweitig zu verwenden und den Besteller seiner Bestellung entsprechend mit Ware gleicher Gattung in angemessen verlängerter Frist zu beliefern.

7.6 Der Besteller hat angelieferte Gegenstände unbeschadet etwaiger Gewährleistungsrechte entgegenzunehmen, auch wenn sie unwesentliche Mängel aufweisen.

8 Gewährleistung, Sonderanfertigung

8.1 Unsere Ware erfüllt alle Vorgaben nach den geltenden technischen Normen und Vorschriften. Für ausländische, in Deutschland nicht geltende Vorschriften gilt dies nur bei ausdrücklicher schriftlicher Vereinbarung.

8.2 Wird aus einer bestellten Sonderfertigung nach vom Katalog abweichenden Spezifikationen geliefert, gelten die mit den damit verbundenen Fertigungstoleranzen verbundenen, handelsüblichen Quantitätsabweichungen in einem Rahmen von + / – 5 % als genehmigt.

8.3 Der Besteller hat die von uns gelieferte Ware sofort nach Eingang auf Identitäts-, Qualitäts- und Quantitätsabweichungen zu überprüfen und etwaige offensichtliche Mängel unverzüglich schriftlich zu rügen. Nicht offensichtliche Mängel sind unverzüglich nach Entdeckung schriftlich zu rügen. Bei verspäteter Mängelanzeige gilt die Ware als genehmigt.

8.4 Die bestellte Ware ist mangelhaft, wenn sie nach billigem Ermessen nicht lediglich unerheblich abweicht von der vertraglich vorausgesetzten Beschaffenheit oder Brauchbarkeit infolge eines vor dem Gefahrübergang liegenden Umstands, insbesondere wegen Konstruktions-, Werkstoff- oder Produktionsmängeln.

8.5 Die Gewährleistung ist insbesondere ausgeschlossen bei Nichtbeachtung des Datenblatts, der Montage- und Einbauhinweise sowie der technischen Daten, bei mangelhafter Wartung bzw. Reinigung, unsachgemäßer Verwendung, Abnutzung, unerlaubten chemischen, elektrochemischen, elektrischen oder physikalischen Einflüssen (u. a. durch Flüssigkeiten), unsachgemäßen und ohne vorherige Genehmigung durch uns erfolgten Änderungen oder Instandsetzungsarbeiten seitens des Bestellers oder Dritter.

8.6 Bei Mängeln leisten wir wie folgt Gewähr:

Bei Auftreten von Mängeln innerhalb von 24 Monaten (aktuelle Produkte der Marke di-soric) oder innerhalb von 12 Monaten (Produkte anderer Marken und abgekündigte di-soric-Produkte) ab Gefahrübergang beseitigen wir nach unserer Wahl den Mangel oder liefern eine neue mangelfreie Ware (Nacherfüllung); bei Vorsatz, arglistig verschwiegenem Mangel oder Verletzung einer Garantie gilt die gesetzliche Frist. Wir können die Nacherfüllung verweigern, solange der Besteller seine Zahlungspflichten uns gegenüber nicht in einem Umfang erfüllt, der dem mangelfreien Teil der Leistung entspricht. Ersetzte Teile werden unser Eigentum.

Der Besteller hat uns die zur Nacherfüllung nach billigem Ermessen notwendige Zeit und Gelegenheit einzuräumen. Dies umfasst die Überprüfung der Fehlerursache, gegebenenfalls vor Ort. Aufgrund unberechtigter, weil nicht von uns zu vertretender Reklamation erfolgte Serviceleistungen, d.h. Prüfungen und Vorort-Einsätze, werden dem Besteller in Rechnung gestellt.

Nur in dringenden Fällen der Gefährdung der Betriebssicherheit oder zur Abwehr unverhältnismäßig hoher Schäden bzw. bei Verzug mit der Mängelbeseitigung kann der Besteller die Nachbesserung selbst vornehmen oder vornehmen lassen, wenn wir unverzüglich von derartigen Umständen und diesem Ansinnen in Kenntnis gesetzt worden sind.

Ist die Nachbesserung unmöglich, nach ausreichend gewährten, jedenfalls zweimaligen Nachbesserungsmöglichkeiten fehlgeschlagen oder sind Nachbesserung bzw. Ersatzlieferung nicht oder im Falle eines Fixgeschäfts nach § 376 HGB nur mit schuldhafter Verzögerung ausgeführt worden, kann der Besteller zwischen Minderung und Rücktritt wählen; Rücktritt ist bei nur geringfügiger Vertragswidrigkeit ausgeschlossen.

Ein Anspruch auf Minderung oder ein Rücktritt ist ausgeschlossen, soweit der Nacherfüllungsanspruch verjährt ist.

9 Haftung

9.1 Wir haften wie folgt:

Wir haften bei schuldhafter Verletzung des Lebens, des Körpers oder der Gesundheit, bei Vorsatz oder bei grober Fahrlässigkeit unseres Inhabers, unserer Vertreter oder leitender Angestellter, bei arglistig verschwiegenen Mängeln oder bei Haftung aus Garantie bzw. nach dem Produkthaftungsgesetz.

Wir haften ferner bei schuldhafter Verletzung einer Kardinalpflicht, also einer wesentlichen Vertragspflicht, deren Erfüllung die ordnungsgemäße Durchführung des Vertrages erst ermöglicht und auf deren Einhaltung der Besteller vertrauen darf; hier ist die Haftung jedoch auf den vertragstypischen vorhersehbaren Schaden begrenzt, es sei denn, dass ein Fall des Vorsatzes oder der groben Fahrlässigkeit bzw. eine Verletzung des Lebens, des Körpers oder der Gesundheit vorliegt.

Bei einer schuldhaften Verletzung von Aufklärungs- oder sonstigen Nebenpflichten gelten unter Ausschluss weitergehender Haftung Ziffer 8 sowie die vorstehenden Regelungen entsprechend.

Eine weitergehende Haftung ist ausgeschlossen.

9.2 Die prozessuale Beweislast bleibt von den vorstehenden Haftungsregelungen ebenso unberührt wie ein etwaiger Herstellerregress (§ 478 BGB).

9.3 Soweit unsere Haftung ausgeschlossen oder beschränkt ist, gilt dies auch für die persönliche Haftung unserer Mitarbeiter, Vertreter und Erfüllungsgehilfen.

10 Eigentumsvorbehalt

10.1 Die gelieferte Ware bleibt bis zur vollständigen Bezahlung sämtlicher Forderungen (einschließlich Kosten und Zinsen) aus der Geschäftsverbindung zwischen uns und dem Besteller unser Eigentum. Die Einstellung einzelner Forderungen in eine laufende Rechnung sowie die Saldoziehung und deren Anerkennung berühren den Eigentumsvorbehalt nicht.

10.2 Der Besteller ist zur Weiterveräußerung der Vorbehaltsware im normalen Geschäftsverkehr berechtigt; eine Verpfändung oder Sicherungsübereignung ist ihm jedoch nicht gestattet. Bei einer Verpfändung oder Sicherungsübereignung des Anwartschaftsrechtes hat der Besteller den Sicherungsnehmer auf unser Eigentum hinzuweisen und uns von der Verpfändung oder Sicherungsübereignung unverzüglich zu unterrichten. Der Besteller ist verpflichtet, unsere Rechte beim Weiterverkauf der Vorbehaltsware auf Kredit zu sichern, insbesondere den Eigentumsvorbehalt weiterzugeben.

10.3 Die Forderungen des Bestellers aus der Weiterveräußerung der Vorbehaltsware tritt der Besteller schon jetzt in Höhe des mit uns vereinbarten Kaufpreises an uns ab; wir nehmen diese Abtretung an. Sofern ein Forderungsübergang nach den bei der Weiterveräußerung getroffenen Vereinbarungen nicht möglich ist, ist der Besteller nur mit ausdrücklicher und schriftlicher Zustimmung von uns zur Weiterveräußerung der Vorbehaltsware berechtigt. Dies gilt auch, wenn die Forderung aus der Weiterveräußerung in eine laufende Rechnung einzustellen ist.

Unabhängig davon, ob die Weiterveräußerung in diesem Fall widerrechtlich oder mit unserer Zustimmung erfolgt, tritt der Besteller schon jetzt seinen Anspruch auf ein Saldoguthaben in Höhe des Fakturenwertes an uns ab; wir nehmen diese Abtretung an.

10.4 Ungeachtet der Abtretung und unseres Einziehungsrechtes ist der Besteller zur Einziehung solange berechtigt, als er seinen Verpflichtungen uns gegenüber nachkommt und nicht in Vermögensverfall gerät. Bei einer wesentlichen Verschlechterung der Vermögenslage des Bestellers – insbesondere mit einem Antrag auf Eröffnung des Insolvenzverfahrens – erlischt das Einziehungsrecht.

Auf unser Verlangen hat der Besteller die zur Einziehung erforderlichen Angaben an uns zu machen und dem Schuldner die Abtretung mitzuteilen. Auf unser Verlangen hat der Besteller jederzeit, d. h. auch wenn er selbst zum Einzug berechtigt ist, uns eine von ihm unterzeichnete Abtretungsanzeige auszuhändigen.

10.5 Eine etwaige Be- und Verarbeitung der Vorbehaltsware nimmt der Besteller für uns vor, ohne dass für uns daraus Verpflichtungen entstehen. Bei Verarbeitung, Verbindung, Vermischung oder Vermengung der Vorbehaltsware mit anderen, nicht uns gehörenden Waren steht uns der dabei entstehende Miteigentumsanteil an der neuen Sache im Verhältnis des Fakturenwertes der Vorbehaltsware zu den übrigen verarbeiteten Waren im Zeitpunkt der Verarbeitung, Verbindung, Vermischung oder Vermengung zu.

Erwirbt der Besteller das Alleineigentum an der neuen Sache, sind sich die Vertragspartner schon jetzt darüber einig, dass der Besteller uns im Verhältnis des Fakturenwertes der verarbeiteten bzw. verbundenen, vermischten oder vermengten Vorbehaltsware Miteigentum an der neuen Sache einräumt; eine unentgeltliche Verwahrung dieser neuen Sache für uns durch den Besteller wird schon jetzt vereinbart.

Wird die Vorbehaltsware zusammen mit anderen Waren, und zwar gleich, ob ohne oder nach Verarbeitung, Verbindung, Vermischung oder Vermengung, weiterveräußert, so gilt die oben vereinbarte Vorausabtretung nur in Höhe des Fakturenwertes der Vorbehaltsware, die zusammen mit den anderen Waren weiterveräußert wird.

10.6 Der Besteller ist verpflichtet, unter Eigentumsvorbehalt gelieferte Ware bis zum Erwerb des vollen Eigentums gegen die üblichen Gefahren, insbesondere Diebstahl-, Feuer- und Wasserschäden, zu versichern sowie uns auf Verlangen dies mittels einer Bestätigung der Versicherung nachzuweisen. Der Besteller hat die Vorbehaltsware pfleglich zu behandeln.

10.7 Wir geben Sicherheiten auf Verlangen des Bestellers frei, wenn und soweit der realisierbare Wert der vom Besteller gewährten Sicherheiten die zu sichernde Forderung um mehr als 10 % übersteigt und der Besteller dies verlangt. Die freie Wahl der freizugebenden Sicherheiten obliegt uns.

10.8 Der Besteller ist verpflichtet, unsere Eigentumsrechte entsprechend den Rechtsvorschriften des Bestimmungslandes der Vorbehaltsware sicherzustellen und etwaige, hierzu nötige Mitwirkungshandlungen zu erbringen. Sofern zwingendes Recht der Wirksamkeit des Eigentumsvorbehalts entgegensteht, hat der Besteller auf unser Verlangen anderweitige, vergleichsweise wertbeständige Sicherheiten zu erbringen.

10.9 Der Gefahrübergang gemäß Ziffer 7 bleibt vom Eigentumsvorbehalt unberührt.

11 Beistellung von Materialien und Unterlagen

Werden vom Besteller Teile oder Material zur Verarbeitung oder als Beistellung zur Abwicklung eines Auftrages angeliefert, so wird – wenn nicht ausdrücklich anders vereinbart – keine technische Eingangsprüfung auf nicht offensichtliche Fehler von uns vorgenommen. Sollten sich die beigestellten Waren als ungeeignet erweisen für die vereinbarte Be- oder Verarbeitung, teilen wir dies dem Besteller mit. Dem Besteller obliegt es, für Abhilfe zu schaffen. Er trägt etwaige Mehrkosten. Lieferzeiten verlängern bzw. verschieben sich angemessen.

12 Garantie

Bezugnahmen auf Normen oder andere Werke der anerkannten Regeln der Technik dienen der Warenbeschreibung und stellen noch keine Garantie dar. Garantien im Rechtssinne erhält der Besteller nur, wenn dies ausdrücklich vereinbart ist oder sich aus Werbe- und Verkaufsunterlagen ausdrücklich ergibt. In diesen Fällen leisten wir Garantie nach folgender Maßgabe:

a. Die Garantie bezieht sich ausschließlich auf die Freiheit von Materialfehlern.
b. Garantieleistungen durch uns setzen einen fachgerechten Einbau und Betrieb entsprechend den technischen Datenblättern und Betriebsanleitungen sowie eine sachgemäße Nutzung der Produkte voraus.
c. Die Garantie beschränkt sich auf den kostenlosen Ersatz der fehlerhaften Produkte; Nebenkosten jeglicher Art und Schäden werden von uns nicht übernommen.
d. Die Garantiefrist beginnt zu laufen mit Gefahrübergang.

13 Gewerbliche Schutzrechte

13.1 Es gilt Ziffer 1.3. Der Besteller hat unsere gewerblichen Schutzrechte, insbesondere alle Domain-, Namens-, Marken- und sonstigen Rechte und Markenzeichen sowie Rechte an Know-how zu beachten. Jedwede Verwendung darf nur im Zusammenhang mit unseren Produkten und allenfalls im Rahmen des Handelsüblichen und unter Beachtung der einschlägigen Schutzgesetze erfolgen.

13.2 Wird aus einer bestellten Sonderfertigung nach vom Katalog abweichenden Spezifikationen geliefert, ist allein der Besteller dafür verantwortlich, dass etwaige Schutzrechte Dritter unberührt bleiben. Gegenüber diesbezüglichen Ansprüchen Dritter wegen Verletzung von gewerblichen Schutzrechten (einschließlich Prozesskosten) stellt er uns auf erstes Anfordern frei und wird uns in derartigen Angelegenheiten nach besten Kräften unterstützen.

14 Salvatorische Klausel

Sollte eine Regelung in diesen AVLB oder eine Regelung im Rahmen sonstiger Vereinbarungen unwirksam, ungültig oder undurchsetzbar sein oder werden, so wird die Wirksamkeit aller anderen Regelungen und Vereinbarungen davon nicht berührt. Die Parteien verpflichten sich, eine solche unwirksame, ungültige oder undurchsetzbare Regelung durch eine wirksame Regelung zu ersetzen, deren Gehalt dem wirtschaftlich Gewollten am nächsten kommt.

15 Erfüllungsort, Gerichtsstand, Rechtswahl, Schriftform

15.1 Erfüllungsort für Lieferungen und Zahlungen ist der Sitz unseres Unternehmens in Urbach, Deutschland.

15.2 Rechtsstreitigkeiten sind bei dem für uns zuständigen deutschen Gericht durchzuführen. Wir haben jedoch das Recht, den Besteller auch an seinem Gerichtsstand gerichtlich in Anspruch zu nehmen.

15.3 Auf die gegenseitigen Rechtsbeziehungen findet ausschließlich das Recht der Bundesrepublik Deutschland Anwendung. Das einheitliche UN-Kaufrecht (CISG) gilt nicht.

15.4 Änderungen oder Ergänzungen dieser Vereinbarung sind nur gültig, wenn sie von uns schriftlich bestätigt wurden; dies gilt auch bezüglich dieses Schriftformerfordernisses.

nVision-i Nutzungslizenz- Vereinbarung (nVision-i NLV)

Die Komponenten der Software nVision-i (einzeln und als Gesamtheit im Folgenden die Software) werden als Download oder als sog. embedded software zusammen mit einem di-soric Vision Sensor von der di-soric GmbH & Co. KG, Steinbeisstraße 6, DE-73660 Urbach, Deutschland (im Folgenden: di-soric) oder über eine der Auslandsgesellschaften der di-soric -Firmengruppe als direkte Lizenzgeber oder über andere Lizenznehmer, insbesondere über einen der weltweiten di-soric Vertriebspartner vertrieben. Je nach Ihrer Bezugsquelle ist der Inhaber der Website Ihres Downloads oder der Ihr Zulieferer einer Version als embedded software, jeweils gegebenenfalls zusammen mit dazugehörigen Medien wie einer Produktdokumentation (insbesondere dem nVision-i Benutzerhandbuch), Ihr jeweils verantwortlicher Vertragspartner (im Folgenden: Ihr Individueller Vertragspartner). Die folgenden Regeln stellen die konkret mit Ihnen vereinbarten Lizenzbedingungen ihres Individuellen Vertragspartners dar.

Ihr nach dieser nVision-i NLV lizenzierter konkreter Nutzungsumfang bestimmt sich ferner, subsidiär, nach den Allgemeinen Verkaufs- und Lieferbedingungen (AVLB) der di-soric in der jeweils gültigen Fassung, wie in der Fassung AVLB 07/2017 hier ebenfalls parallel zu Verfügung gestellt und im Internet jeweils in aktueller Fassung lesbar und downloadbar, bzw. gegebenenfalls statt der AVLB nach Allgemeinen Vertriebsbedingungen Ihres Individuellen Vertragspartners, sofern diese den nVision-i ELV und diesen nVision-i NLV nicht widersprechen. Etwaige allgemeine Geschäfts- oder Einkaufsbedingungen Ihrerseits finden keine Anwendung.

1. Geistiges Eigentum

Die Software ist gesetzlich und vertraglich geschützt als geistiges Eigentum, insbesondere auch auf Basis des deutschen Urheberrechts, europäischer und internationaler gesetzlichen Bestimmungen und Abkommen.

Zusätzlich zu den vorliegenden Regelungen und insofern vorrangig gilt die Ihnen zusammen mit dieser nVision-i NLV zu Verfügung gestellte nVision-i Endkunden- Lizenzvereinbarung (nVision-i ELV).

Zusätzlich zu den vorliegenden Regelungen und insofern auch vorrangig gelten für OpenSource- Software- Bestandteile, wie in der nVision-i ELV erläutert, die im Zusammenhang mit der Installation eingebundene OpenSource –Textdatei, das OpenSource Verzeichnis, das nVision-i Benutzerhandbuch und insbesondere im Rahmen ihres Anwendungsbereichs vorrangig vor dieser Lizenzvereinbarung die dort zitierten und enthaltenen, anwendbaren Lizenzbedingungen.

2. Lizenz / Eingeräumte Rechte

Ihnen wird hiermit das nicht ausschließliche Recht eingeräumt zur bestimmungsgemäßen Verwendung der Software gemäß dem nVision-i Benutzerhandbuch, also zur Evaluation des Einsatzes von di-soric Vision Sensoren in einer automatisierten Produktion und zum Zweck der Kalibrierung und Steuerung von di-soric Vision Sensoren. Ihnen wird die Software in maschinenlesbarer Form (object code) zu Verfügung gestellt. Es besteht kein Anspruch auf Zugriff auf den Quellcode oder Herausgabe des Quellcodes; § 69e des Deutschen Urheberrechtsgesetzes (UrhG) bleibt unberührt. Abgesehen von der Erstellung einer Sicherungskopie und von Kopien für weitere Verwendungen der vorbezeichneten Art sind anderweitige Nutzungen ausgeschlossen. Insbesondere ist jede Änderung, Zurückentwicklung, Übersetzung oder Herauslösung von Teilen der Software unzulässig. Dies gilt auch für jede Form der Dekompilierung mit Ausnahme einer notwendigen Herstellung der Interoperabilität (§ 69i Deutsches UrhG).

Ihnen wird auch das Recht zur Übertragung der Software erteilt zum Zweck der vorbezeichneten Verwendungen und vorbehaltlich eines Widerrufs wegen eines wichtigen Grundes. Haben Sie die Software als embedded software bezogen, sind die Nutzung der Software bei Übertragung der Hardware zusammen mit der embedded software vollständig aufzugeben und sämtliche angefertigten Kopien zu löschen. Jede Vervielfältigung oder Übertragung der Software hat die vorliegende Lizenzvereinbarung und Hinweise zum Geistigen Eigentum zu umfassen und deren Befolgung sicherzustellen. Urhebervermerke, Seriennummern und anderweitige Merkmale zur Programmidentifikation dürfen nicht entfernt oder verändert werden.

Andere Leistungen als die Einräumung dieser Nutzungsrechte, insb. Installations- oder Konfigurationsleistungen, sind nicht Gegenstand dieser nVision-i NLV.

3. Pflichten als Anwender

Die Einrichtung einer funktionsfähigen und für die Funktionen der Software ausreichend dimensionierten Hard- und Softwareumgebung liegt in Ihrer alleinigen Verantwortung. Dies umfasst die Beachtung der Hinweise für die Installation und den Betrieb der Software in dem nVision-i Benutzerhandbuch und die regelmäßige Kontrolle der Website von di-soric oder Ihres Individuellen Vertragspartners hinsichtlich etwaiger neuer Hinweise zum Betrieb der Software oder zu Updates. Dies umfasst auch angemessene Vorkehrungen für den Fall, dass die Software ganz oder teilweise nicht ordnungsgemäß arbeitet, z.B. durch regelmäßige Datensicherung, Störungsdiagnose und regelmäßige Überprüfung der Datenverarbeitungsergebnisse.

Der Einsatz jeder Steuerungssoftware erfordert vor deren Einsatz im laufenden Betrieb eine gründliche Untersuchung auf Mangelfreiheit und auf Verwendbarkeit in der bestehenden Hard- und Softwarekonfiguration. Dies gilt auch für die Software.

Sie sind verpflichtet, die Software gesichert zu speichern, zu installieren, zu betreiben und aufzubewahren, sodass ein unberechtigter Zugang und eine unzulässiges Kopieren verhindert werden. Insbesondere sind etwaige Kopien der Software und Zugangsdaten an einem in geeigneter Form geschützten Ort zu verwahren. Es ist untersagt, technische Maßnahmen zum Schutz der Software zu umgehen.

4. Gewährleistung und Softwarepflege

Der Funktionsumfang, die Systemvoraussetzungen und die technischen Spezifikationen der Software (gesamt im Folgenden: die Spezifikationen) bestimmen sich nach dem nVision-i Benutzerhandbuch, welches Ihnen separat zum Lesen und Download angeboten wird und welches Ihnen im Rahmen der Installation zusammen mit der Software zu Verfügung gestellt wird. Updates werden grundsätzlich im Rahmen der gesetzlichen Bestimmungen angeboten.

Gewährleistungsrechte bestimmen sich ausschließlich nach den hier folgenden vertraglichen Regelungen mit Ihrem Individuellen Vertragspartner, ebenso auch eine etwaige Softwarepflege:

Ihr Individueller Vertragspartner gewährleistet während der je nach Fall individualvertraglich, in den AVLB oder in seinen Allgemeinen Vertriebsbedingungen vereinbarten Gewährleistungsfrist, beginnend ab Download bzw. sonst ab Übergabe, dass die Funktionsweise der Software im Wesentlichen den Spezifikationen der Software entspricht. Besteht ein Pflegevertrag, bestimmen sich Mängelbeseitigungsfristen nach den dort vereinbarten Fristen. In den Spezifikationen oder in der Software- Dokumentation festgelegte Angaben stellen keine Garantien im Rechtssinn dar, es sei denn, sie sind ausdrücklich als solche bezeichnet.

Bei Vorliegen eines Gewährleistungsfalls wird Ihr Individueller Vertragspartner nach eigener Wahl Mängel der Software durch Fehlerbehebung, Ersatzbeschaffung, Updates oder Releases einer neuen Version der Software beheben. Als Sachmangel gilt eine Abweichung von den Spezifikationen, wenn sie nachgewiesen wird und reproduzierbar ist. Bei Rechtsmängeln erhalten Sie nach Wahl Ihres Individuellen Vertragspartners eine rechtlich einwandfreie Nutzungsmöglichkeit der Software oder eine derart geänderte Version, dass die Software keine Rechte Dritter mehr verletzt. Ihm stehen zwei Versuche zur Nachbesserung zu. Gelingt es ihm nicht, die Mängel innerhalb eines angemessenen Zeitraums zu beheben, sind Sie zur Reduzierung der Vergütung („Minderung“) berechtigt. Sollte es sich um einen wesentlichen Mangel handeln, sind Sie statt zur Minderung auch zum Rücktritt berechtigt.

Ihnen stehen keine Gewährleistungsansprüche zu,

- wenn Sie die Software nicht bestimmungsgemäß oder missbräuchlich nutzen, oder
- wenn Sie die Software ohne vorherige schriftliche Zustimmung Ihres Individuellen Vertragspartners modifiziert oder verändert haben, oder
- wenn Probleme oder Fehler darauf beruhen, dass die Software mit Programmen genutzt wurde, die mit der Software nicht kompatibel sind, es sei denn, Sie weisen nach, dass der Mangel auf die Software zurückzuführen ist.

Sofern Ihnen aufgrund vertraglicher oder gesetzlicher Vorschriften ein Gewährleistungsanspruch auf Schadenersatz oder Ersatz vergeblicher Aufwendungen zusteht, unterliegt dieser den nachstehenden Haftungsregelungen.

5. Haftung

Ist di-soric Ihr Individueller Vertragspartner, gelten die Haftungsregelungen in den jeweils gültigen AVLB. Im Übrigen haftet Ihr Individueller Vertragspartner, sofern mit Ihnen keine abweichenden Haftungsausschlüsse und –begrenzungen individuell oder in Allgemeinen Vertriebsbedingungen vereinbart sind, wie folgt:

Ihr Individueller Vertragspartner haftet bei schuldhafter Verletzung des Lebens, des Körpers oder der Gesundheit, bei Vorsatz oder bei grober Fahrlässigkeit seines Inhabers, seiner Vertreter oder leitenden Angestellten, bei arglistig verschwiegenen Mängeln oder bei Haftung aus Garantie bzw. nach dem Produkthaftungsgesetz.

Er haftet ferner bei schuldhafter Verletzung einer Kardinalpflicht, also einer wesentlichen Vertragspflicht, deren Erfüllung die ordnungsgemäße Durchführung des Vertrages erst ermöglicht und auf deren Einhaltung Sie vertrauen dürfen; hier ist die Haftung jedoch auf den vertragstypischen vorhersehbaren Schaden begrenzt, es sei denn, dass ein Fall des Vorsatzes oder der groben Fahrlässigkeit bzw. eine Verletzung des Lebens, des Körpers oder der Gesundheit vorliegt.

Bei einer schuldhaften Verletzung von Aufklärungs- oder sonstigen Nebenpflichten gelten unter Ausschluss weitergehender Haftung Ziffer 4 sowie die vorstehenden Regelungen entsprechend. Ihr Individueller Vertragspartner haftet insbesondere auch nicht für Schäden, die auf eine ungeeignete, unsachgemäße oder nicht bestimmungsgemäße Nutzung der Software zurückzuführen sind.

Eine weitergehende Haftung ist ausgeschlossen. Die prozessuale Beweislast bleibt von den vorstehenden Haftungsregelungen ebenso unberührt wie ein etwaiger Herstellerregress (§ 478 BGB). Soweit die Haftung hiernach ausgeschlossen oder beschränkt ist, gilt dies auch für die persönliche Haftung der Mitarbeiter, Vertreter und Erfüllungsgehilfen Ihres Individuellen Vertragspartners.

6. Laufzeit und Rechtsbehelfe

Diese Lizenz gilt für unbestimmte Zeit. Ihr Individueller Vertragspartner hat das Recht, diese Lizenz zu kündigen mit sofortiger Wirkung bei einem Verstoß gegen diese Lizenzbedingungen. Im Fall einer Kündigung Ihres Individuellen Vertragspartners haben Sie jede weitere Nutzung der Software unmittelbar einzustellen und sämtliche Kopien der Software an Ihren Individuellen Vertragspartner herauszugeben bzw. stattdessen, auf Verlangen Ihres Individuellen Vertragspartners, alle Installationen und Kopien endgültig zu löschen und dies Ihrem Individuellen Vertragspartner schriftlich zu bestätigen.

7. Ausfuhrbestimmungen; keine militärische Verwendung

Die Einhaltung sämtlicher für Sie geltender Exportbestimmungen liegt in Ihrer alleinigen Verantwortung, zu deren Beachtung Sie sich hiermit verpflichten. Eine Verwendung zu militärischen Zwecken ist ausgeschlossen. Sie sichern ferner zu, dass Sie keiner Personen- oder Firmengruppe angehören, der die Software nach den deutschen oder den für Ihren Individuellen Vertragspartner geltenden Export- oder Embargobestimmungen nicht zur Verfügung gestellt werden darf.

8. Schlussbestimmungen

Sollten eine oder mehrere Bestimmungen dieser nVision-i NLV unwirksam sein oder werden, so bleibt die Wirksamkeit der übrigen Bestimmungen hiervon unberührt. Das Rechtsverhältnis zwischen Ihnen und Ihrem Individuellen Vertragspartner und alle Streitigkeiten, die sich aus oder im Zusammenhang mit diesem Vertragsverhältnis ergeben, unterliegen dem Recht, welchem auch der Vertrag mit Ihrem Individuellen Vertragspartner unterliegt, in Zweifelsfällen dem Recht der Bundesrepublik Deutschland (BRD). Soweit Sie Kaufmann, juristische Person des öffentlichen Rechts oder öffentlich-rechtliches Sondervermögen sind oder keinen allgemeinen Gerichtsstand im Land des Hauptsitzes Ihres Individuellen Vertragspartners haben oder nach Vertragsschluss ihren Wohnsitz oder gewöhnlichen Aufenthalt aus diesem Land verlegen oder Ihr Wohnsitz oder gewöhnlicher Aufenthalt im Zeitpunkt einer Klageerhebung nicht bekannt ist, ist ausschließlicher Gerichtsstand für sämtliche Streitigkeiten aus und in Verbindung mit dieser nVision-i NLV der Geschäftssitz Ihres Individuellen Vertragspartners; Ihrem Individuellen Vertragspartner steht es frei, auch an ihrem Hauptgeschäftssitz Klagen einzureichen und insbesondere auch einstweiligen Rechtsschutz zu beantragen.

SOLUTIONS. CLEVER. PRACTICAL.

© di-soric | Alle Angaben ohne Gewähr. Irrtümer, Druckfehler und technische Änderungen vorbehalten. | 600007-0000DE · USE-nVISION-i · 202505

di-soric Hauptsitz

Deutschland: di-soric GmbH & Co. KG | Steinbeisstrasse 6 | 73660 Urbach
Tel +49 71 81 98 79-0 | Fax +49 71 81 98 79-179 | info@di-soric.com

di-soric Niederlassungen

China: di-soric Industrial Automation (Suzhou) Co. Ltd. | Tel +86 199 5127 5458 | info@di-soric.cn
Frankreich: di-soric SAS | Tel +33 476 61 65 90 | info.fr@di-soric.com
Österreich: di-soric GmbH & Co. KG | Tel +43 7228 72 366 | info.at@di-soric.com
Singapur: di-soric Pte. Ltd. | Tel +65 6694 7866 | info.sg@di-soric.com

Weitere Informationen unter: www.di-soric.com/international

www.di-soric.com