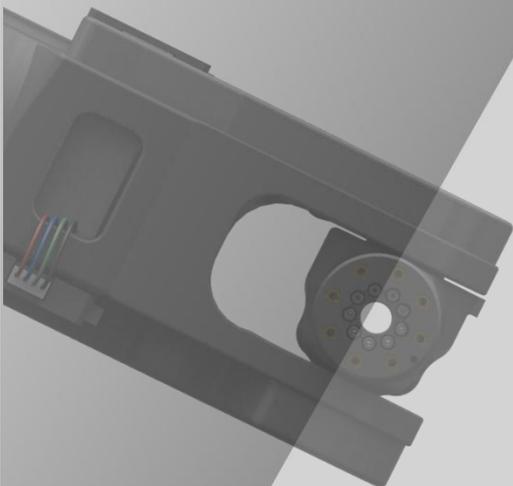
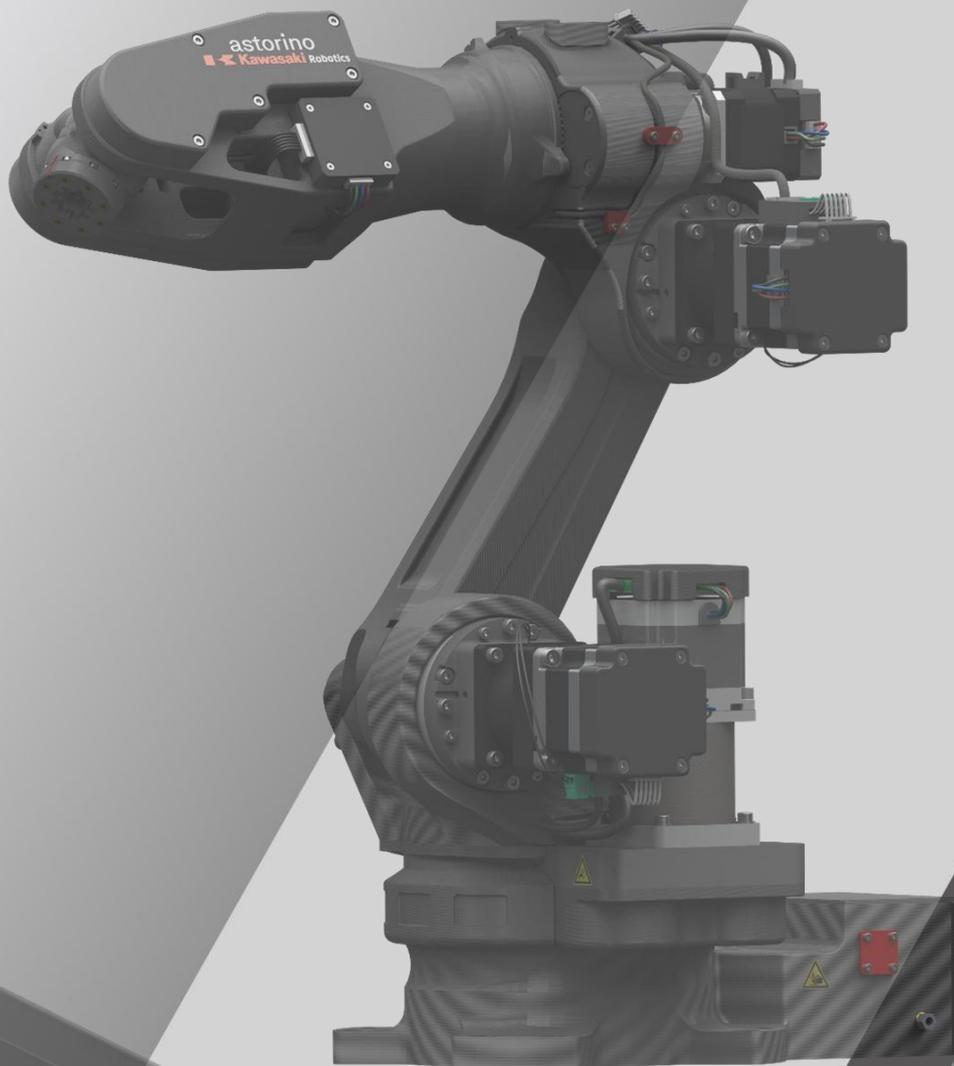


Astorino

Betriebsanleitung



Einführung

Diese Anleitung beschreibt die Bedienung eines 6-Achsen-Bildungsroboters "Kawasaki Robotics Astorino" und der mit ihm verbundenen "Astorino"-Software, die Teil des Lieferumfangs ist. **Die Anleitung gilt ab der Version Firmware 3.8.4 und der Softwareversion astorino 1.9.2.**

ASTORINO ist ein Bildungsroboter, der eigens für Bildungsstellen und -institutionen entworfen worden ist. Schüler und Studenten können ASTORINO nutzen, um Automatisierung und Robotisierung der industriellen Prozessen in Praxis zu lernen.

Bei weiteren Fragen kontaktieren Sie bitte die technische Hilfe von ASTOR.

Kontakt:

Technische Hilfe ASTOR, Abteilung Robotics-Zubehör

E-Mail: Astorino@astor.com.pl

ASTORINO Betriebsanleitung

1. Die dem Astorino beigefügte "Astorino"-Software, besitzt die Lizenz ausschließlich zur Nutzung mit diesem Roboter und darf in keinem anderen Umfeld weder verwendet, noch kopiert, noch verbreitet werden.
2. ASTOR und Kawasaki Robotics haften nicht für Unfälle, Schäden und/oder Probleme, die mit falscher Benutzung des Astorino-Roboters verursacht sind.
3. ASTOR und Kawasaki Robotics behalten sich das Recht vor, diese Anleitung ohne vorherige Mitteilung zu verändern, zu korrigieren oder zu aktualisieren.
4. Diese Anleitung kann ohne vorherige schriftliche Zustimmung von ASTOR und Kawasaki Robotics weder gänzlich noch teilweise gedruckt oder kopiert werden.
5. Bewahren Sie diese Anleitung an einem sicheren Ort griffbereit auf, so dass sie jederzeit genutzt werden kann. Sollte die Anleitung verloren gehen oder ernsthaft beschädigt werden, nehmen Sie bitte Kontakt mit ASTOR auf.

Copyright © 2024 ASTOR & Kawasaki Robotics GmbH.

Alle Rechte vorbehalten.

Symbole

Elemente, die in dieser Anleitung besonders beachtet werden müssen, sind mit den folgenden Symbolen gekennzeichnet.

Die richtige Funktion des Roboters ist sicherzustellen und den Verletzungen oder Vermögensschäden vorzubeugen, indem man nach Sicherheitsanweisungen in Feldern mit diesen Symbolen vorgeht.



Warnhinweis

Wird die nachfolgende Anleitung nicht befolgt, können Verletzungen entstehen.

[VORSICHT]

Dadurch werden Vorsichtsmaßnahmen betreffend die Spezifikation des Roboters, der Bedienung, des Lernprozesses und der Wartung bestimmt.



Warnhinweis

- 1. Die Genauigkeit und die Wirksamkeit der Diagramme, der Verfahren und der Klarstellungen, die in diesem Handbuch enthalten sind, kann nicht mit absoluter Sicherheit bestätigt werden. Sollten jegliche Probleme auftreten, ist mit der Kawasaki Robotics GmbH oder mit der Firma Astor unter der vorgenannten Adresse Kontakt aufzunehmen.**
- 2. Zur Gewissheit, dass alle Arbeiten sicher durchgeführt werden, ist die Anleitung mit Textverständnis zu lesen. Ferner hat man sich mit allen geltenden Rechtsvorschriften, Regelungen und verbundenen Materialien sowie mit Erklärungen zur Sicherheit, die in jedem Kapitel beschrieben sind, in Kenntnis zu setzen. Bereiten Sie entsprechende Sicherheitsmittel und Verfahren auf den realen Arbeitsablauf vor.**

Paraphrasen

In diesem Handbuch werden folgende Schreibweisen angewandt:

- Beim Drücken einer konkreten Taste ist diese in geschweifte Klammern gesetzt, z. B. <F1> lub <Enter>.
- Beim Drücken eines Dialogfeldes oder einer Symbolleiste ist die Bezeichnung der Taste in eckige Klammern gesetzt, z. B. [OK] oder [Reset].
- Die Wahlfelder sind mit Quadratfeld gekennzeichnet. Sind diese aktiviert, befindet sich im - Symbol auch der kleine Wahl-Tag .

ASTORINO Betriebsanleitung

Spis treści

Einführung	Błąd! Nie zdefiniowano zakładki.
Symbole.....	1
Paraphrasen	2
Inhaltsverzeichnis	3
1 Bezeichnungen in dieser Anleitung	7
1.1 Beschreibung des ASTORINO-Roboters.....	7
2 Technische Spezifikation	8
3 Lieferumfang.....	Błąd! Nie zdefiniowano zakładki.
4 Bewegungsbereich	Błąd! Nie zdefiniowano zakładki.
5 Montageabmessungen	11
6 Zubehör-Montagepunkte.....	12
7 Lastdiagramm	Błąd! Nie zdefiniowano zakładki.
8 Elektrische Anschlüsse.....	16
9 Sicherheitshinweise.....	Błąd! Nie zdefiniowano zakładki.
10 Auspacken und Inbetriebnahme	18
10.1 Anschließen von Zubehör	Błąd! Nie zdefiniowano zakładki.
10.2 Systemanforderungen.....	20
10.3 Installation der Steuereinheit.....	20
10.4 Installation der Astorino-Software	21
10.5 Vorbereitung von Astorino auf den Betrieb.....	22
11 Koordinatensysteme	Błąd! Nie zdefiniowano zakładki.
11.1 BASE-Koordinatensystem	25
11.2 JOINT-Koordinatensystem	26
12 Betriebsarten des Roboters.....	27
12.1 Lehrmodus (Teach)	27
12.2 Wiederholmodus (Repeat)	27
13 Manuelle Bedienung des Roboters	28
13.1 JOINT (Punkt-zu-Punkt-Bewegung)	28
13.2 BASE (Bewegung in Bezug aufs Grundkoordinatensystem)	29
13.3 TOOL (Bewegung in Bezug aufs Werkzeugsystem)	30
13.4 WORK.....	32
14 Roboterbewegung	33
14.1 Lineare Interpolation	34
14.2 Punkt-zu-Punkt-Interpolation.....	34
14.3 Kreisinterpolation	35
15 Astorino-Software	36
15.1 Grundlegende Informationen	36
15.2 Bereich Status	37
15.3 Control-Karte / Steuerung	38
15.3.1 Bereich Motoren / Antriebe	39

ASTORINO Betriebsanleitung

15.3.2	Bereich Control / Steuerung.....	41
15.3.3	Bereich Connection / Verbindung.....	41
15.4	JOG-Karte	42
15.4.1	Jogging-Bereich.....	45
15.4.2	Bereich Current Position / Aktuelle Position.....	47
15.4.3	Bereich Step - Teach / Schritt-Einstellungen	47
15.4.4	Bereich Teach Point / Punkt Einteachen	48
15.4.5	Bereich Execute Motion Command / Bewegungsbefehl ausführen	48
15.5	Karte Points / Punkte	49
15.6	Karte Home/Tool / Hausposition/Werkzeug.....	50
15.6.1	Bereich Tool.....	51
15.6.2	Bereich Wizard / Kreator	51
15.6.3	WORK	52
15.6.4	WORK WIZARD	52
15.6.5	Bereich Power OFF position / Position Abschalten	53
15.6.6	Bereich Zeroing order / Auf null setzen - Reihenfolge	53
15.7	Karte Moving Area / Arbeitsraum.....	54
15.7.1	XYZ Limits	54
15.7.2	Range	55
15.7.3	Working Space	56
15.8	Karte Programm.....	57
15.9	Karte System Settings / System-Einstellungen.....	60
15.10	Bereich Calibration / Kalibrierung.....	61
15.11	Feld Terminal	61
15.11.1	Bereich Status und Konfigurierung.....	62
15.11.1.1	IO	62
15.11.1.2	MODBUS	63
15.11.1.3	Clamp.....	63
15.11.1.4	Dedizierte Signale	64
15.11.1.5	Erkennung von Kollisionen.....	64
15.11.1.6	Förderband.....	Błąd! Nie zdefiniowano zakładki.
15.11.1.7	Ethernet.....	66
15.11.1.8	Firmware.....	66
15.12	Visualisierungsfenster.....	67
15.12.1	Bedienung des Visualisierungsfensters	67
15.12.2	Objektclassen	68
15.12.3	Generator einfacher Formen	69
15.12.4	Menü der Objektmodifikation	71
15.12.5	Menü Visualisierungseinstellungen	72
15.12.6	Working Space - Visualisierung	73
15.12.7	Visualisierung der Arbeitsreichweite des Roboters.....	73
15.13	Karte About / Informationen.....	74

ASTORINO Betriebsanleitung

15.14	Aktualisierung der Firmware	75
15.14.1	Gründlegende Informationen	75
15.14.2	Aktualisierungsverfahren	76
15.14.3	Datenwiederherstellung nach Aktualisierungsstörung	78
15.15	AS-Sprache	80
15.16	Programmieren	84
15.16.1	Erstellung eines neuen Programms	84
15.16.2	Eigenes Programm schreiben	85
15.16.3	Programm beim Roboter laden	85
15.16.4	Ingangsetzen der aktuell ausgewählten Zeile	86
15.16.5	Programm aktivieren	87
15.16.6	Programm anhalten	88
16	Beispielprogramme	89
16.1	Pick&Place – Palletierung-Beispiel	89
16.2	Beispielprogramm Eingang/Ausgang	91
16.3	Beispielprogramm für die serielle Datenübertragung	91
17	Daten des Werkzeugs	93
17.1	Dane Werkzeugia ze znanych wymiarów	93
17.2	Automatische Konfiguration des Werkzeugs (Koordinaten)	95
17.2.1	Übersicht der Funktionen der automatischen Werkzeugkonfiguration	95
17.2.2	Erforderliche Daten zur automatischen Konfiguration der Werkzeugkoordinaten ...	96
17.2.3	Einteachen von vier Grundpositionen	96
17.2.4	Einteachen von vier Grundpositionen	99
18	Automatische Konfiguration des WORK-Systems (KOORDINATEN)	104
18.1.1	Einteachen von drei Grundpunkten	104
19	Autokalibrierung der Kollisionserkennung	106
20	Eingänge/Ausgänge – 3,3V	107
20.1	I/O 3.3V- Adapter	109
20.1.1	I/O 3.3V- Adapter – Montage	109
21	Ausgänge und Eingänge am Roboterarm	110
22	I/O 24V-Modul	111
22.1	Anschluss an den Roboter	Błąd! Nie zdefiniowano zakładki.
22.2	Anschließen der Eingänge	Błąd! Nie zdefiniowano zakładki.
22.3	Anschließen der Ausgänge	Błąd! Nie zdefiniowano zakładki.
23	MODBUS TCP	114
23.1	Betriebsarten des Modbus-Netzes	114
23.2	Modbus-Objekttypen bei Astorino	115
23.3	Konfiguration des Ethernet-Ports	116
23.4	ASTRAADA HMI-Panel – Beispiel	117
23.5	Nutzung der Modbusregister als wirkliche Werte	122
24	Kalibrierung	123
25	Informationen über den Hersteller	124

ASTORINO Betriebsanleitung

Anlage A – Auf null setzen - Standardverfahren	125
Anlage B – PETG-Material	127
Anlage 3 – Anschluss Typ PNP	128
Anlage 4 – Teensy 4.1	129

1 Bezeichnungen in dieser Anleitung

In diesem Abschnitt finden Sie Definitionen der Termine, die in dieser Anleitung angewandt werden.

Der Autor dieses Handbuches ist bemüht, die allgemein geltende Terminologie bei Einhaltung der möglichst großen Logik anzuwenden. Es ist leider anzumerken, dass sich die Wahrnehmung der angewandten Terminologie je nach dem Gesichtspunkt unterscheiden kann, auch wenn dasselbe Thema behandelt wird. Es ist ebenfalls festzustellen, dass sich im Laufe der Entwicklung von Robotern, Computern und Software auch die Terminologie auf verschiedenen Wegen entwickelt hat. In einer modernen Anleitung finden wir also keine Terminologie, die mit Meinungen aller Nutzer und Experten immer hundertprozentig übereinstimmen wird.

1.1 Beschreibung des ASTORINO-Roboters

ASTORINO ist ein sechssachsiger Bildungsroboter mit Schrittmotoren, die in einer geschlossenen Steuerungsschleife arbeiten. Der Roboter ist eigens für Bildungsstellen und -einrichtungen, wie z. B. Schulen und Universitäten, entworfen worden.

Die Roboterkonstruktion stützt sich auf den 3D-Druck mit spezieller Kohlefaser. Unter Anwendung von gelieferten STL-Dateien kann man beschädigte Teile nachdrucken.

Die Programmierung und die Steuerung erfolgen mittels "Astorino"-Software, die man auf dem gelieferten USB-Speicher finden kann und die neueste Version kann man vom FTP-Server Kawasaki Robotics herunterladen:

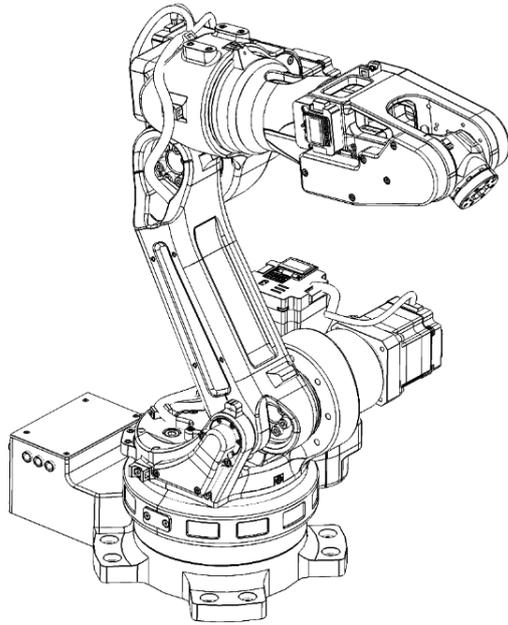
<https://ftp.kawasakiRoboter.de/Software/Astorino/>

Ähnlich wie es bei Industrierobotern der Fall ist, ist der Kawasaki Robotics Astorino-Roboter in der AS-Sprache programmiert und ermöglicht dem Nutzer echte industrielle Anwendungen für die Roboter der Kawasaki Robotics zu programmieren.

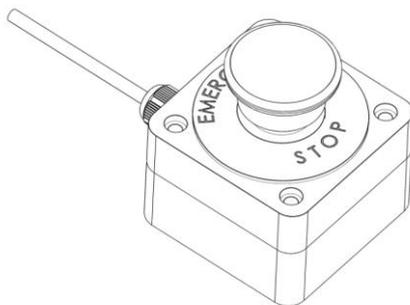
2 Technische Spezifikation

Leistungsmerkmale		ASTORINO
Typ		sechssachsiger Roboter
Maximale Traglast		1 kg
Anzahl Achsen		6
Maximale Reichweite		578 mm
Wiederholgenauigkeit		±0,1 mm
Bewegungsbereich	Achse 1 (JT1)	±158°
	Achse 2 (JT2)	-90°÷127°
	Achse 3 (JT3)	0°÷168°
	Achse 4 (JT4)	±240°
	Achse 5 (JT5)	±120°
	Achse 6 (JT6)	±360°
Geschwindigkeit einzelner Achse	Achse 1 (JT1)	38°/s
	Achse 2 (JT2)	26°/s
	Achse 3 (JT3)	26°/s
	Achse 4 (JT4)	67,5°/s
	Achse 5 (JT5)	67,5°/s
	Achse 6 (JT6)	128.5°/s
Zulässiger Moment	Achse 4 (JT4)	6,2 Nm
	Achse 5 (JT5)	1,45 Nm
	Achse 6 (JT6)	1,1 Nm
Betriebsumgebung	Temperatur	0–40°C
	Feuchtigkeit	35–80%
Kontroller		Teensy 4.1
Eingänge/Ausgänge		8/8 (PNP 8 mA, NPN 15 mA)
		2/2 (24V PNP na JT3)
Maximale Stromaufnahme		144 W
Speisung		100–240 V, 50–60 Hz
Gewicht		12 kg
Montage-Position		auf dem Fußboden
Material		PETG
Farbe		schwarz
Kommunikation		MODBUS TCP, TCP/IP, UDP, Serial
Kollisionserkennung		Beschleunigungssensor
Sicherheit beim Stromausfall		Bremsen bei JT2 und JT3
Optionen	Modul I/O 24V	8 × Eingänge/Ausgänge
	7. Achse	Linearführung
	Visionssystem	OpenMV
	Bandverfolgung	2 Encoder

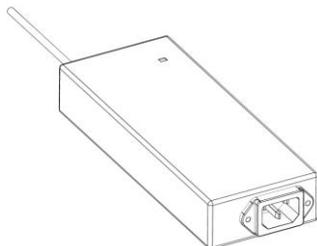
3 Lieferumfang



ASTORINO-Roboter

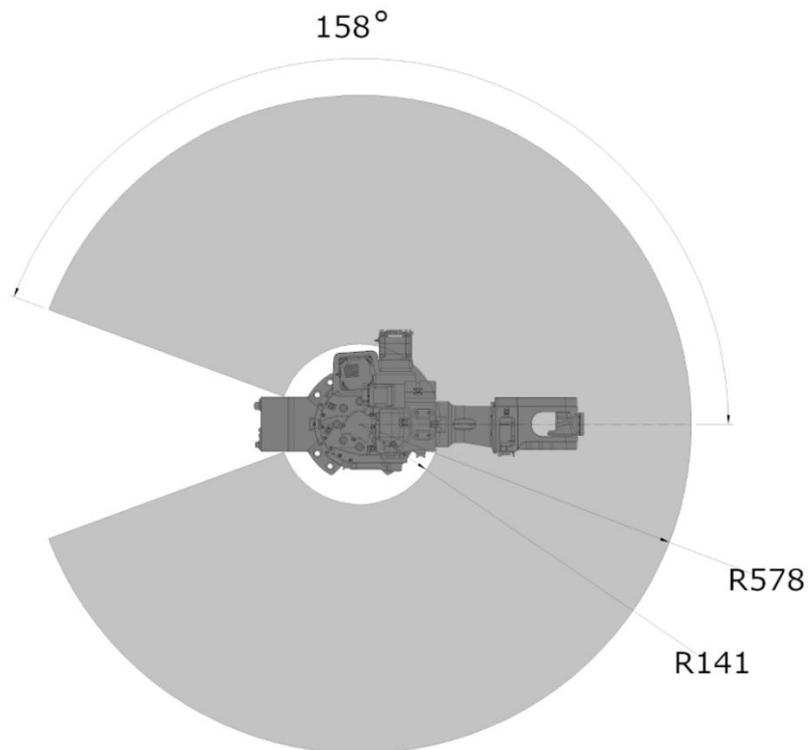
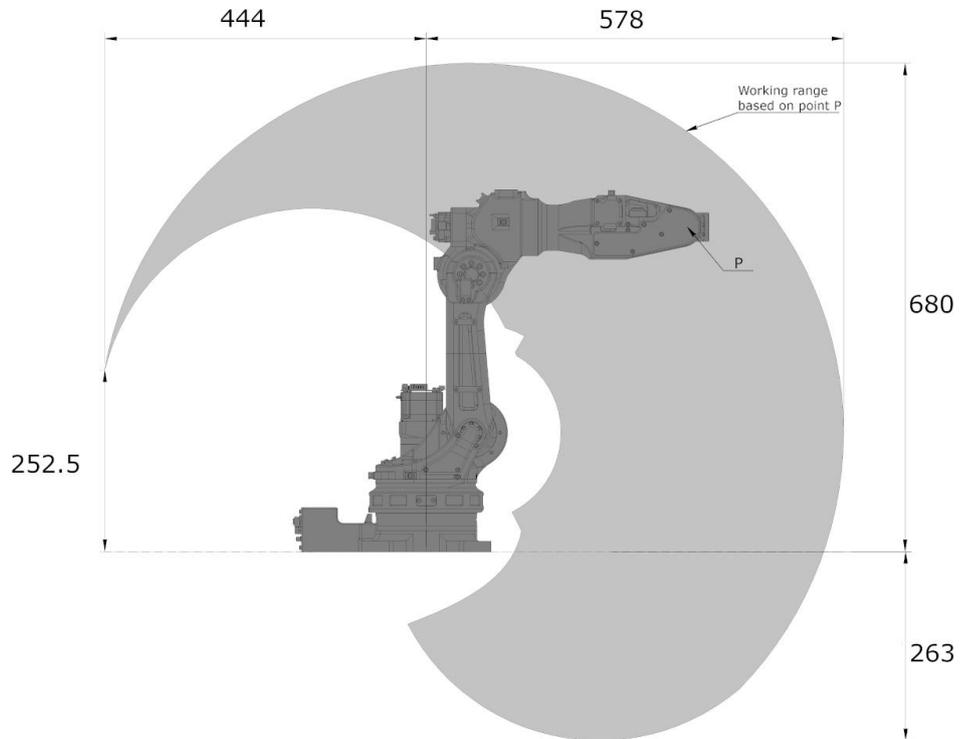


Sicherheitstaste im Gehäuse

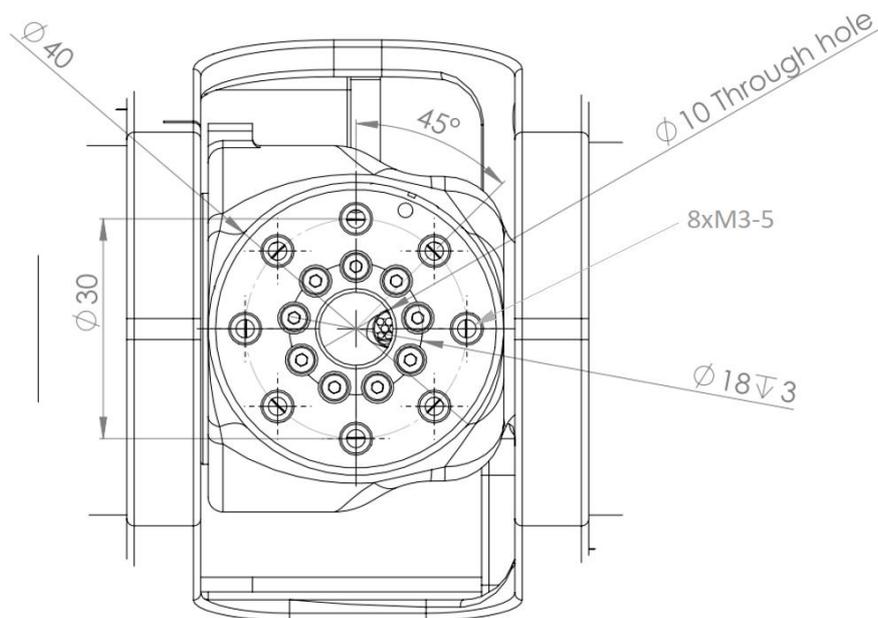
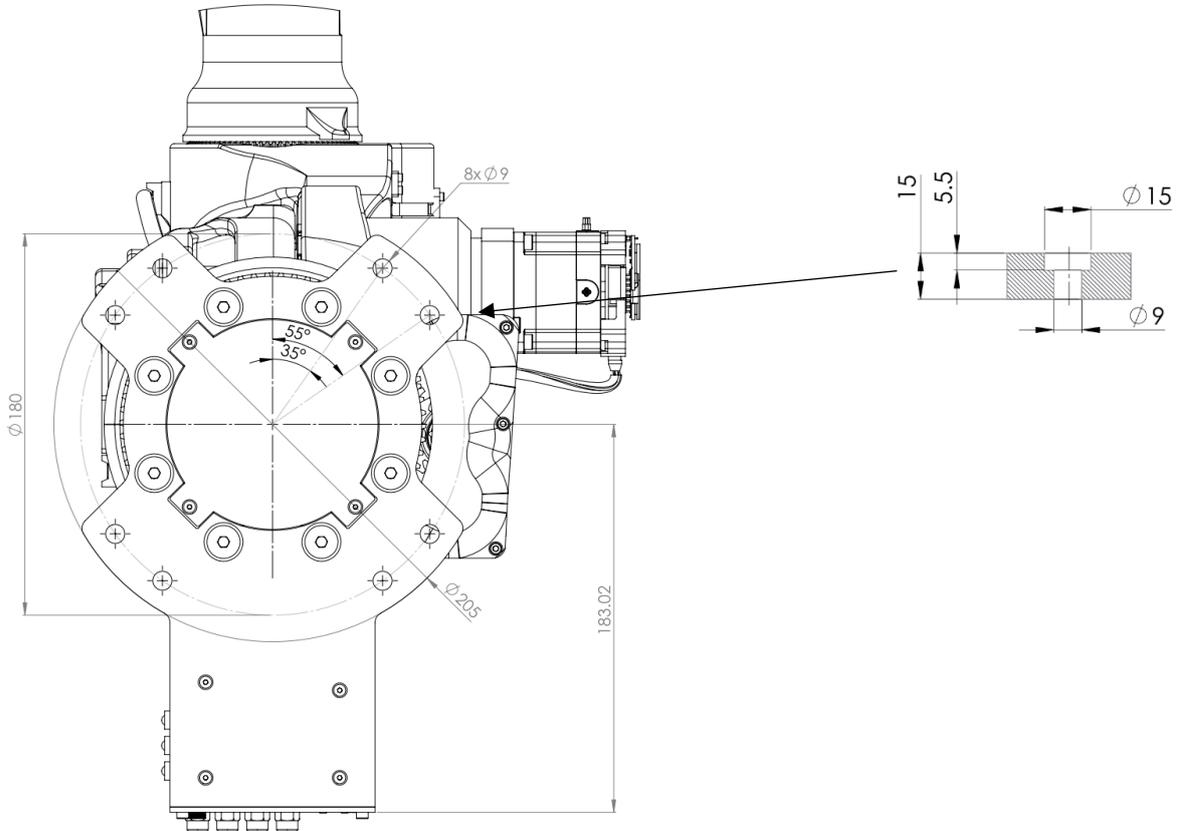


Netzteil 24V/DC, USB-Kabel und
USB-Speicherstick

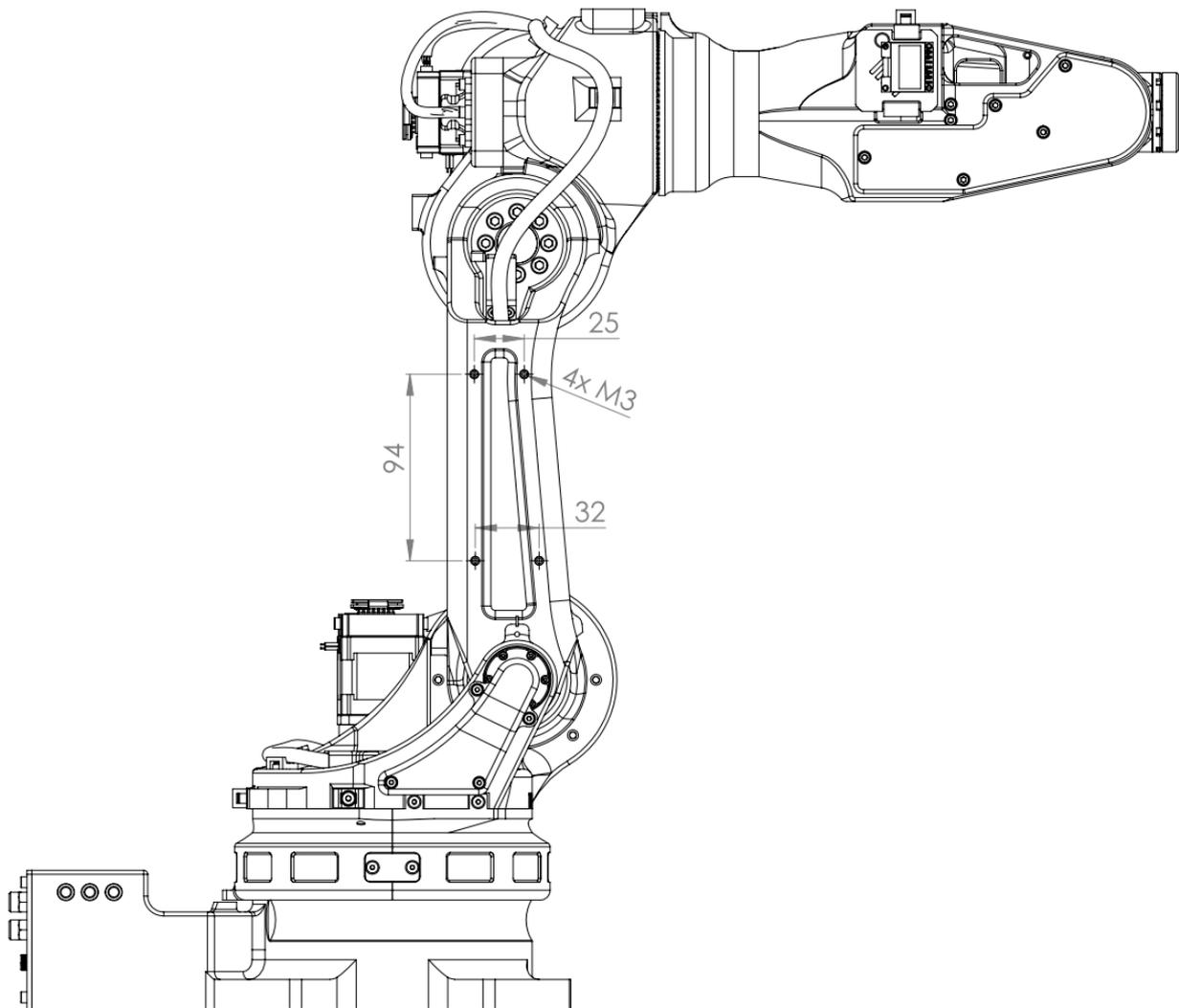
4 Bewegungsbereich



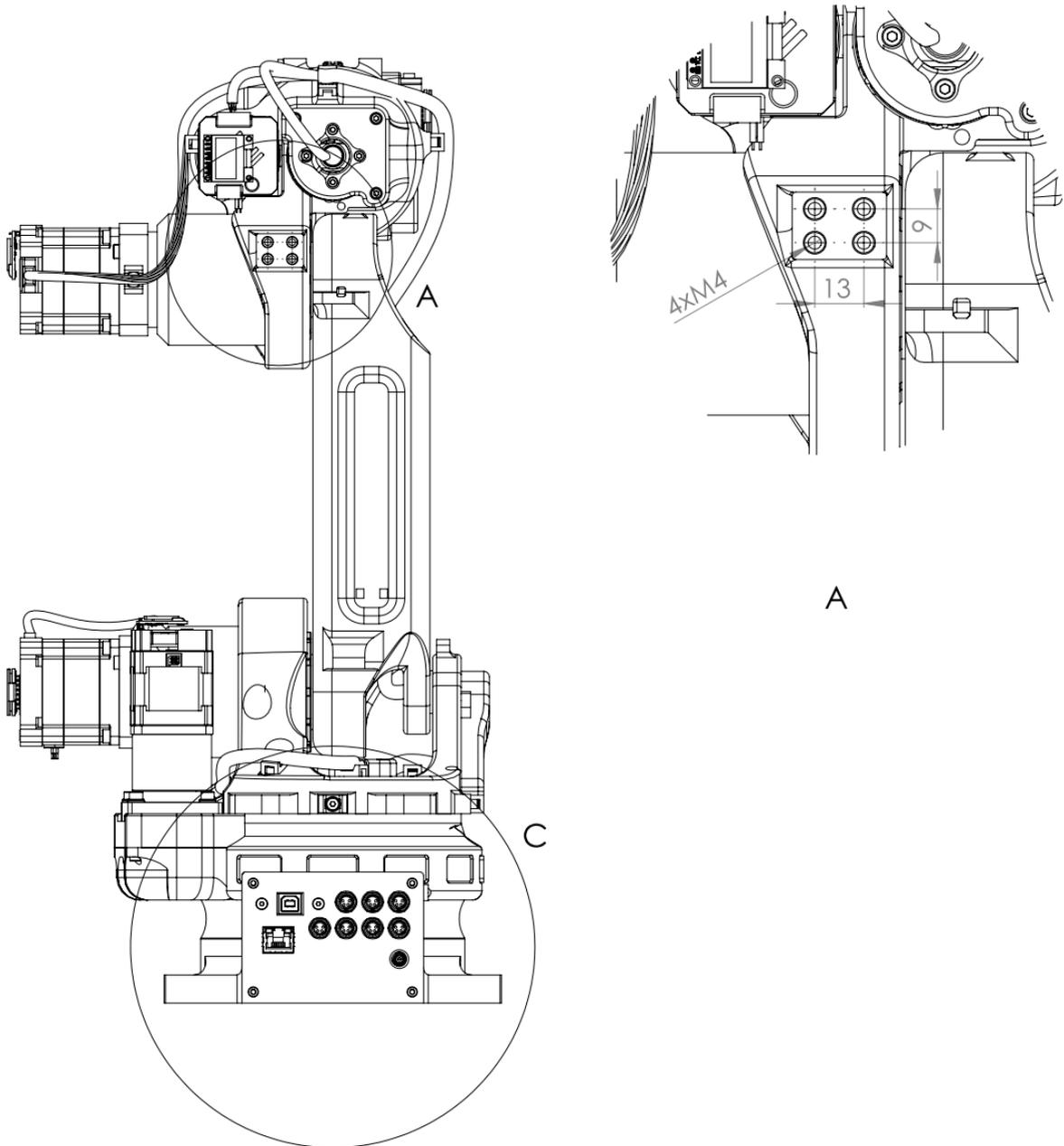
5 Montageabmessungen



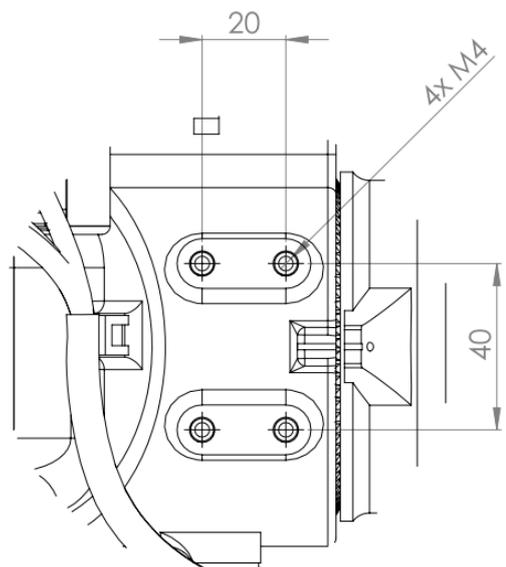
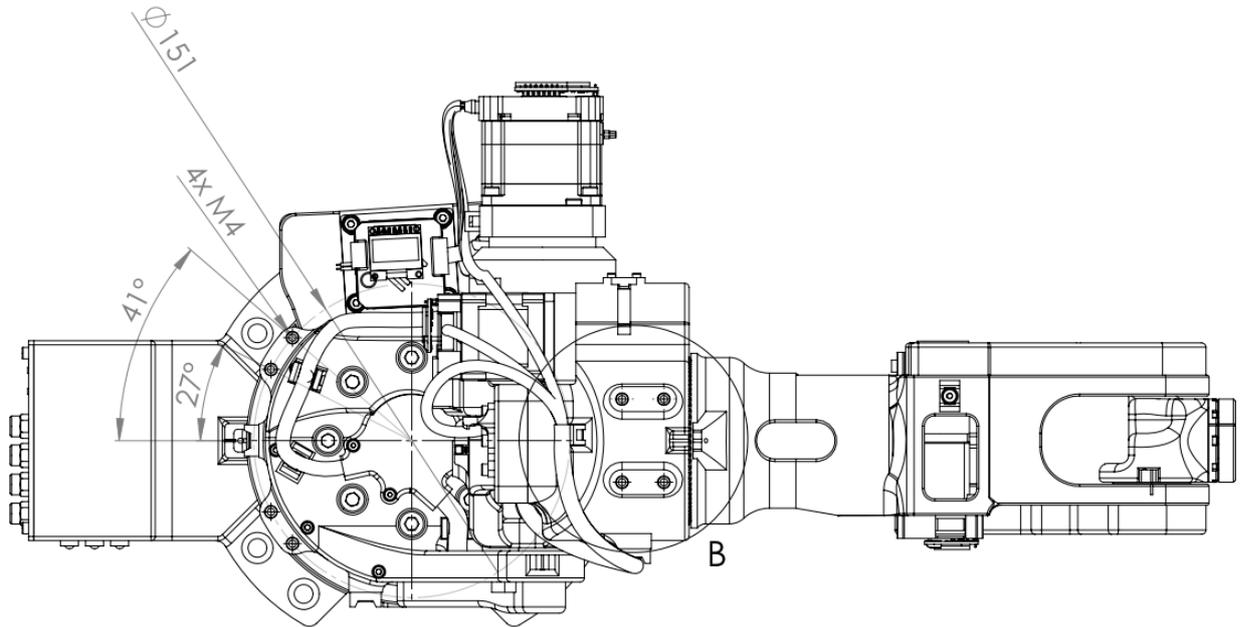
6 Zubehör-Montagepunkte



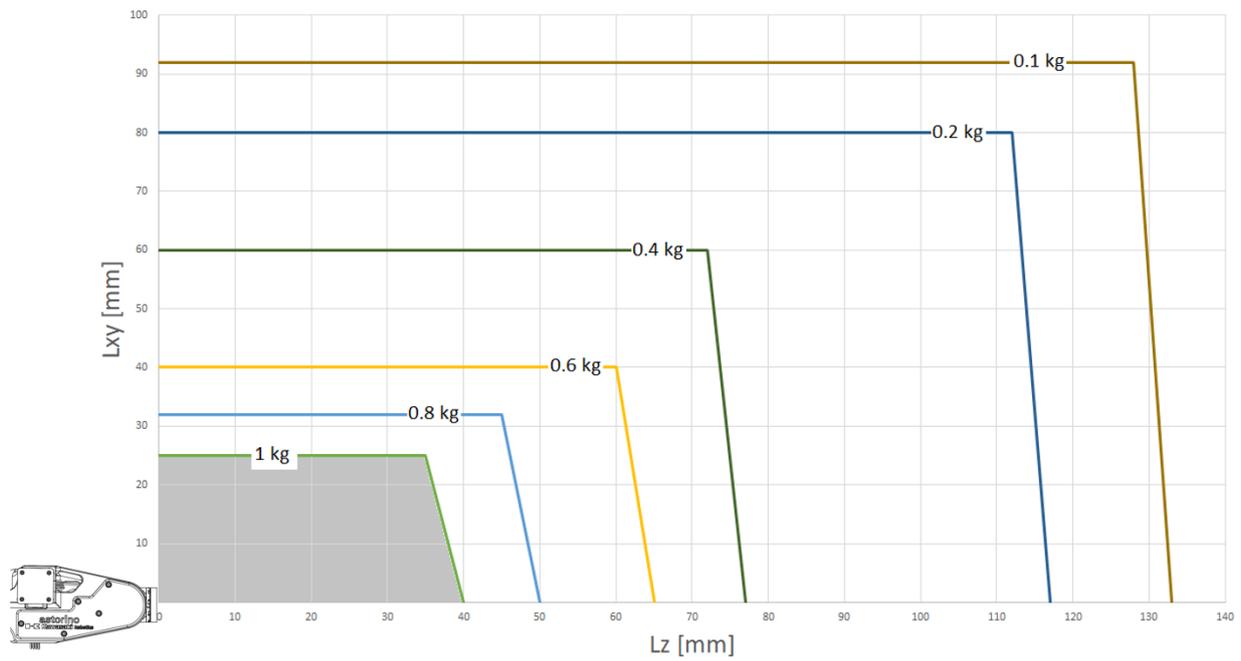
ASTORINO Betriebsanleitung



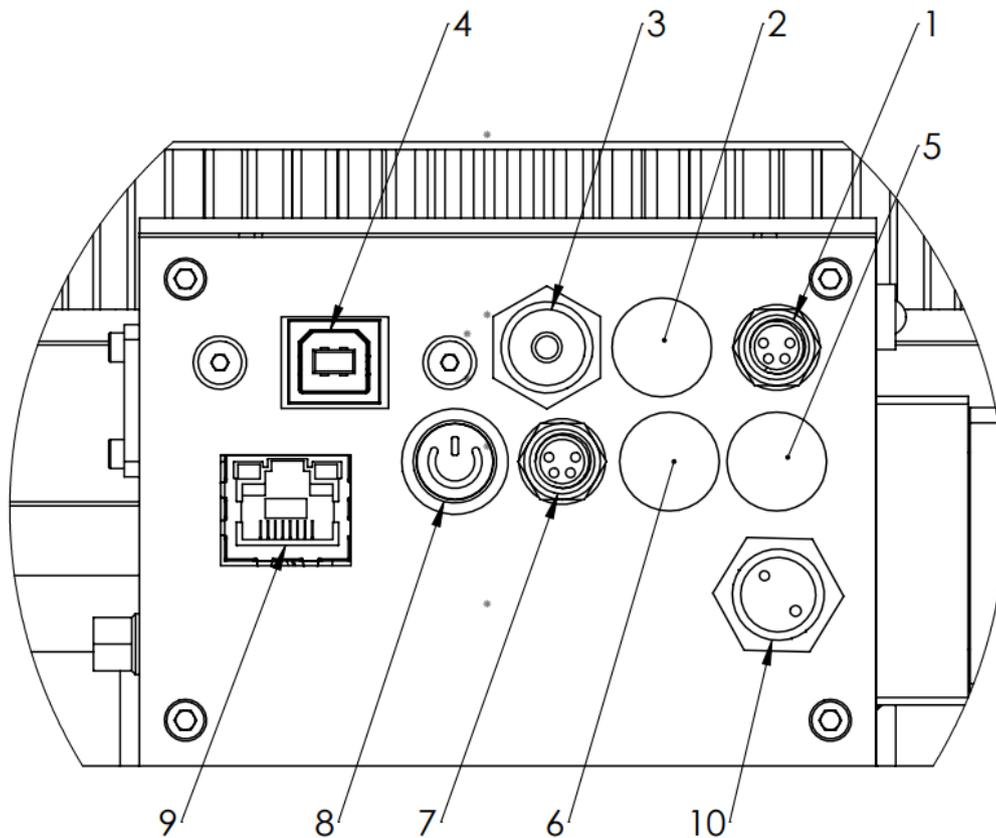
ASTORINO Betriebsanleitung



7 Lastdiagramm



8 Elektrische Anschlüsse



1.	M8-Steckdose 4-pin – Außen-Not-Halt-Taster (E-Stopp)
2.	Außen-Not-Halt-Taster SAFETY-FENCE (OPTION)
3.	Druckeinlass Ø4,0 mm
4.	USB-B-Port
5.	OPTION 2 (Enkoder 2 – Bandförderer 2/JT7)
6.	OPTION 1 (Enkoder 1 – Bandförderer 1)
7.	Vision-System /serielle Datenübertragung (Serial) (Pins: 1-GND, 2 – 5V [Vision], 3-TX, 4- RX)
8.	ON/OFF-Schalter
9.	Ethernet-Port (RJ45)
10.	Einspeisung

9 Sicherheitshinweise

[VORSICHT]

Bei Bedienung des Roboterarms oder bei Inbetriebsetzung der Roboterzelle ist immer die Sicherheit der Nutzer und anderer Personen zu beachten!

- In seiner Grundversion hat der Roboter keine Elemente, die mit Sicherheit der robotisierter Stelle verbunden sind. Abhängig von der Anwendung kann es erforderlich sein, diese hinzuzufügen. Die Grundversion des Roboters ist mit einem Not-Halt-Taster ausgestattet.
- CE-Kennzeichnung: der Roboterarm muss beim Betrieb in Produktionsanwendungen der Risikobeurteilung unterliegen und muss mit geltenden Sicherheitsvorschriften übereinstimmen, damit die persönliche Sicherheit sichergestellt ist. Abhängig vom Ergebnis der Beurteilung sind weitere Sicherheitsbestandteile zu integrieren. Es sind in der Regel Sicherheitsrelais und Türschalter. Verantwortlich dafür ist der Ingenieur für die Inbetriebnahme des Systems. Die Bildungsapps erfordern keine weiteren Sicherheitselemente.
- Die Steuereinheit des Roboters enthält ein 24 V-Netzteil, welches selbst eine Netzspannung (100/240 V) erfordert. Das Etikett auf dem Netzteil ist zu prüfen. Nur qualifiziertes Personal darf das Netzteil ans Netz anschließen und es in Betrieb setzen.
- Die Arbeiten an der Roboterelektronik sollen ausschließlich durch qualifiziertes Personal ausgeführt werden. Aktuelle Richtlinien zu elektrostatischen Entladungen (ESD) sind zu überprüfen.
- Der Roboter ist immer bei der Arbeit des Menschen am Sockel des Roboters (Steuereinheit) oder bei jedweder Elektronik, die an die Steuereinheit des Roboters angeschlossen ist, von der Speisung (100/240 V) zu trennen.
- Hot Plugging ist NICHT gestattet! Das kann eine dauerhafte Beschädigung der Motormodule zur Folge haben. Man soll Module und Steck-/Trennan-schlüsse (z. B. Not-Halt-Taster, digitale Eingangs-/Ausgangsmodule, Motoranschlüsse) bei eingeschalteter Speisung weder installieren noch entfernen.
- Der Roboterarm muss auf stabiler Oberfläche aufgestellt und angeschraubt oder anderweitig gesichert werden.
- Der Roboter ist ausschließlich in trockener und sauberer Umgebung zu verwenden und aufzubewahren
- Empfehlung: Das System ist ausschließlich in Raumtemperatur (15° bis 32°C) zu verwenden.

10 Auspacken und Inbetriebnahme

Nachdem der Roboter aus der Verpackung herausgenommen ist, ist er auf einer soliden Oberfläche aufzustellen und abzusichern.

10.1 Anschließen von Zubehör

- Die Sicherheitsanweisungen sind zu befolgen.
- Der Roboter ist auf einem entsprechenden Sockel, Tisch oder auf einer Metallplatte aufzustellen. Der Roboter ohne Greifer oder anderes Zubehör führt grundlegende Vorgänge wie auf null setzen und Einteachen von Bewegungen in direkter Nähe des Robotersockels aus. Aus Sicherheitsgründen empfehlen wir den Robotersockel vorm Einschalten der Speisung anzuschrauben.
- Der Netzteilstecker mit zwei Kontaktstücken und der M8-Stecker mit vier Kontaktstücken des Außen-Not-Halt-Tasters sind an die Anschlüsse auf dem Robotersockel anzuschließen.

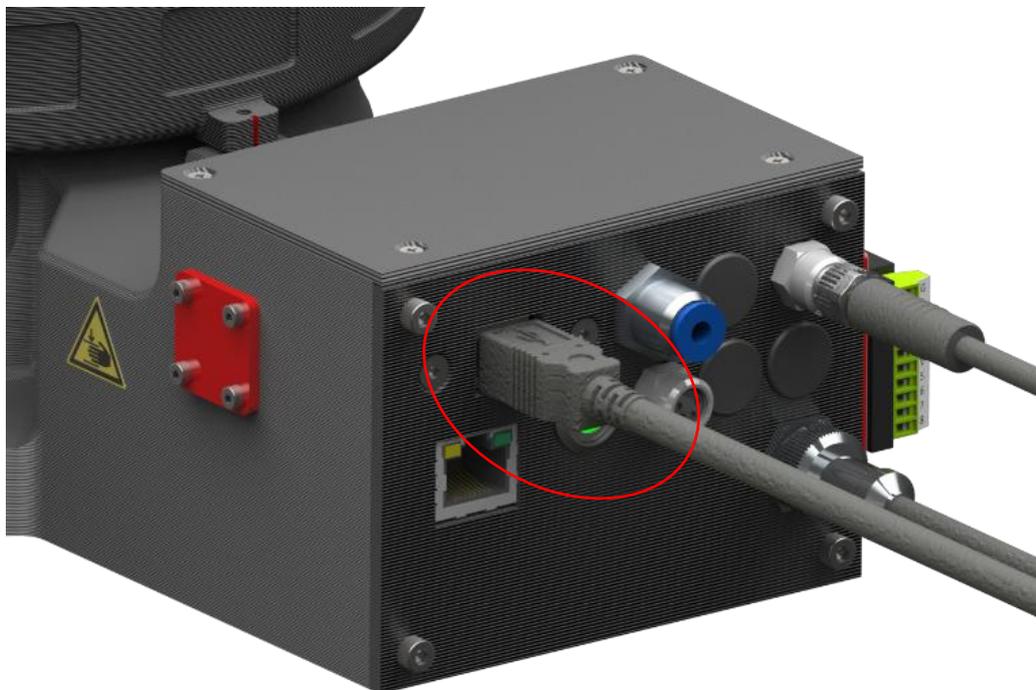


ASTORINO Betriebsanleitung

- Einspeisung durchs Drücken der hinterleuchteten Taste einschalten.



- USB ist an den USB-B-Port des Robotersockels anzuschließen und demnächst ist er an den Computer anzuschließen.



ASTORINO Betriebsanleitung

10.2 Systemanforderungen

Bevor die Software astorinoIDE installiert wird, vergewissern Sie sich, dass der Computer folgende Hardware- und Software-Anforderungen erfüllt.

Teil	Anforderungen
CPU	2.0 Ghz oder schneller
Speicherkapazität	min. 4 GB
Festplatte	100 MB freier Speicherplatz
Grafikkarte	beliebige
Bildschirmeinstellungen	Mindestauflösung 1280 x 720, empfohlene Anzeigeskalierung 100 %
Maus	mit drei Tasten

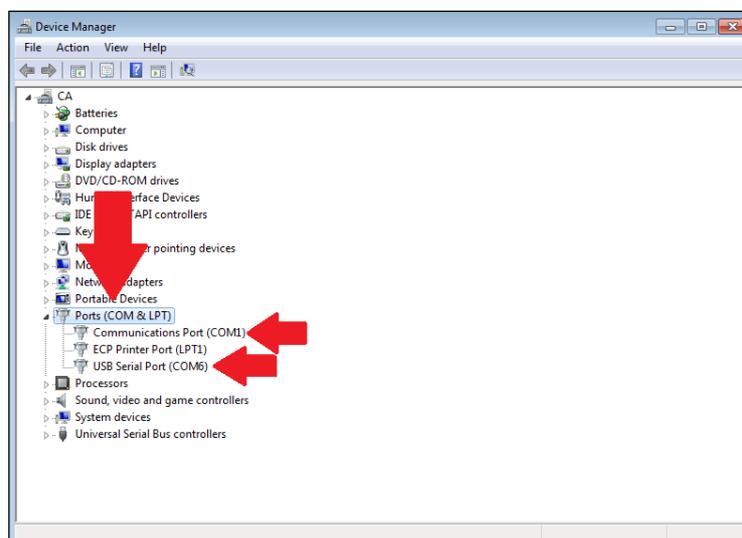
System	Version
Windows	7, 8, 8.1, 10, 11

10.3 Installation der Steuereinheit

Das Betriebssystem installiert erforderliche Steuereinheiten ab der Version Windows 8 automatisch. Nach erfolgreicher Installation soll der Roboter im Geräte-Manager in der Position <Ports> erscheinen.

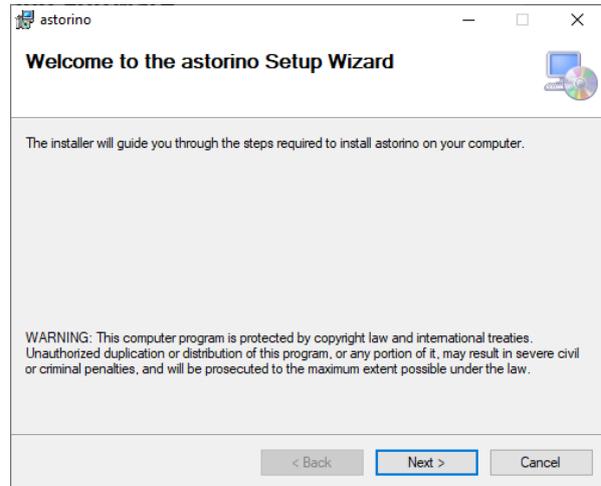
Wenn Sie das System Windows 7 benutzen, installieren Sie die Steuereinheiten (vom FTP-Server Kawasaki oder vom USB-Speicherstick heruntergeladen), bevor der Roboter an den Computer angeschlossen wird.

Aufrufen des Geräte-Managers über <Windows + R> devmgmt.msclub ⇒ Ikone im Auswahlménü über <Windows + X> klicken

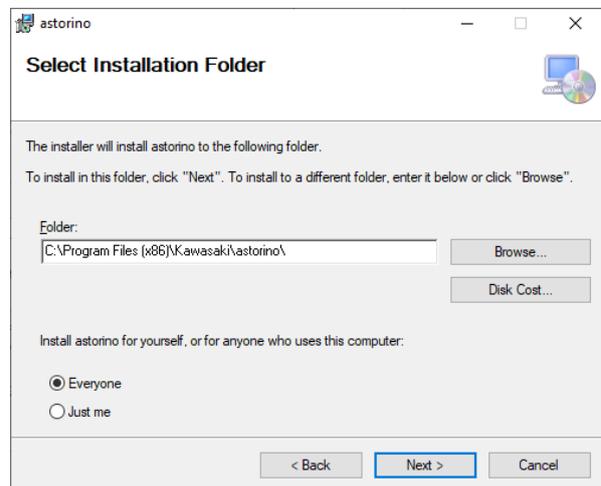


10.4 Installation der Astorino-Software

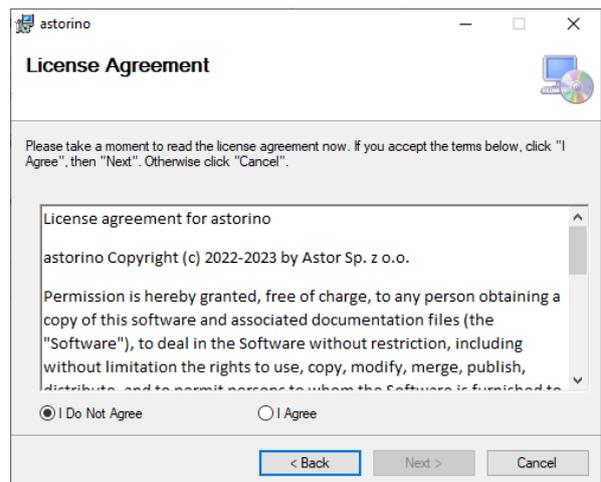
Astorino_x.x.x.exe einschalten



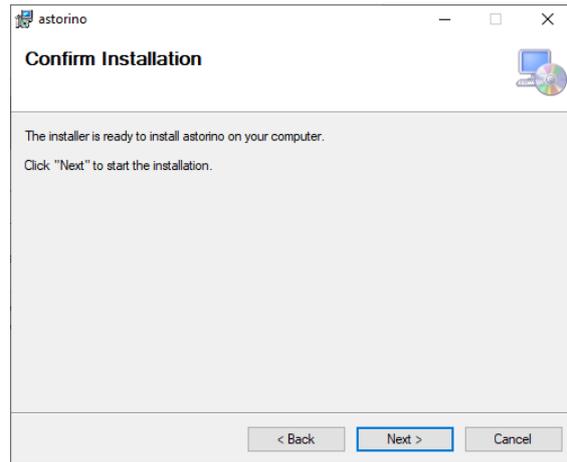
Bestätigen oder das Installationsverzeichnis anpassen



Lizenz akzeptieren

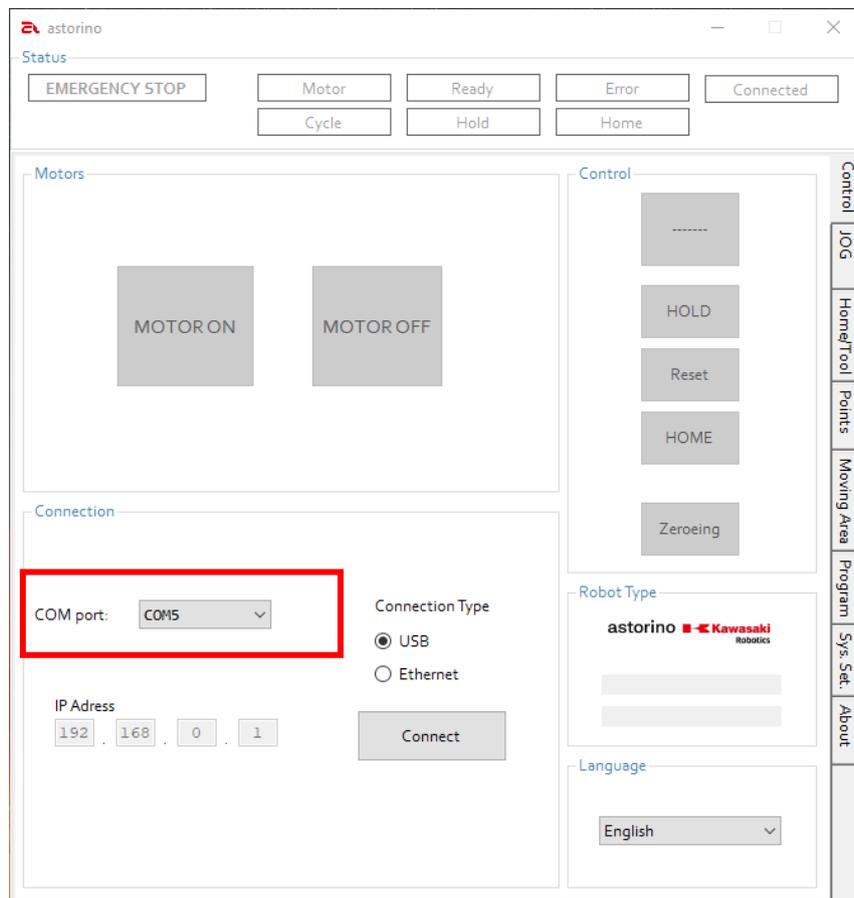


Installation beginnen



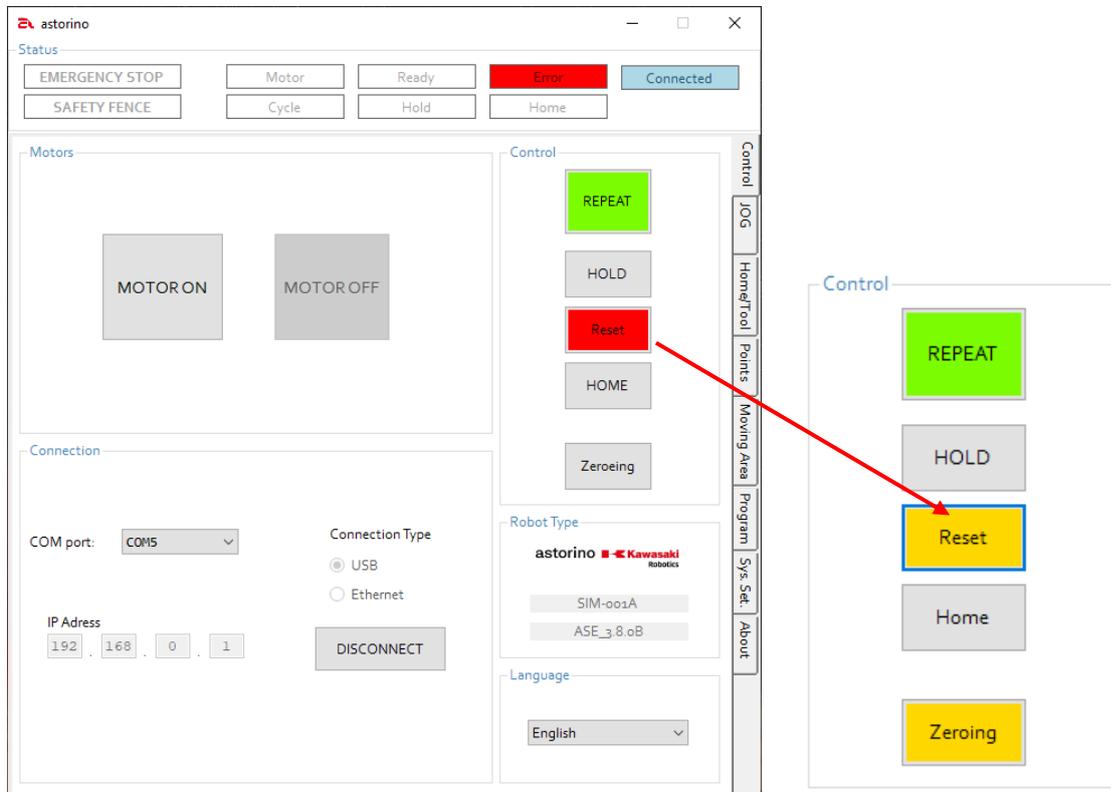
10.5 Vorbereitung von Astorino auf den Betrieb

- Astorino-Software eröffnen.
- Der COM-Port, an welchen der Roboter angeschlossen ist, soll auf der Dropdown-Liste im Bereich Verbindung im Menü [Steuerung] erscheinen.



ASTORINO Betriebsanleitung

- **[Reset]** klicken, wenn die Taste rot hinterleuchtet ist (Not-Halt-Taster prüfen)

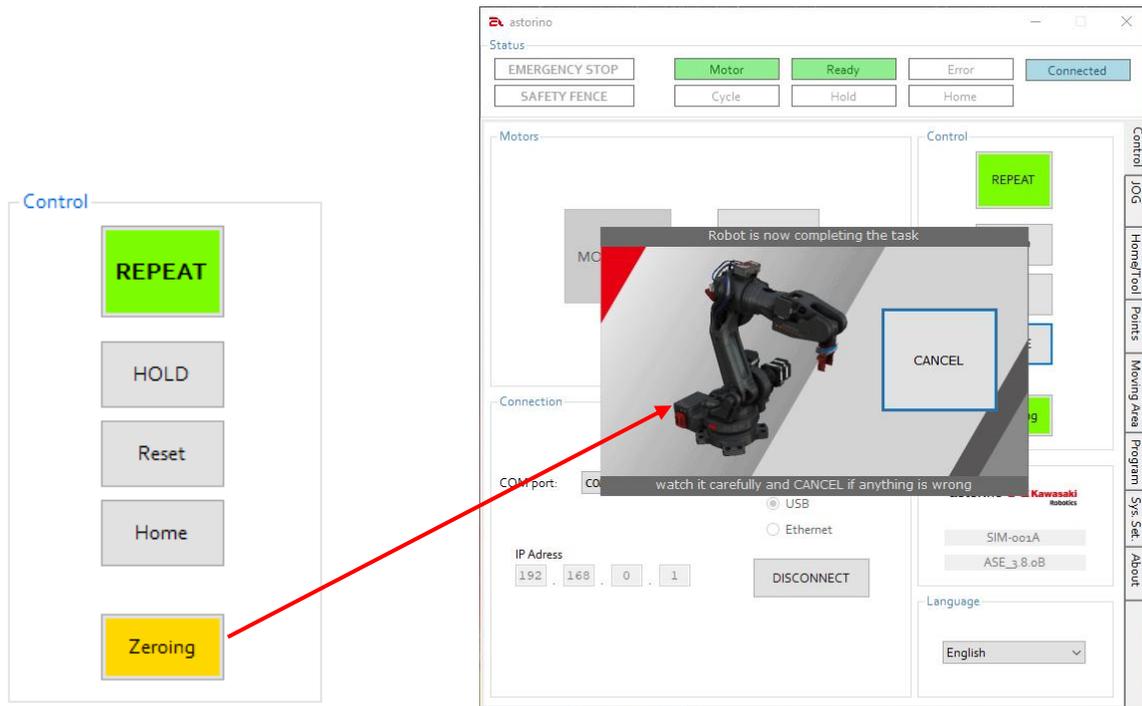


- Nun kann man durchs Klicken der grauen Taste **[MOTOR ON]** Motoren starten
- gelbes Feld **[Zeroing]** klicken, um den Vorgang auf null zu setzen, durchzuführen.

Man muss immer dann auf null setzen, wenn der Roboter von der Einspeisung getrennt wird oder wenn die Motoren abgeschaltet sind.

- Vergewissern Sie sich, dass der Roboter an keinen Gegenstand stößt!

ASTORINO Betriebsanleitung



- Nachdem es auf null gesetzt wurde, steht der Roboter in der Position 0 Grad an jeder Achse. Nun ist er bereit, programmiert zu werden.

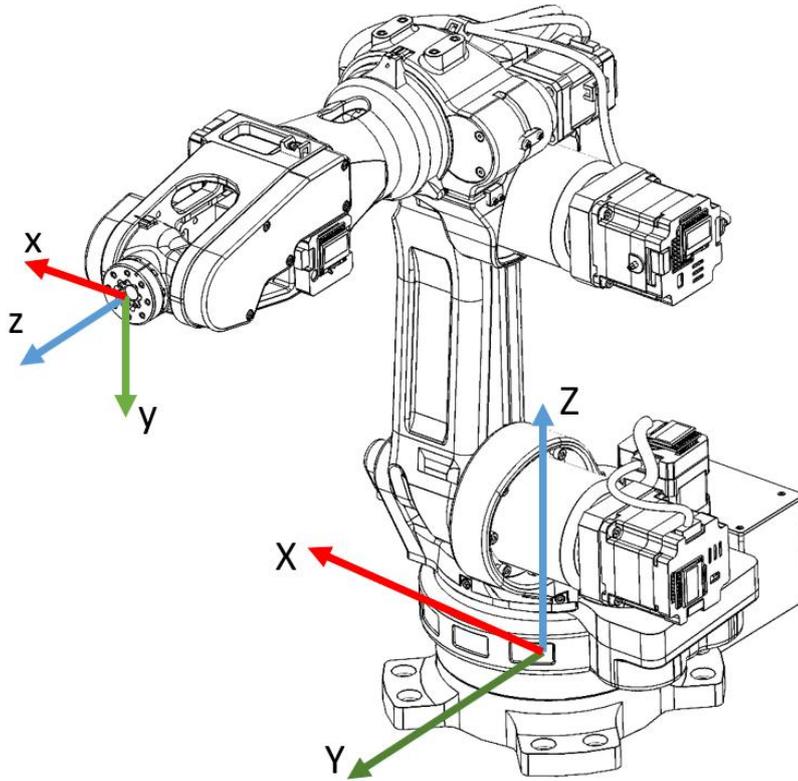


VORSICHT!

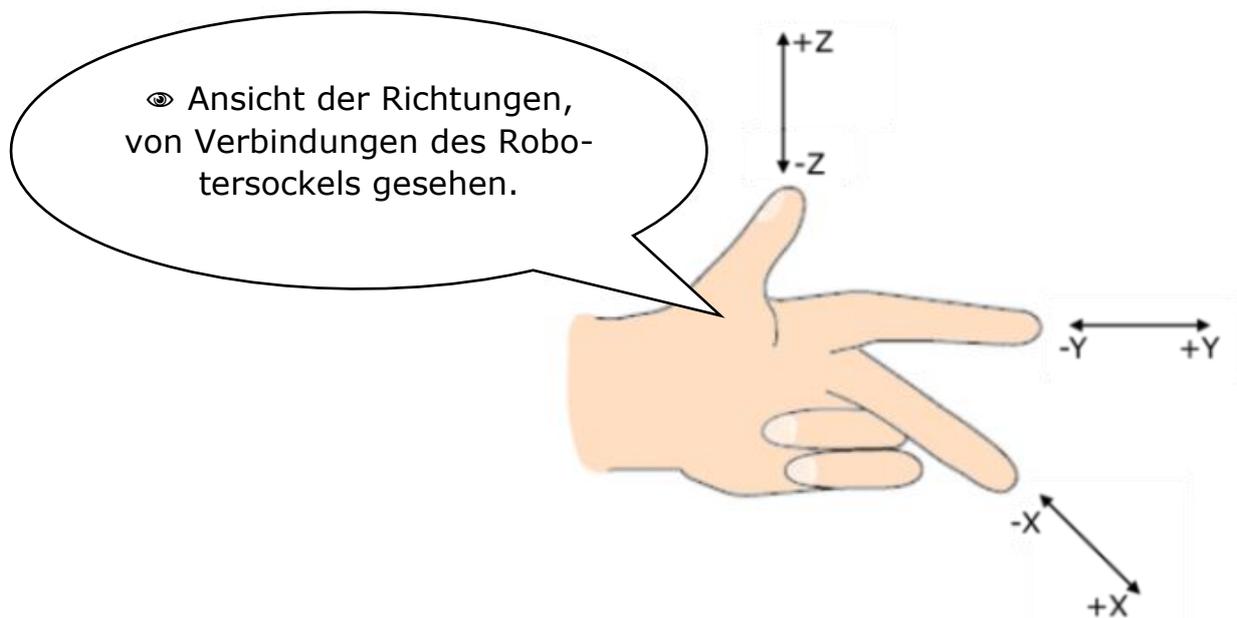
Die senkrechte Stellung ist eine Standardstellung des Arms, nachdem es auf null gesetzt wurde. Wenn die Einstellungen des Verfahrens auf null setzen geändert wurden, dann kann die Endstellung auch anders sein!

11 Koordinatensysteme

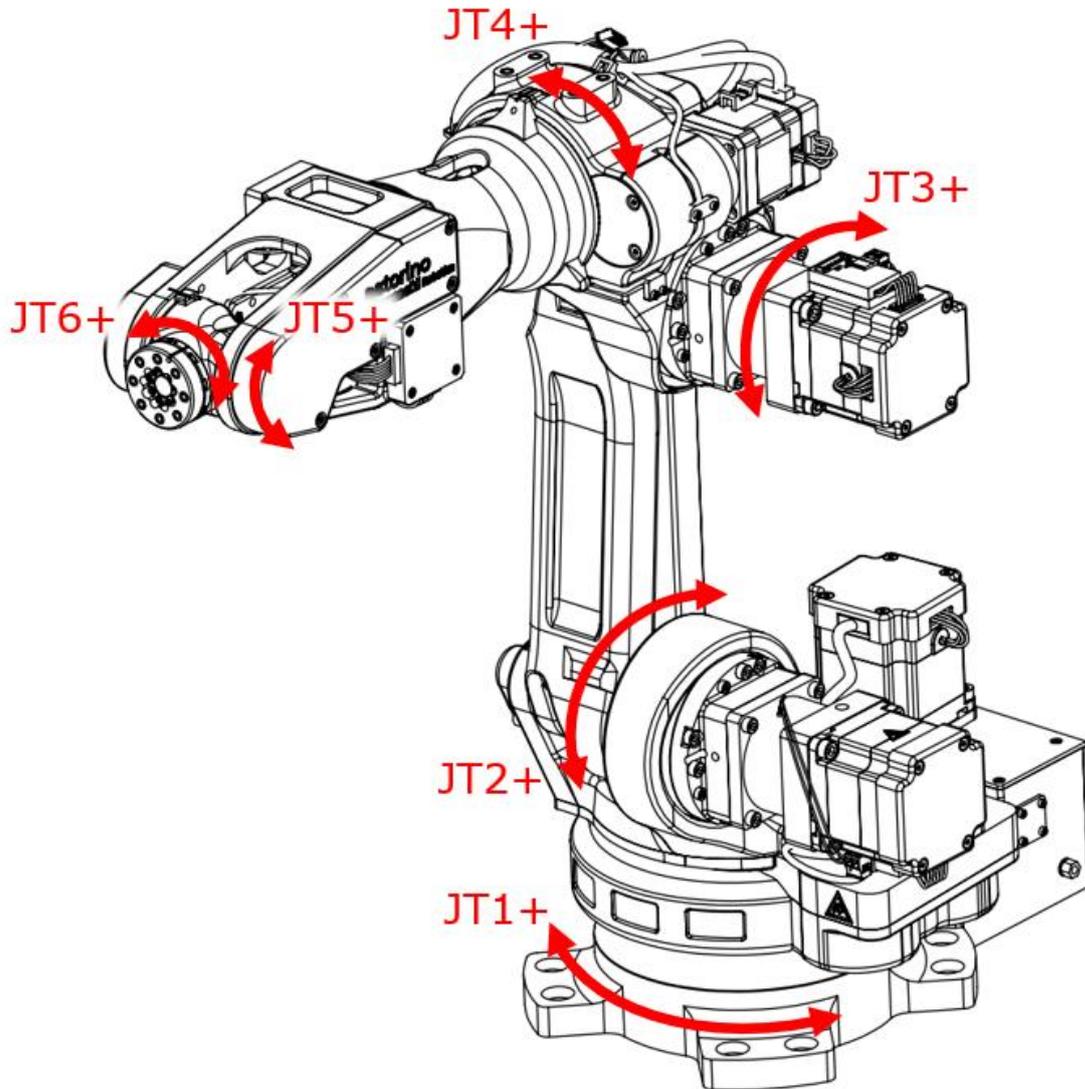
11.1 BASE-Koordinatensystem



Die Linke-Hand-Regel hilft die Achsenrichtungen merken:



11.2 JOINT-Koordinatensystem



JT = Achse

Die jeweiligen Verbindungen werden aufsteigend nummeriert, vom Robotersockel beginnend. JT bezeichnet die Achse.

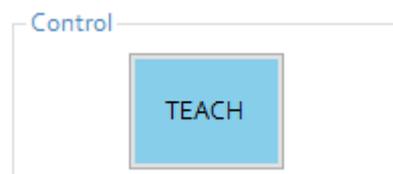
12 Betriebsarten des Roboters

12.1 Lehrmodus (Teach)

Dieses Modus ermöglicht die Handvorgänge des Roboters, wie Bewegungen ausführen, zu teachen. In diesem Modus ist die Höchstgeschwindigkeit auf 60 mm/s und 12 Grad/s für jede Achse beschränkt.

Das Teachen wird definiert als Programmieren des Roboters, erforderliche Aufträge auszuführen und jeweilige Positionen zu registrieren.

Der Roboter ist im Lehrmodus, wenn der Hintergrund des Schalters [REPEAT/TEACH] blau ist.



Im Lehrmodus, wenn der Eingang Safety Fence in hohem Zustand ist, sind die Robotervorgänge nicht beschränkt.

12.2 Wiederholmodus (Repeat)

Wiedergabe des vom Nutzer geschriebenen Programminhaltes. In diesem Modus sind die Geschwindigkeiten nicht eingeschränkt und der Roboter kann sich mit der Höchstgeschwindigkeit 250 mm/s bewegen.

Der Roboter ist im Wiederholmodus, wenn der Schalterhintergrund [REPEAT/TEACH] grün ist.

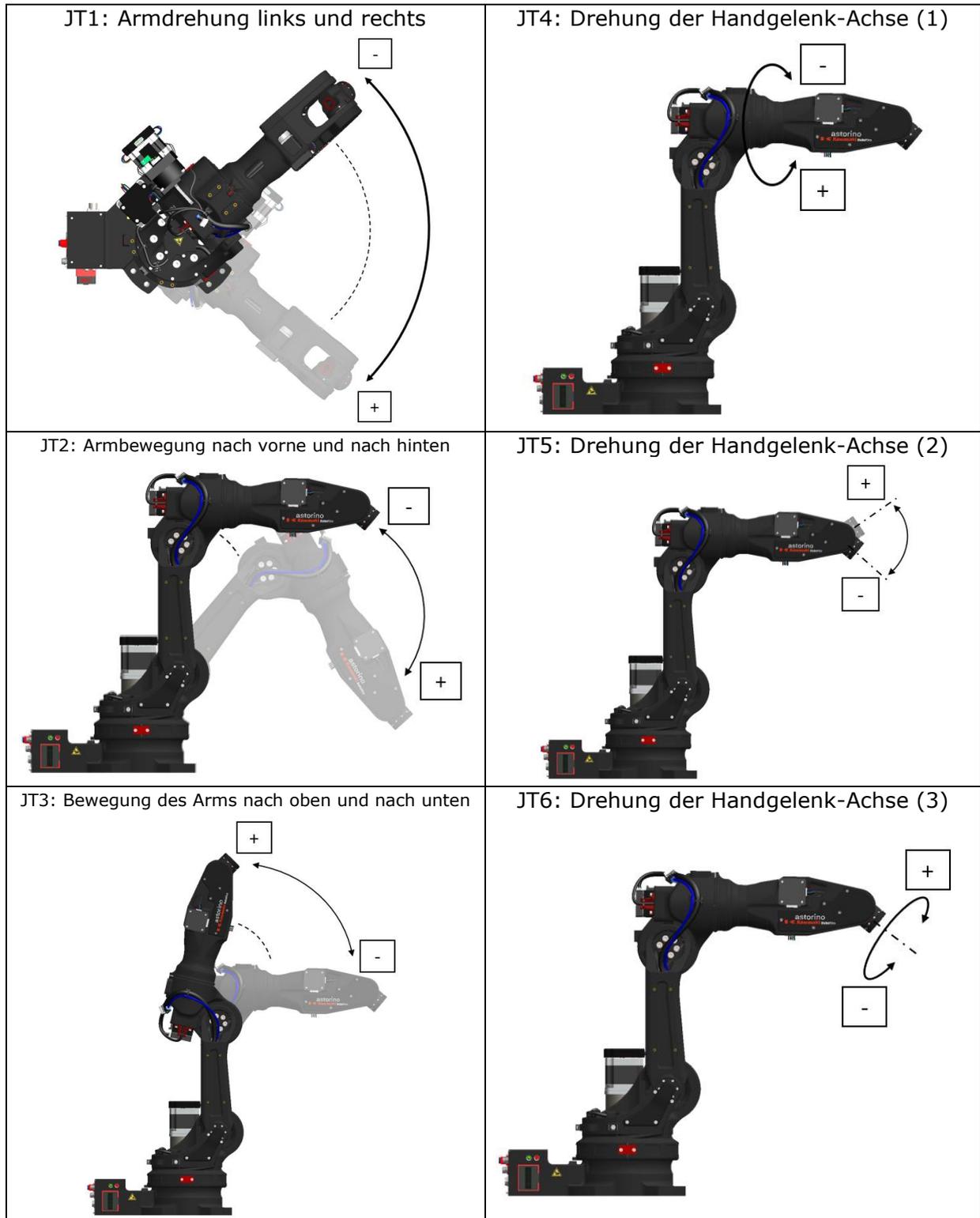


Im Wiederholmodus, wenn der Eingang Safety Fence in hohem Zustand ist, sind die Robotervorgänge beschränkt.

13 Manuelle Bedienung des Roboters

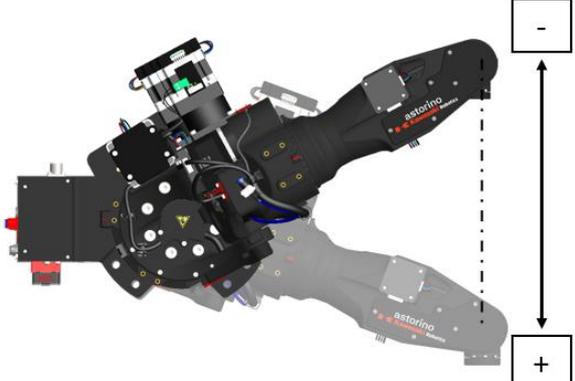
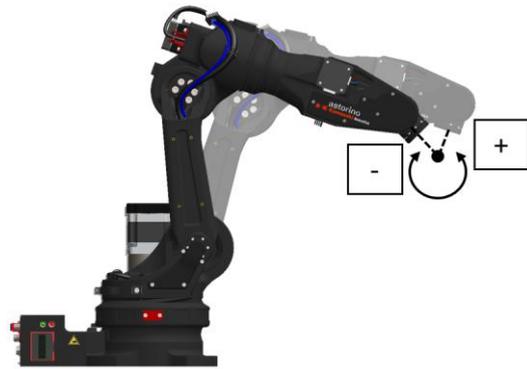
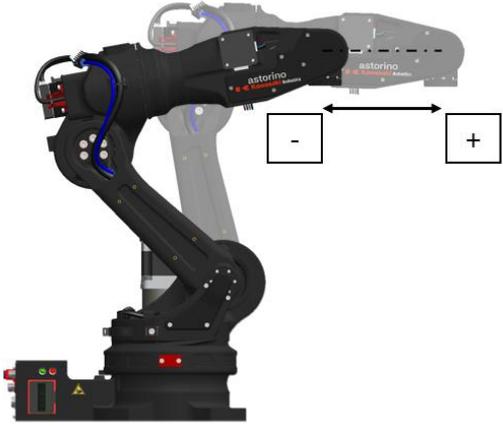
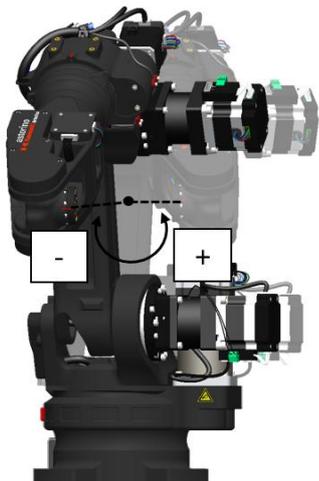
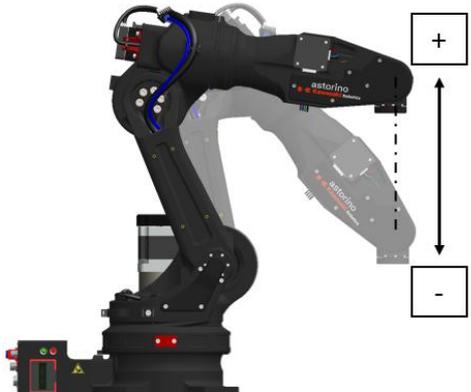
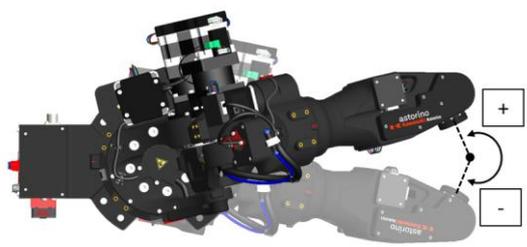
Im Lehrmodus, das auf dem aktuell gewählten Bewegungsmodus (BASE, JOINT, TOOL) gestützt ist, ist manuelles Bewegen des Roboterarms möglich.

13.1 JOINT (Punkt-zu-Punkt-Bewegung)



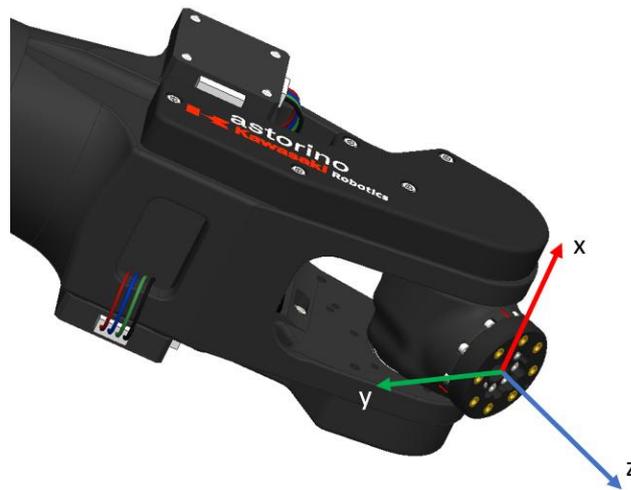
ASTORINO Betriebsanleitung

13.2 BASE (Bewegung in Bezug aufs Grundkoordinatensystem)

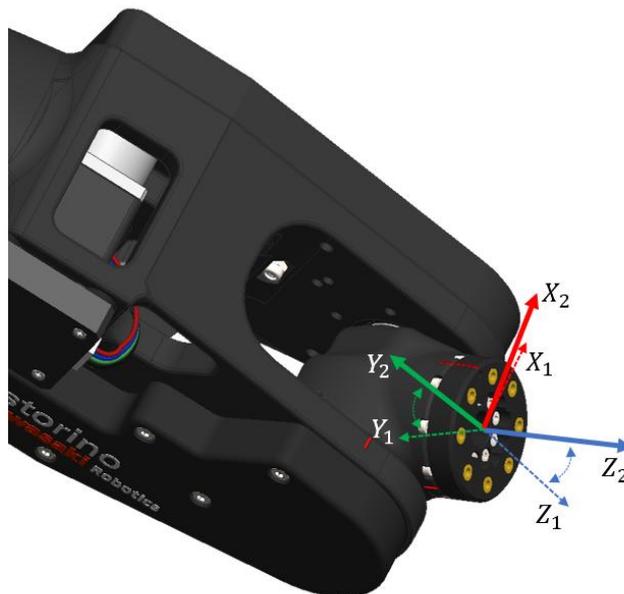
<p>X: parallele Bewegung zur Basisordinate X (die Orientierung des Handgelenks bleibt konstant)</p> 	<p>RX: Drehung um die Ordinate X (TCP wird nicht bewegt)</p> 
<p>Y: parallele Bewegung zur Basisordinate Y (die Orientierung des Handgelenks bleibt konstant)</p> 	<p>RY: Drehung um die Ordinate Y (TCP wird nicht bewegt)</p> 
<p>Z: parallele Bewegung zur Basisordinate Z (die Orientierung des Handgelenks bleibt konstant)</p> 	<p>RZ: Drehung um die Ordinate Z (TCP wird nicht bewegt)</p> 

13.3 TOOL (Bewegung in Bezug aufs Werkzeugsystem)

Das Koordinatensystem ist an dem Werkzeug, welches an der Achse 6 installiert ist, definiert. Vorgänge, die auf dieses Koordinatensystem gestützt sind, werden sich mit der Bewegungsrichtung abhängig von der Transformation der Koordinaten an die Null-Koordinaten des Werkzeugs richten. Die Koordinaten des Werkzeugs ändern sich ebenfalls mit der Änderung der Handgelenk-Orientierung, wie es auf nachfolgenden Abbildungen gezeigt wurde, obwohl sich nur der Unterarm bewegt, ohne dass Handgelenkachsen bewegt werden.

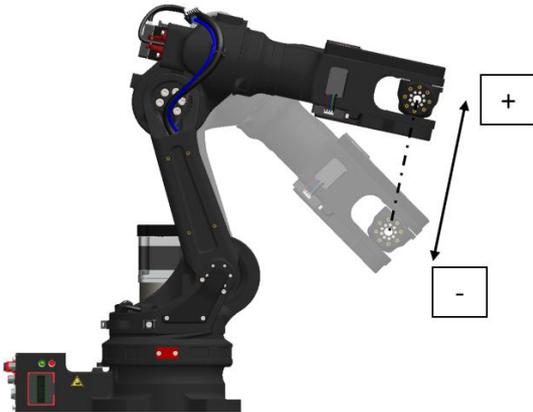


Der Arm an einem anderen Ort und Orientierung

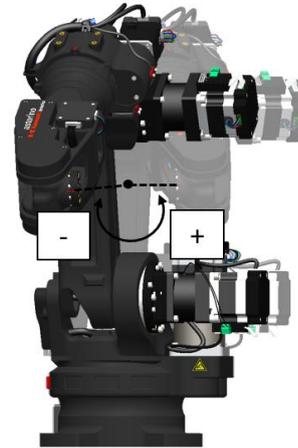


ASTORINO Betriebsanleitung

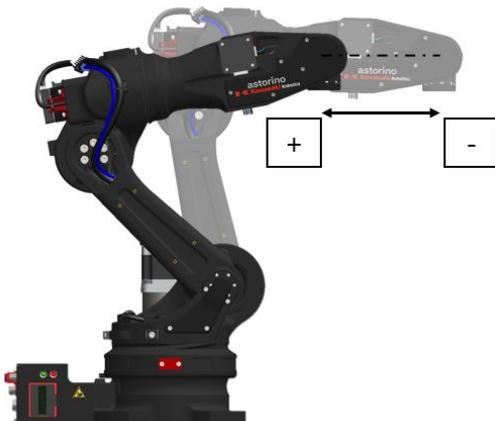
x: parallele Bewegung zur Ordinate X des Werkzeugs (die Orientierung des Handgelenks ist konstant)



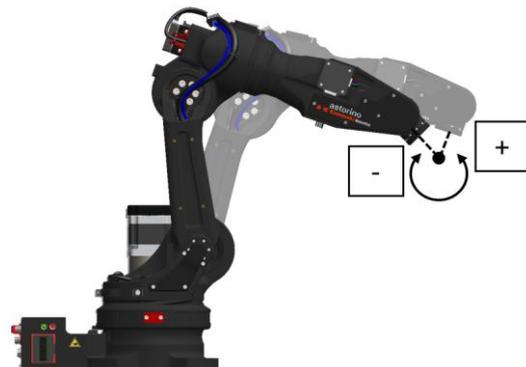
rx: Drehung um die Ordinate X des Werkzeugs (TCP wird nicht bewegt)



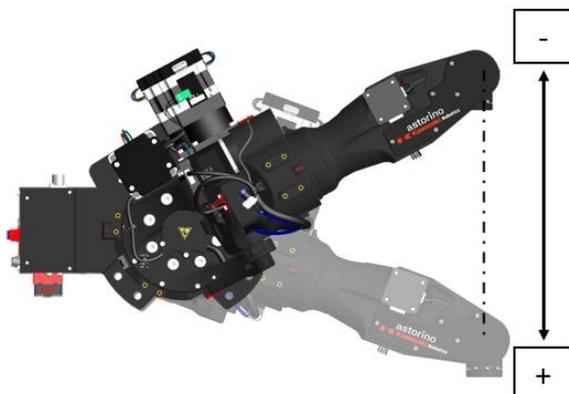
y: parallele Bewegung zur Ordinate Y des Werkzeugs (die Orientierung des Handgelenks ist konstant)



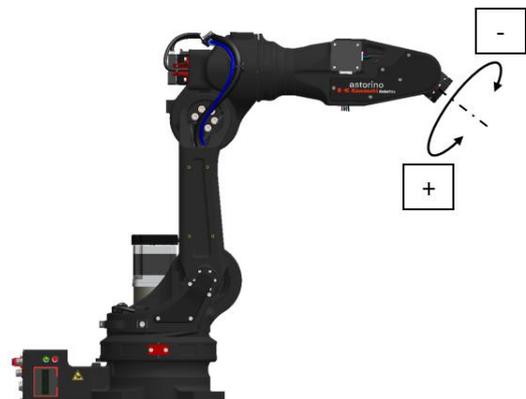
ry: Drehung um die Ordinate Y des Werkzeugs (TCP wird nicht bewegt)



z: parallele Bewegung zur Ordinate Z des Werkzeugs (die Orientierung des Handgelenks ist konstant)

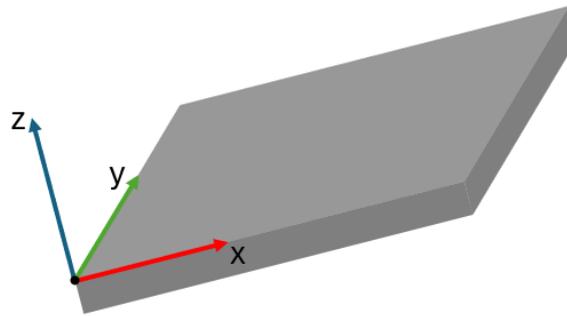


rz: Drehung um die Ordinate Z des Werkzeugs (TCP wird nicht bewegt)

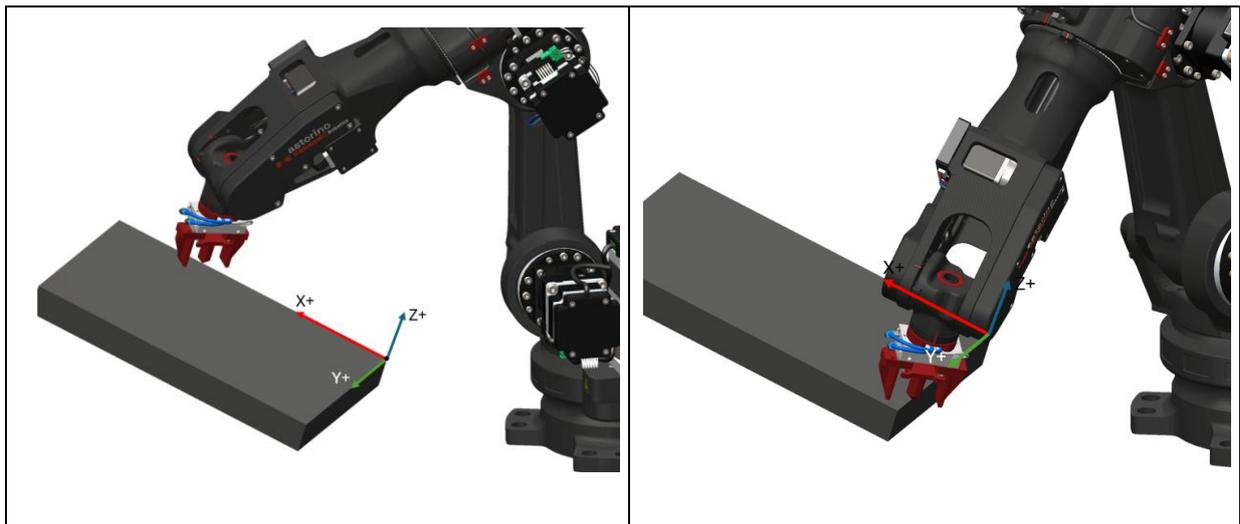


13.4 WORK

Das Koordinatensystem WORK wird auf beliebiger Ebene im Raum, der sich in Bewegungsreichweite des Roboters befindet, definiert. Vorgänge, die auf dieses Koordinatensystem gestützt sind, werden sich mit der Bewegungsrichtung abhängig von der Transformation der Koordinaten vom BASE-System unterscheiden.

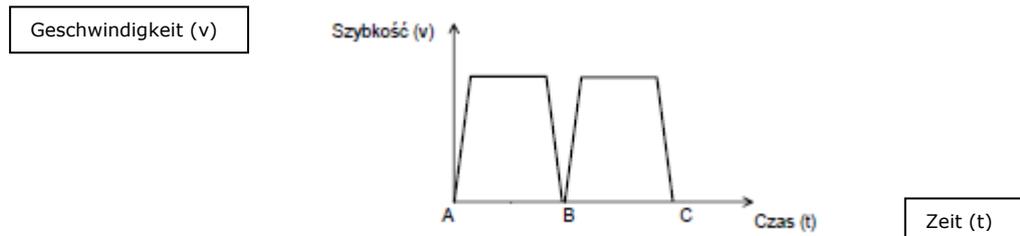


Roboterarm in verschiedenen Posen in Bezug auf das WORK-System



14 Roboterbewegung

Die Beschleunigung für das zweite Segment beginnt nach Beendigung der Umsetzung des ersten Segmentes, wenn sich die aktuelle Stellung am Bestimmungspunkt befindet. Die Neigung des Geschwindigkeitsanstiegs wird durch den ACCEL-Parameter und die Abbremslinie durch den DECEL-Parameter bestimmt.



Der Astorino-Roboter kann sich auf drei verschiedene Arten bewegen. Diese Arten werden Interpolationen genannt. Wir können folgende nennen:

- Lineare Interpolation
- Punkt-zu-Punkt-Interpolation
- Kreisinterpolation

Bei antropomorphischen Armen des Roboters (6 Achsen) sind gewisse Positionen vorhanden, die Singularitäten genannt werden. Eine einzelne Position, bei welcher das Problem einer strukturellen unkontrollierten Lage auftreten kann, ist dann vorhanden, wenn zum Beispiel JT4 und JT6 parallel zueinander oder JT1 i JT6 parallel zueinander verlaufen. Diese Konfigurationen liefern viele mathematische Lösungen der Aufgabe der inversen Kinematik, daher kann die Bewegung über diese Punkte unvorhersehbar sein und viele sehr schnelle Achsenbewegungen einführen.

Beispiele der Singularitäten

JT4 und JT6 sind parallel

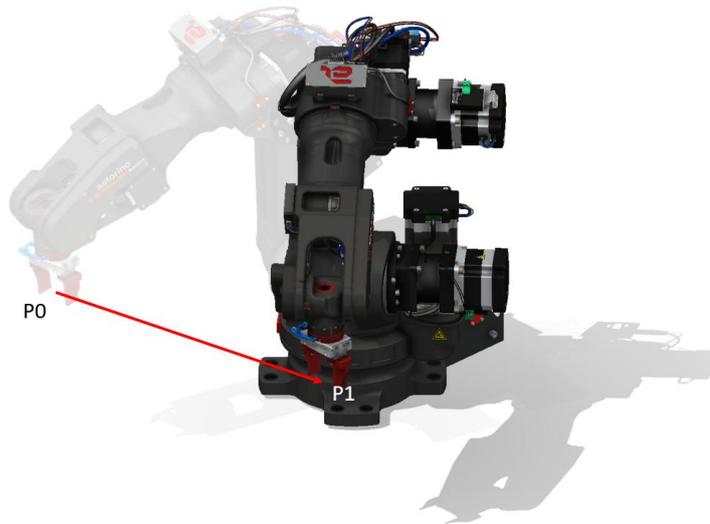


JT1 und JT6 sind parallel



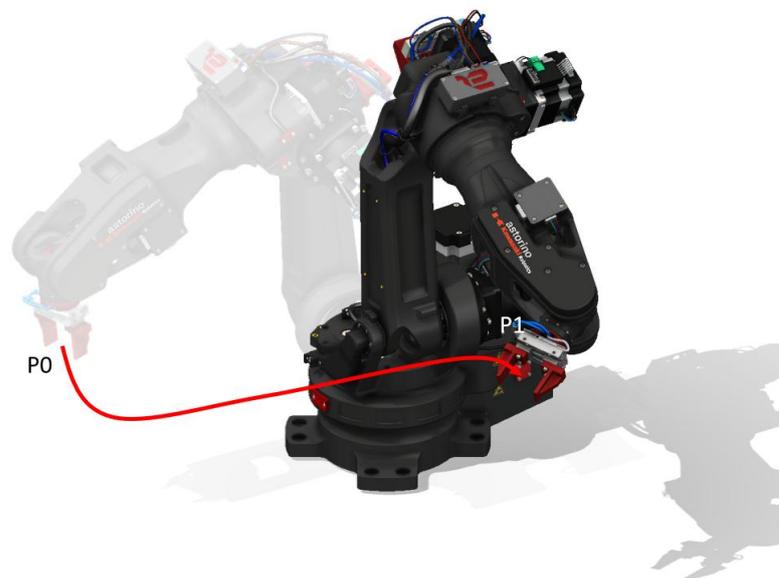
14.1 Lineare Interpolation

Bei der Interpolation dieser Art bewegt sich der Roboter von der aktuellen Position auf den Bestimmungsort so hin, dass sich TCP geradlinig im 3D-Raum bewegt.



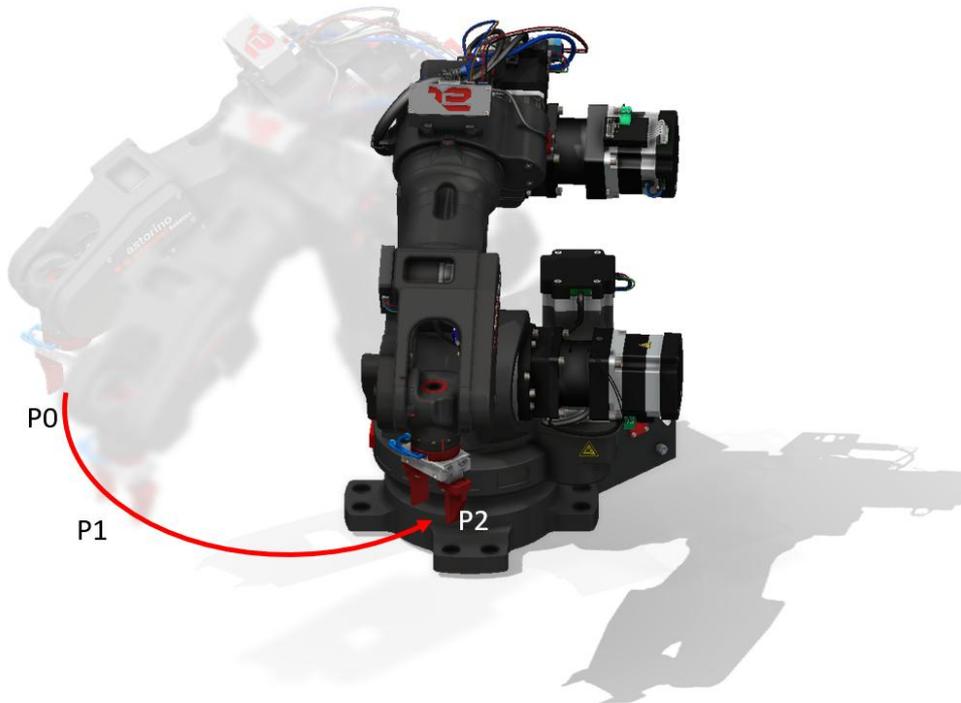
14.2 Punkt-zu-Punkt-Interpolation

Bei diesem Typ der Interpolation bewegt sich der Roboter von der aktuellen Position auf den Bestimmungsort so hin, dass alle Achsen die Bewegung zu demselben Zeitpunkt enden. Diese Bewegung bildet einen unvorhersehbaren TCP-Pfad im 3D-Raum. Diese Bewegung erlaubt das Problem der Bewegung über singuläre Punkte zu vermeiden.



14.3 Kreisinterpolation

Bei diesem Bewegungstyp bewegt sich Roboter von der aktuellen Position auf den Bestimmungsort über den Mittelpunkt so hin, dass TCP im 3D-Raum einen Bogen bildet.

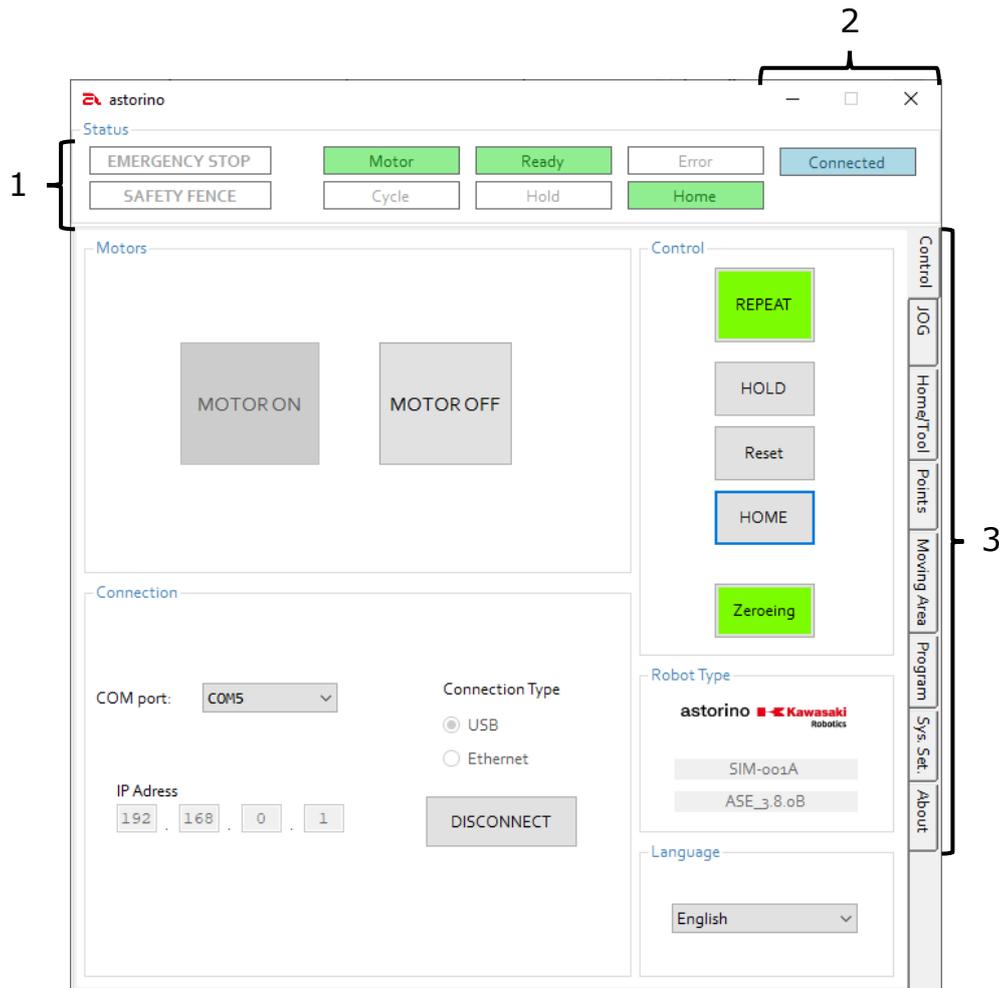


15 Astorino-Software

15.1 Grundlegende Informationen

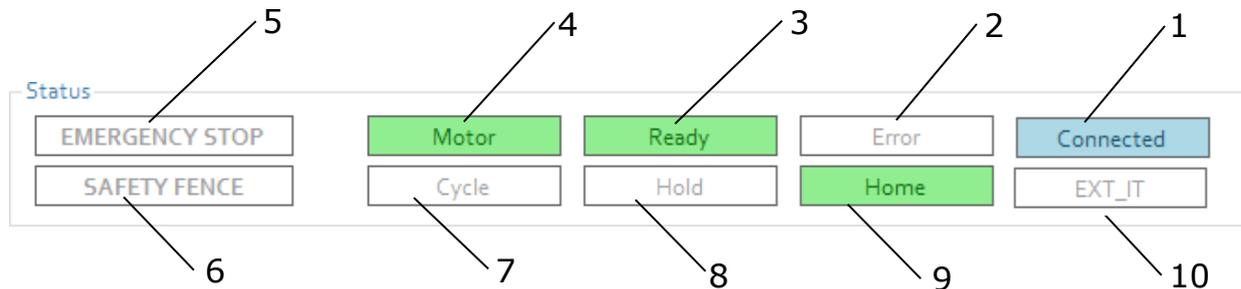
Sämtliche Daten werden auf einer internen microSD-Karte gespeichert, die sich auf der Platte des Mikrokontrollers im Sockel des Roboters befindet. Wenn der Roboter ausgeschaltet ist, werden die Daten des Nutzers nicht gelöscht.

Hauptfensterübersicht.



- | | |
|------------------------------|--|
| 1. Status | Aktueller Status des angeschalteten Roboters |
| 2. Kontrolle der Anwendungen | Schließen oder Minimieren der Astorino-Anwendung |
| 3. Vorgangskarten | Umschalten zwischen verschiedenen Vorgangskarten |

15.2 Bereich Status



Wenn der Hintergrund des jeweiligen Feldes hinterleuchtet wird, bedeutet das:

1. Connected	An die Astorino-Software angeschlossen
2. Error	Ein Fehler ist aufgetreten
3. Ready	Wenn kein Not-Halt vorkommt und keine Fehler anstehen, sind die Antriebe eingeschaltet und es wurde auf null gesetzt
4. Motors	Antriebe sind eingeschaltet
5. EMERGENCY STOP	Die Not-Halt-Taste ist betätigt und aktiv
6. SAFETY FENCE	Das Anhalten des Sicherheitszauns ist aktiv
7. Cycle	Wenn die Programmsequenz aktiviert ist
8. Hold	Wenn der Roboter angehalten wurde
9. Home	Der Roboter befindet sich in der Haus-Position
10. EXT_IT	Wenn der Roboter durch ein externes Signal angehalten wurde

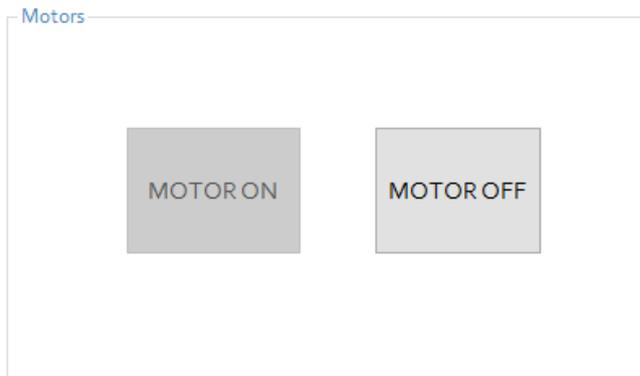
15.3 Control-Karte / Steuerung



MOTORS	Zustand der Motoren und Motorkontrolle
CONTROL	Steuerung des Betriebsmodus, Anhalten des Roboters oder Übergang zur Anfangsposition, Fehlerbestätigung und auf null setzen
ROBOTER-TYPE	Version der Firmware des Roboters und Seriennummer
Connection	Auswahl der Verbindung und Konfiguration der Schnittstelle, Festlegung der Verbindung oder Unterbrechung
Language	Auswahl der anzuzeigenden Sprache

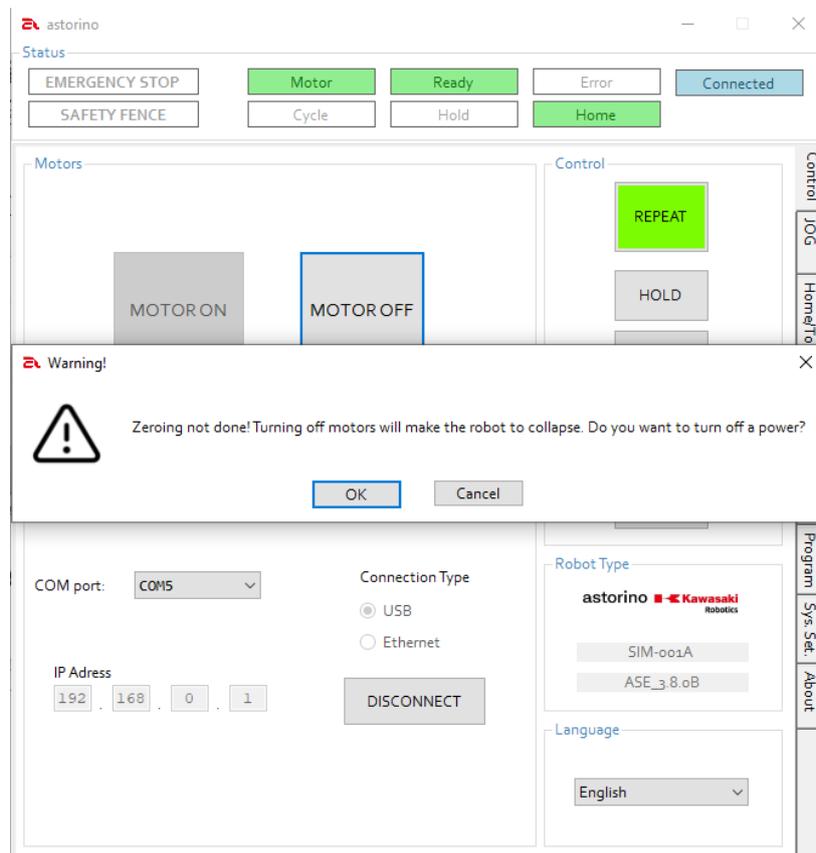
ASTORINO Betriebsanleitung

15.3.1 Bereich Motoren / Antriebe



[MOTOR ON] klicken, um die Steuereinheiten der Schrittmotoren zu aktivieren. Das ist nur dann möglich, wenn kein Fehler ansteht! (Fehlerfeld)

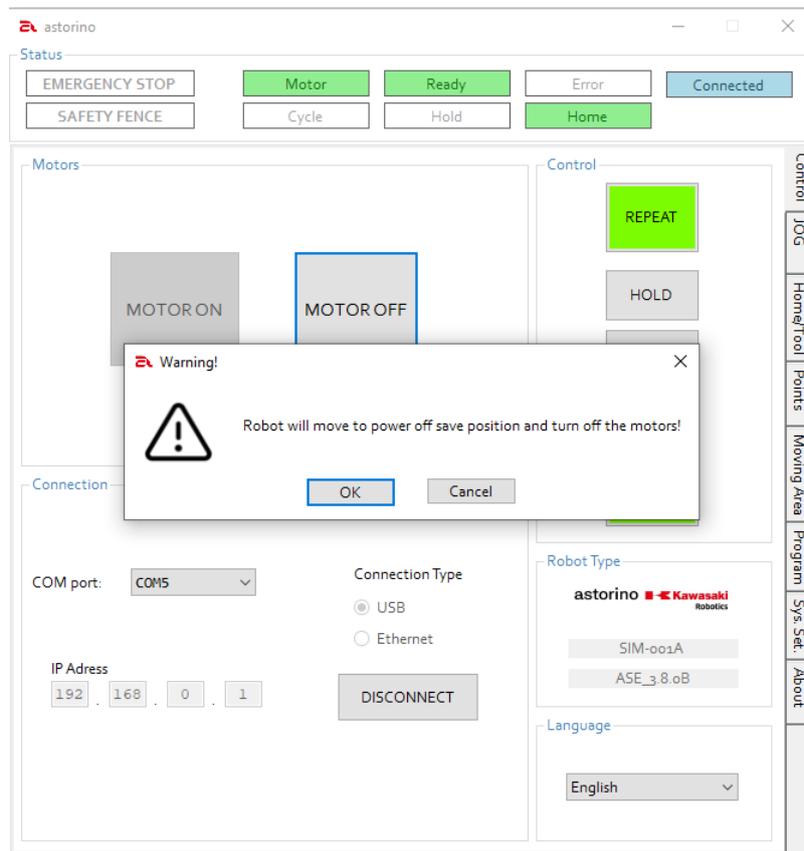
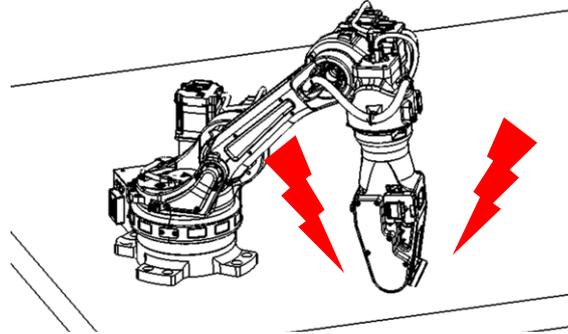
Sind die Motoren bereit, wird dies im Bereich **Status** (Motors) angezeigt.



MOTOR OFF klicken, um die Steuereinheiten des Schrittmotors abzuschalten.

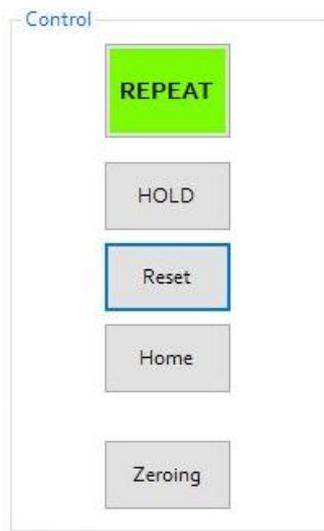
 **WARNHINWEIS**

Wenn die Achse nicht auf null gesetzt wurde, ist besondere Vorsicht geboten und die Achse 4 festzuhalten, sonst kann der Roboter stürzen.



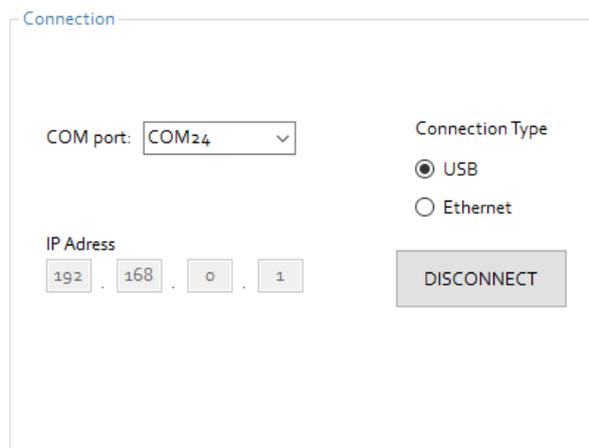
Wenn es [Zeroing] beendet wurde, begibt sich Astorino automatisch in die sichere Abschaltposition, nachdem die Warnmeldung bestätigt wurde.

15.3.2 Bereich Control / Steuerung



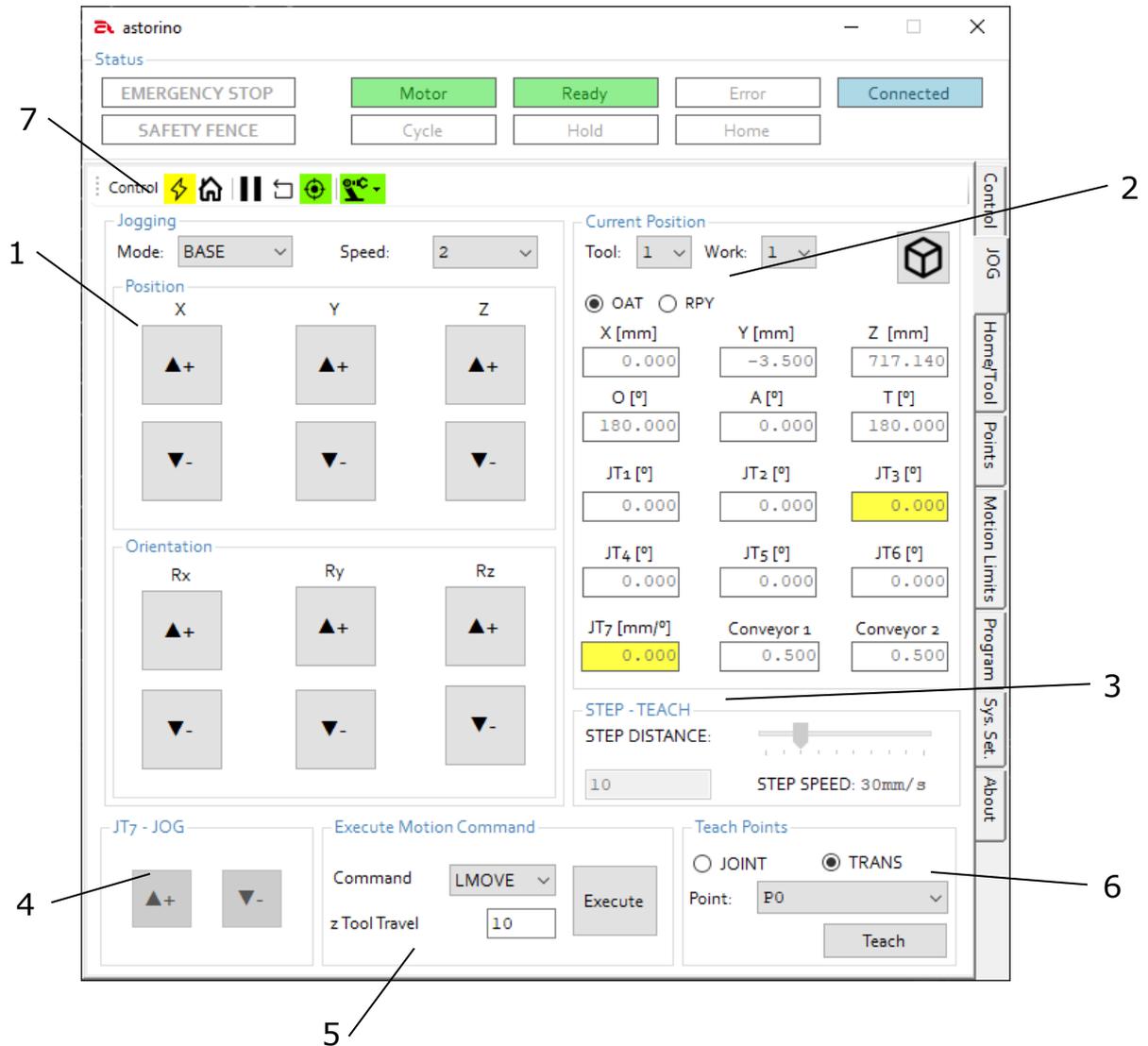
1. **[REPEAT/TEACH]** – diese Taste ist zu klicken, um das Betriebsmodus zu wechseln. Die Taste wechselt zu grün, wenn er im Automatikmodus (REPEAT) ist und zu blau, wenn er im Lehrmodus (TEACH) ist.
2. **[HOLD]** – klicken, um den Roboter anzuhalten.
3. **[Reset]** – klicken, um Fehler zu quittieren.
4. **[Home]** – klicken, damit der Roboter die Bewegung zur Haus-Position durchführt.
5. **[Zeroing]** – klicken, um die Achsen des Roboters auf null zu setzen (z. B. nach jedem Neustart).

15.3.3 Bereich Connection / Verbindung



1. COM-Port – zeigt die COM-Porte an, an welche der Roboter angeschlossen ist.
2. Connection Type - den Typ der verwendeten Verbindung (USB/ Ethernet) auswählen.
3. IP Adress – die IP-Adresse des Roboters eingeben.
4. **[CONNECT/DISCONNECT]** – Taste klicken, um sich mit dem z Roboter zu verbinden oder ihn abzuschalten.

15.4 JOG-Karte



1. Jogging

Modus der Bewegung und Geschwindigkeit auswählen

2. Current Position

Auswahl des Werkzeugs (Tool), Anzeigen der Winkel*, aktuelle Roboterposition

3. STEP-TEACH

Schrittgröße und Schrittgeschwindigkeit einstellen

4. JT7 – JOG

Linienachse verschieben (JT7) – wenn sie verfügbar ist

5. Execute Motion Command

Verfahren zum gekennzeichneten Punkt von der Sektion Teach Points

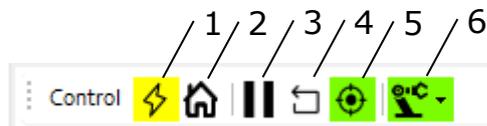
6. Teach Point

Punkt auswählen, bis zu welchem geteicht oder an welchen verfahren werden soll

ASTORINO Betriebsanleitung

7. Control

Doppelte Funktionalität von der Steuerung-Karte



1. MOTOR EINSCHALTEN/ABSCHALTEN
2. HAUSPOSITION
3. EINSTELLUNG DES BETRIEBS
4. RESET
5. AUF NULL SETZEN
6. UMSCHALTEN REPEAT/TEACH

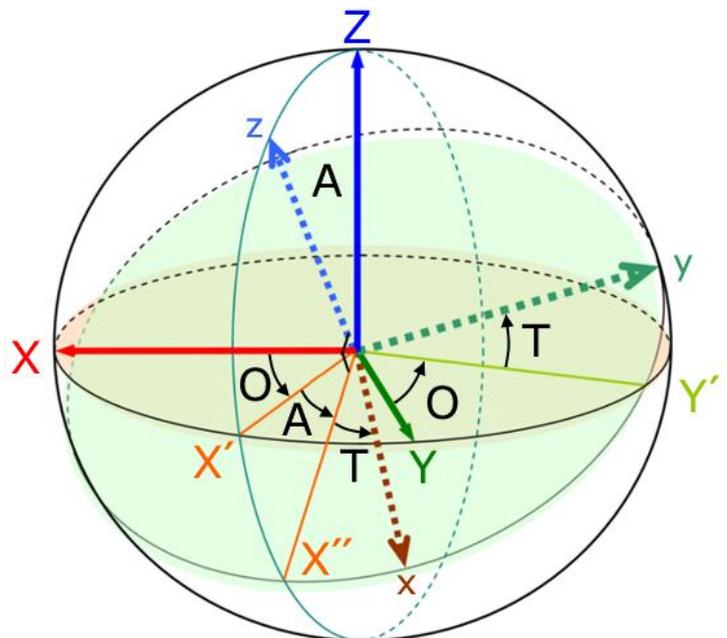
* Anzeigen des Winkels oder angewandte Drehsequenzen

Bei Berechnung der Bewegung nutzt der Roboter die EULER-Winkel (OAT) zur Berechnung des Pfads und der Position des Roboters.

Zur In-Teach-Vereinfachung wird **Roll-Pitch-Yaw** angewendet, weil er intuitiver für den Nutzer ist und demnächst wird er automatisch zur OAT-Position durch den Roboter umgewandelt.

Klassische Euler-Winkel O,A,T

Das Format der Position (POSE), das durch Kawasaki-Roboter verwendet wird, besteht aus den XYZ-Positionen in Millimetern und OAT-Orientierung, die durch drei Winkel in Graden bestimmt wird, wo sich **<O>** um die Achse **Z** dreht, **<A>** sich um die gewendete Y-Achse (**Y'**) dreht, und **<T>** sich um die gewendete Z-Achse dreht.



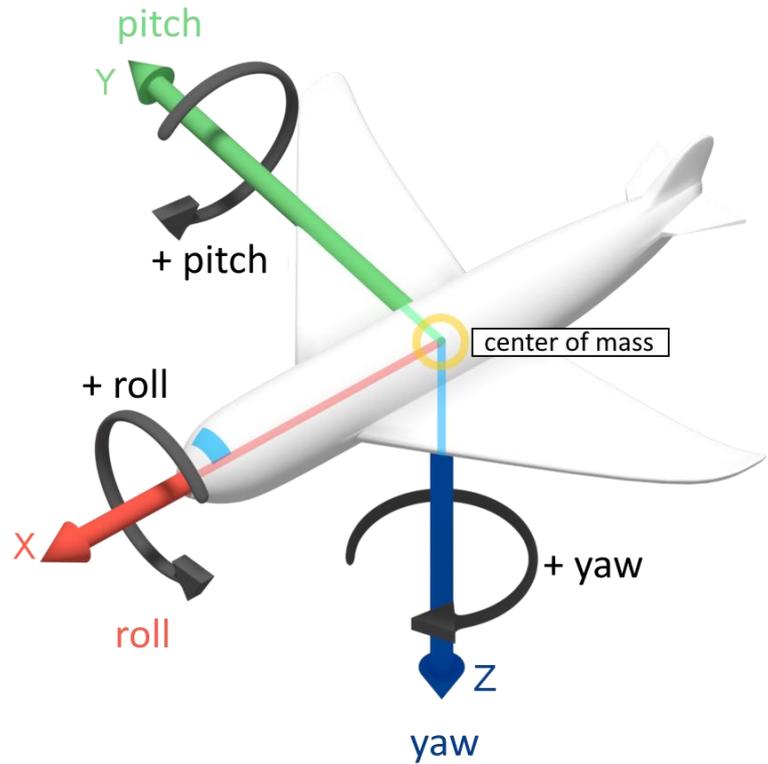
Roll-Pitch-Yaw (RPY):

Die Roll-Pitch-Yaw-Winkel sind spezielle Euler-Winkel (Lagewinkel), die zur Beschreibung der Orientierung eines Objektes im dreidimensionalen Raum angewendet werden.

Roll, X-Drehachse

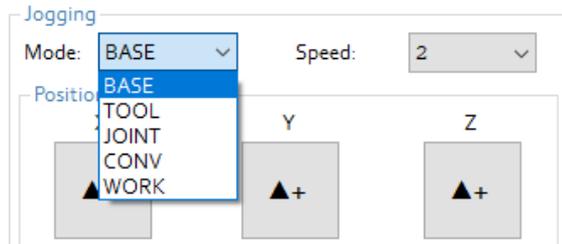
Pitch, Y-Drehachse

Yaw, Z-Drehachse



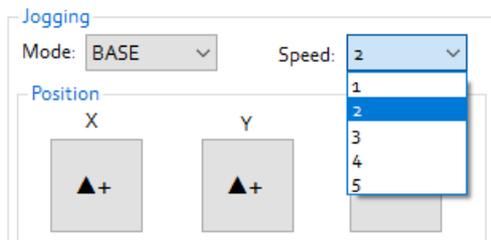
ASTORINO Betriebsanleitung

15.4.1 Jogging-Bereich



Bezugssystem auswählen:

- BASE (Basis-Koordinaten)
- TOOL (Werkzeug-Koordinaten)
- JOINT (Einachsen-Modus)
- CONV (synchron in Verbindung mit optionalem Außen-Encoder - Förderer)
- WORK (Koordinaten des WORK-Systems)

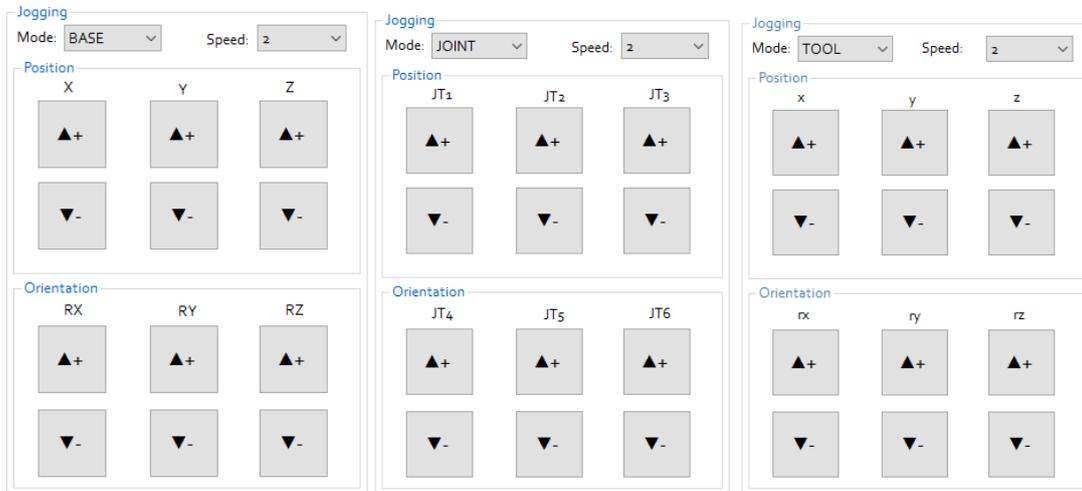


Einstellung der Bewegungsgeschwindigkeit des Roboters.

Geschwindigkeit	Kartesisches Modus C	Einachsen-Modus (JOINT)
1	Bewegung in Entfernung	Drehung um festgelegten Winkel
2	5 mm/s	2°/s
3	10 mm/s	4°/s
4	30 mm/s	8°/s
5	60 mm/s	12°/s

ASTORINO Betriebsanleitung

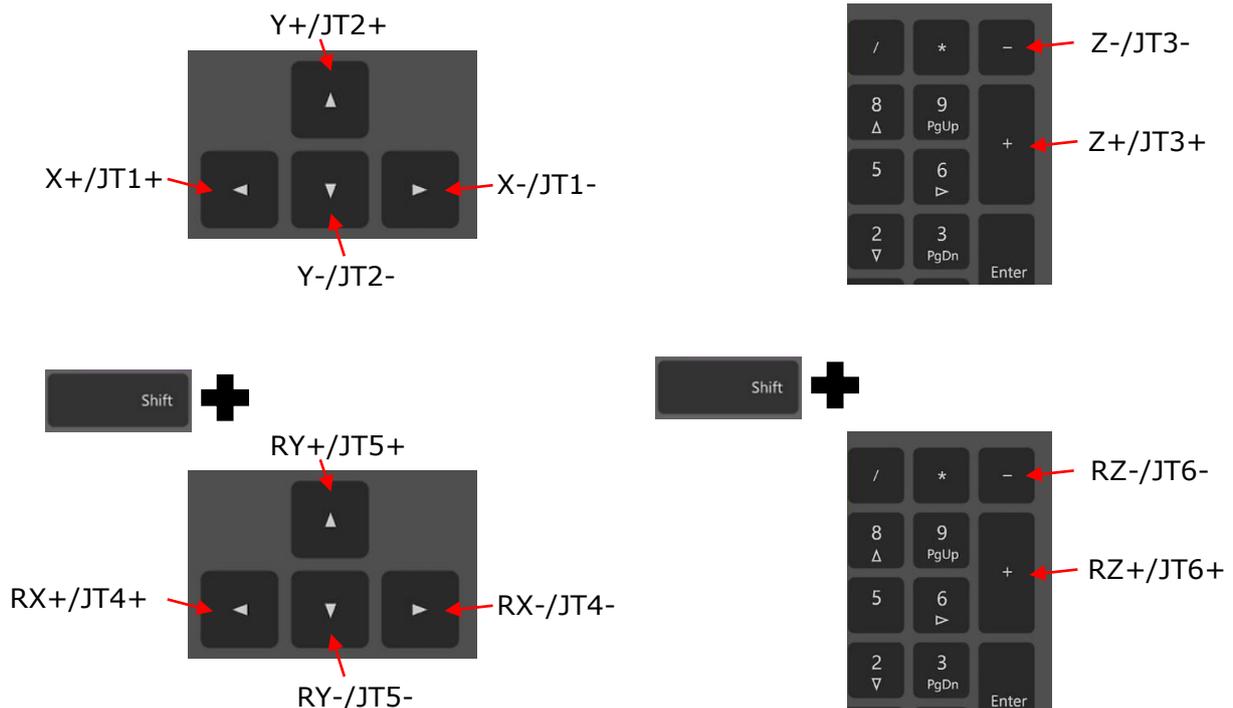
"▲+" i "▼-" bewegen den Roboter im In-Teach-Modus mit ausgewählter Geschwindigkeit:



- BASE (kartesisches Koordinatensystem): X, Y, Z, RX, RY, RZ
- TOOL (kartesisches Koordinatensystem): x, y, z, rx, ry, rz
- JOINT (Einzelachse): JT1, JT2, JT3, JT4, JT5, JT6
- CONV (kartesisches Koordinatensystem): X, Y, Z, RX, RY, RZ
- WORK (kartesisches Koordinatensystem): X, Y, Z, RX, RY, RZ

Beträgt die aktuell ausgewählte Geschwindigkeit 1 (Schritt), so führt der Roboter nach dem Drücken der JOG-Tasten (+/-) eine Schrittbewegung aus. Die Schrittbewegung ist in der Sektion STEP-TEACH eingestellt.

Zusätzlich kann der Roboter mithilfe der Tastatur bewegt werden:



15.4.2 Bereich Current Position / Aktuelle Position

Current Position

Tool: 1 Work: 1  

OAT RPY

X [mm]	Y [mm]	Z [mm]
0.000	-3.500	717.140
O [°]	A [°]	T [°]
180.000	0.000	180.000
JT ₁ [°]	JT ₂ [°]	JT ₃ [°]
0.000	0.000	0.000
JT ₄ [°]	JT ₅ [°]	JT ₆ [°]
0.000	0.000	0.000
JT ₇ [mm/°]	Conveyor 1	Conveyor 2
0.000	0.500	0.500

Werkzeugauswahl

(1-3, 4 – Reservierung für Programm-System)

Visualisierung zeigen

DryRun-Modus eingeschaltet (Roboterbetrieb ohne physische Bewegung)

Auswahl des WORK-Systems (1-2)

Umschalten zwischen Anzeigen der Winkel OAT oder RPY

Aktuelle Position des Arms

15.4.3 Bereich Step - Teach / Schritt-Einstellungen

STEP - TEACH

STEP DISTANCE:



STEP SPEED: 3deg/s

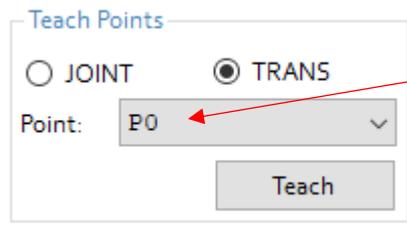
SCHRITTENTFERNUNG

Wert in mm oder ° für die Bewegung

SCHRITTSCHWINDIGKEIT

Geschwindigkeit in % oder mm/s

15.4.4 Bereich Teach Point / Punkt Einteachen

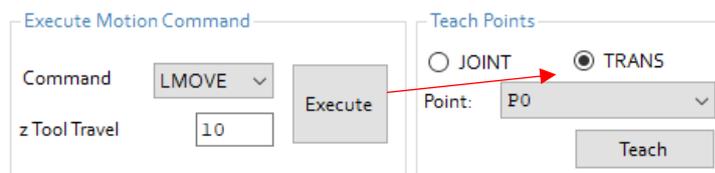


Den zu speichernden Punkt aus der Liste wählen.

Auswählen, ob der Punkt in kartesischen Werten (TRANS) oder in Achsenwinkeln (JOINT) gespeichert werden soll.

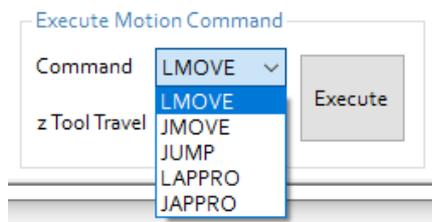
Zum Schluss die Taste [Teach] klicken, um diesen Punkt im Roboterspeicher zu speichern.

15.4.5 Bereich Execute Motion Command / Bewegungsbefehl ausführen

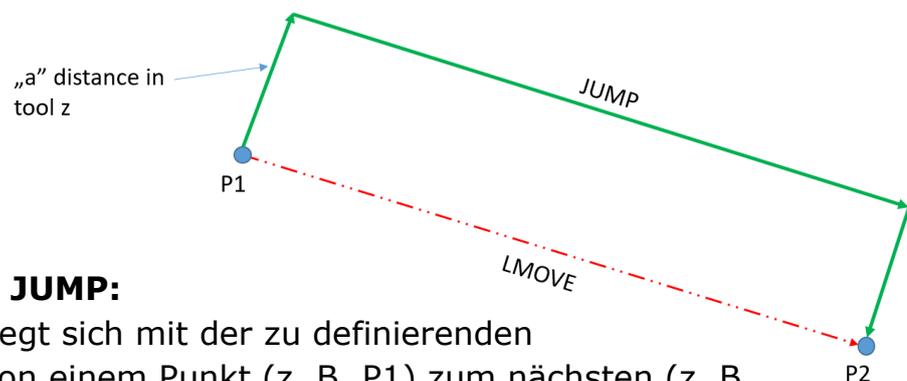


Command (mögliche Bewegungsbefehle):

- LMOVE (linear)
- JMOVE (Punkt-zu-Punkt)
- JUMP (Sonder-*)
- LAPPRO
- JAPPRO



Nachdem die Taste [Execute] angeklickt wurde, wird der Befehl für den im Bereich Teach Point ausgewählten Punkt ausgeführt.

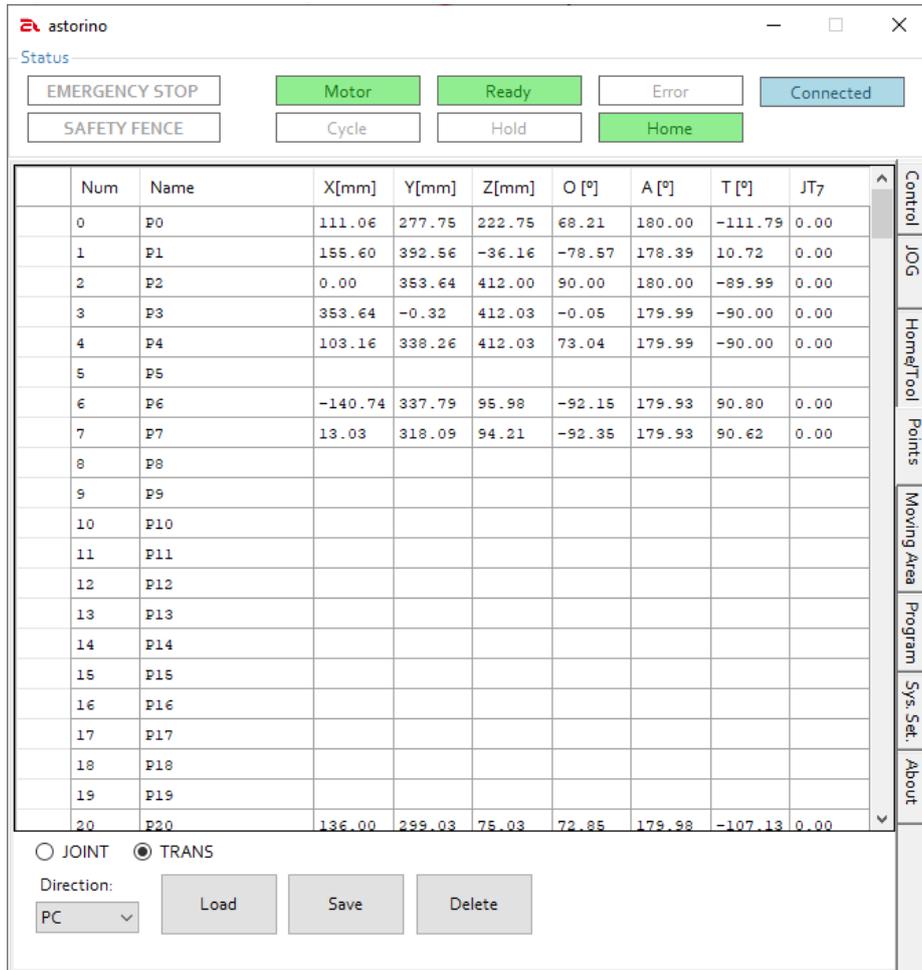


Sonderbefehl * JUMP:

Der Roboter bewegt sich mit der zu definierenden "a"-Entfernung von einem Punkt (z. B. P1) zum nächsten (z. B. P2).

Die "a"-Entfernung ist die Richtung Z in Koordinaten des Werkzeugs (TOOL).

15.5 Karte Points / Punkte



Num	Name	X[mm]	Y[mm]	Z[mm]	O [°]	A [°]	T [°]	JT7
0	P0	111.06	277.75	222.75	68.21	180.00	-111.79	0.00
1	P1	155.60	392.56	-36.16	-78.57	178.39	10.72	0.00
2	P2	0.00	353.64	412.00	90.00	180.00	-89.99	0.00
3	P3	353.64	-0.32	412.03	-0.05	179.99	-90.00	0.00
4	P4	103.16	338.26	412.03	73.04	179.99	-90.00	0.00
5	P5							
6	P6	-140.74	337.79	95.98	-92.15	179.93	90.80	0.00
7	P7	13.03	318.09	94.21	-92.35	179.93	90.62	0.00
8	P8							
9	P9							
10	P10							
11	P11							
12	P12							
13	P13							
14	P14							
15	P15							
16	P16							
17	P17							
18	P18							
19	P19							
20	P20	136.00	299.03	75.03	72.85	179.98	-107.13	0.00

In diesem Reiter werden alle in der Software gespeicherten Punkte in tabellarischer Form angezeigt.

Man kann alle TRANS- oder JOINT-Punkte anzeigen. Die Punkte von 0 bis 99 sind als Px gekennzeichnet, zum Beispiel P0 oder P10, die Punkte von 100 bis 255 sind Punkte des Nutzers und haben einen vom Nutzer festgelegten Namen.

Aus der Dropdown-Liste kann man Richtung auswählen, in welcher die Daten des Punktes geladen oder gespeichert werden sollen. Sie können angeschlossenen Computer oder die Steuereinheit des Roboters auswählen.

Die Taste **[LOAD]** lädt die Daten aus dem Speicher des Roboters oder aus Dateien *.loc zur Steuereinheit des ASTORINO-Roboters.

Mithilfe der Taste **[SPEICHERN]** Sie die Daten im Roboter-Speicher oder in der Datei *.loc in der Datei *.loc am Computer.

Die Taste **[LÖSCHEN]** löscht das ausgewählte Element aus der Astorino-Software und aus dem Roboterspeicher.

15.6 Karte Home/Tool / Hausposition/Werkzeug

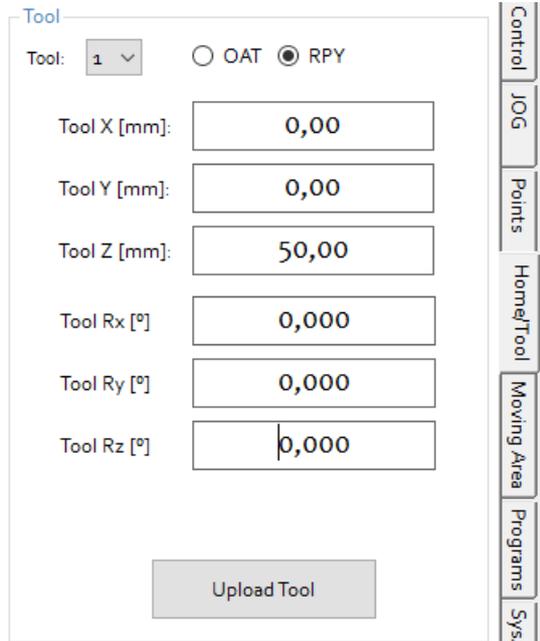
Bereich Hausposition/Home

Die Taste **[Set Home]** dient zur Speicherung der aktuellen Position des Roboters als Haus-Position.

Auswahlfeld Manual klicken, um manuell die Daten der HOME-Position einzupflegen. Demnächst die Taste **[Set Home]** klicken, um die Daten im Roboterspeicher zu speichern.

Das Drücken der Taste **[Home]** im Reiter Control bewirkt, dass der Roboter künftig in die gespeicherte Position zurückkehrt!

15.6.1 Bereich Tool



Tool: 1 OAT RPY

Tool X [mm]: 0,00

Tool Y [mm]: 0,00

Tool Z [mm]: 50,00

Tool Rx [°]: 0,000

Tool Ry [°]: 0,000

Tool Rz [°]: 0,000

Upload Tool

Verschiedene Werkzeuge, wie Greifer, Endstücke oder andere, kann man aufrufen und mithilfe des Auswahlmenüs des Werkzeugs parametrieren.

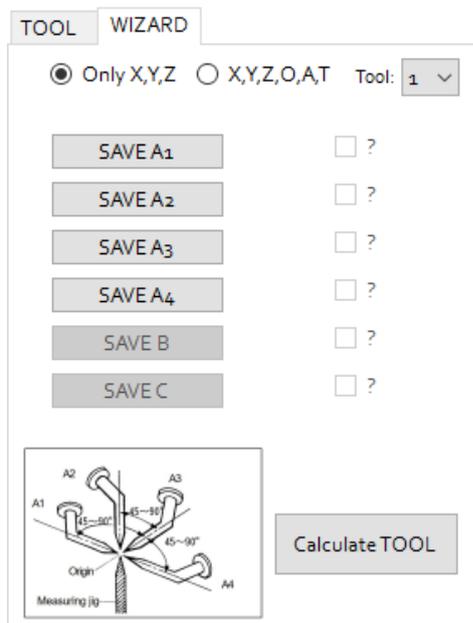
Manuell verbundene Daten des Werkzeugs eingeben.

Entweder sind diese Daten bereits bekannt und durch das Projekt dokumentiert oder Sie müssen diese selbst bestimmen und eingeben!

Setzen Sie sich mit der Sektion zur Eingabe der Werkzeugdaten in dieser Anleitung in Kenntnis.

Nachdem die Taste [Upload Tool] geklickt wurde, werden die eingepflegten Daten im Roboterspeicher aktualisiert.

15.6.2 Bereich Wizard / Kreator



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1 ?

SAVE A2 ?

SAVE A3 ?

SAVE A4 ?

SAVE B ?

SAVE C ?

Calculate TOOL

Diese Sektion ermöglicht dem Nutzer ein neues TCP (TOOL CENTER POINT) in der 4-Punkt- oder 6-Punkt-Methode zu berechnen.

Die 4-Punkt-Methode erlaubt die Werte x,y,z der Werkzeugdaten zu berechnen.

Die 6-Punkt-Methode erlaubt die Werte x,y,z und Rx,Ry,Rz (OAT) der Werkzeugdaten zu berechnen. Setzen Sie sich mit dem Sektion zur Berechnung des Werkzeugs in dieser Anleitung in Kenntnis.

15.6.3 WORK

Diese Sektion ermöglicht dem Nutzer die Werte des WORK-Systems zu prüfen und zu ändern.

Entweder sind diese Daten bereits bekannt und dokumentiert, oder sie müssen manuell bestimmt und eingepflegt werden.

Nach Anklicken der Taste [Upload Work] werden die eingepflegten Daten im Roboterspeicher gespeichert.

15.6.4 WORK WIZARD

Diese Sektion ermöglicht dem Nutzer ein neues Koordinatensystem WORK mithilfe der 3-Punkte-Methode zu berechnen.

Alle 3 Punkte auf der Bezugsebene inteachen und demnächst die Taste [Calculate Work] klicken.

15.6.5 Bereich Power OFF position / Position Abschalten

Power off position

JT1:	JT2:	JT3:	JT4:	JT5:	JT6:	JT7:	Save
0.000	-90.00	160.00	0.000	-90.00	0.000	0.000	Current
							Default

Dieser Bereich enthält Informationen zur sicheren Abschalten-Position.

- die Taste **[Save]** dient zur Speicherung der manuell eingepflegten Position.
- die Taste **[Current]** verwenden, um die aktuelle Position des Roboters zu speichern.
- **[Default]** dient zum Resetten der Werte zu Werkeinstellungen.

15.6.6 Bereich Zeroing order / Auf null setzen - Reihenfolge

Power off position Zeroing order

Default Manual SAVE

JT1	JT2	JT3	JT4	JT5	JT6	JT7
2	3	4	5	6	7	1
<input checked="" type="checkbox"/> go to 0						

Dieser Bereich enthält Informationen über die auf null setzen - Reihenfolge der Achsen. Diese Sektion erlaubt dem Nutzer die auf null setzen - Reihenfolge für alle Achsen einzustellen. Den Schritt [1..7] für alle Achsen auswählen (viele Achsen können in demselben Schritt auf null gesetzt werden) und wählen, ob sich die Achse in der Position 0 (null) nach Beendigung des Vorgangs befinden sollte oder nicht.

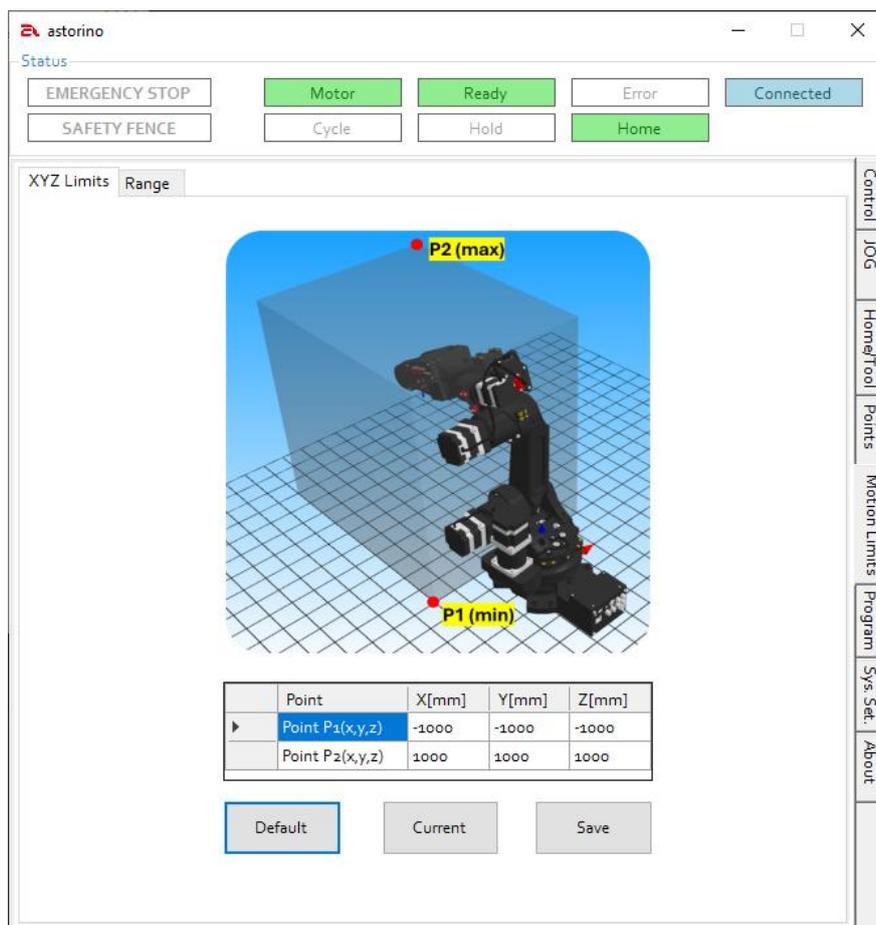
- die Taste **[Default]** stellt die auf null setzen - Reihenfolge in die voreingestellte Reihenfolge ein,
- die Taste **[Manual]** aktiviert die Sektion der manuellen Einstellungen,
- die Taste **[Save]** dient zur Speicherung der manuell eingepflegten Position.

15.7 Karte Moving Area / Arbeitsraum

In der Karte [Moving Area] kann man den erlaubten Arbeitsraum des Astorino-Roboters definieren.

Um den erlaubten Arbeitsraum modifizieren zu können, soll man auf ein höheres Zugriffsniveau steigen. Um dies zu machen, soll man im Terminal des Roboters den Befehl „z_user 3“ eintragen.

15.7.1 XYZ Limits



Mithilfe zweier Punkte **P1** und **P2** wird ein virtueller rechteckiger Umfang erstellt, der den Bereich festlegt, in welchem sich der Roboter bewegen kann und den er nicht verlassen darf.

Punkt **P1** – zeigt Mindestwerte an,

Punkt **P2** – zeigt Höchstwerte an

- mithilfe [Get Current] wird die aktuelle Position des Roboters registriert und in der ausgewählten Zeile der Positionstabelle gespeichert.
- den Bereich kann man auch manuell definieren und Werte eintragen.
- mithilfe [SAVE] werden die Daten in den Roboterspeicher übertragen.
- die Taste [Default] resettet die Werte zu Werkeinstellungen.

ASTORINO Betriebsanleitung

15.7.2 Range

astorino

Status

EMERGENCY STOP Motor Ready Error Connected

SAFETY FENCE Cycle Hold Home

XYZ Limits Range

Define limits for each robot joint

	Min		Max
JT1:	-158.00	÷	158.00
JT2:	-90.00	÷	127.00
JT3:	-166.50	÷	0.00
JT4:	-240.00	÷	240.00
JT5:	-120.00	÷	120.00
JT6:	-359.90	÷	359.90
JT7:	-0.50	÷	408.00

Default SAVE

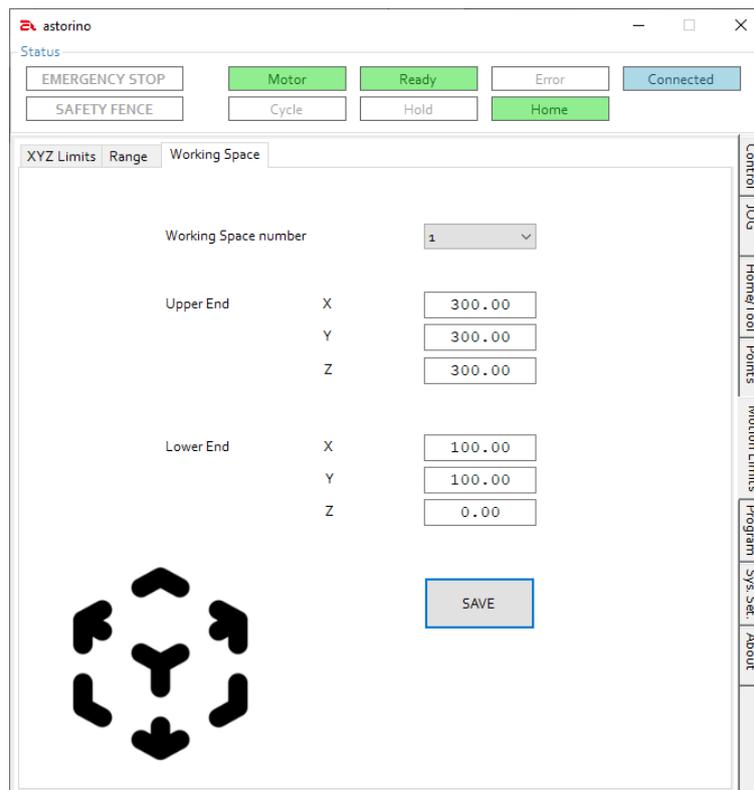
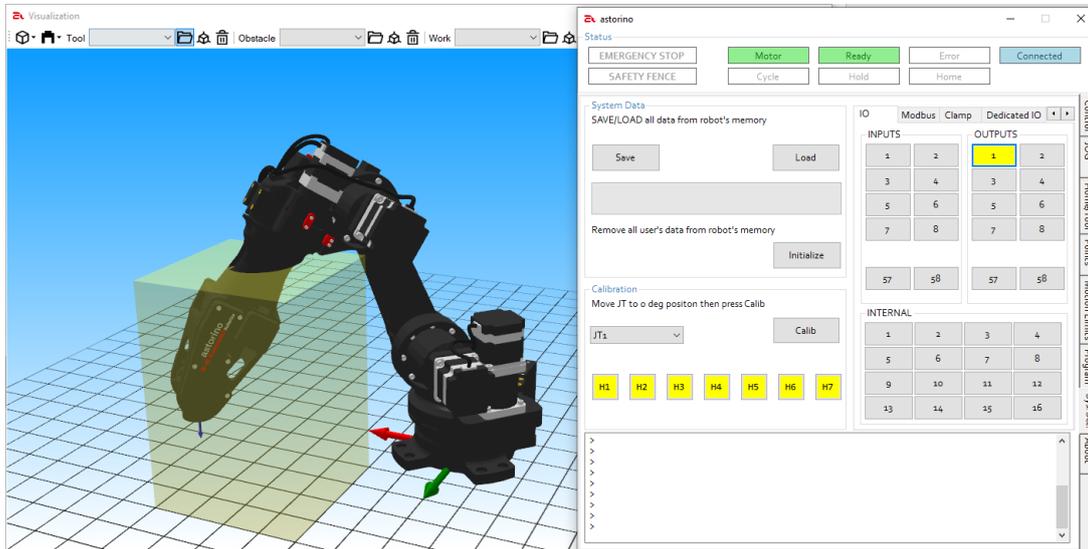
Control JOG Home/Tool Points Motion Limits Program Sys. Set. About

In diesem Reiter kann der Nutzer die Bewegungsgrenzen aller Achsen separat einstellen. Der Höchstwert und der Mindestwert vom Winkel jeder Verbindung ist einzupflegen und demnächst [SAVE] zu drücken, damit Änderungen an den Roboter gesendet werden. Das Drücken der Taste [Default] bewirkt, dass Werkeinstellungen des Achsenbereiches wieder eingesetzt werden.

ASTORINO Betriebsanleitung

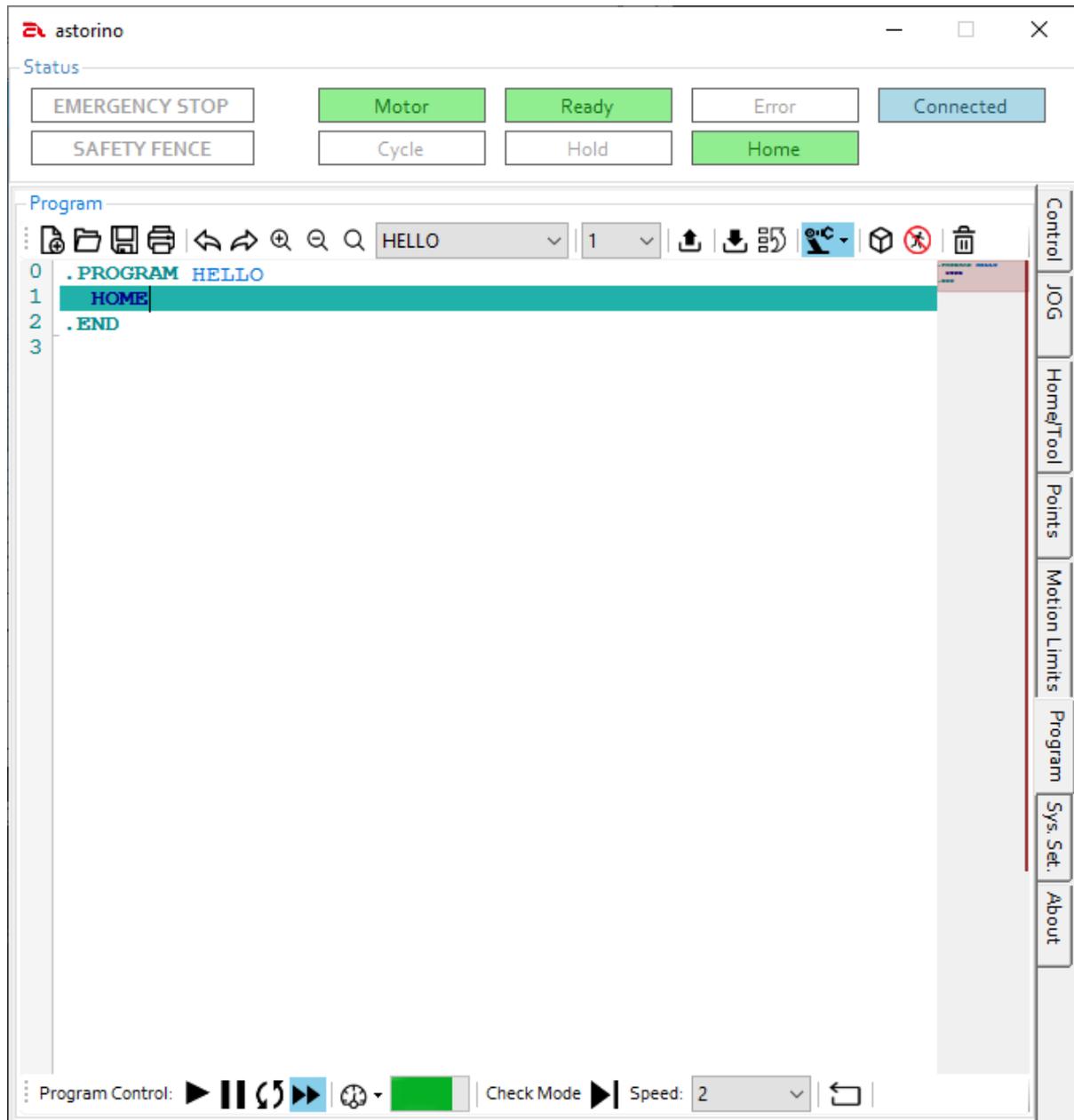
15.7.3 Working Space

Der Arbeitsraum ist ein virtueller Bereich, der zum Informieren anderer Geräte über die Lage des Roboterarms verwendet werden kann. Der Arbeitsraum nutzt die für die Arbeit dedizierten Signale.



In diesem Reiter kann der Nutzer Bereiche des Arbeitsraums einstellen. Werte des oberen und unteren Endes jeden Arbeitsraums eintragen und demnächst [SPEICHERN] drücken, um Änderungen an den Roboter zu übersenden. Sie können von bis zu vier verschiedenen Arbeitsräumen einstellen.

15.8 Karte Programm

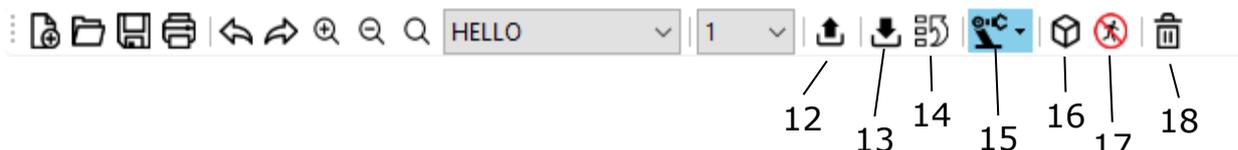


In diesem Reiter können Sie den Roboter in der vereinfachten Version der Kawasaki AS-Sprache programmieren.

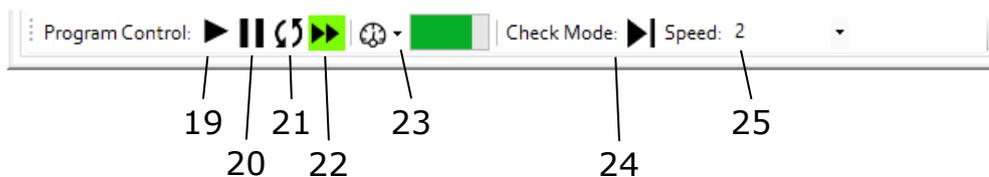
ASTORINO Betriebsanleitung



1. Erstellung eines neuen Programms
2. Programm aus dem Computer auswählen und auf den Roboter laden
3. Ausgewähltes Programm als Datei *.pg am PC-Computer speichern
4. Programm-Code drucken
5. Zurück,
6. Wieder,
7. Inhalt des Programmfensters vergrößern (Zoom)
8. Inhalt des Programmfensters verkleinern (Zoom)
9. Vergrößerung auf Voreinstellungen resetten
10. Liste der Programmauswahl,
11. Liste der Schrittwahl des ausgewählten Programms,



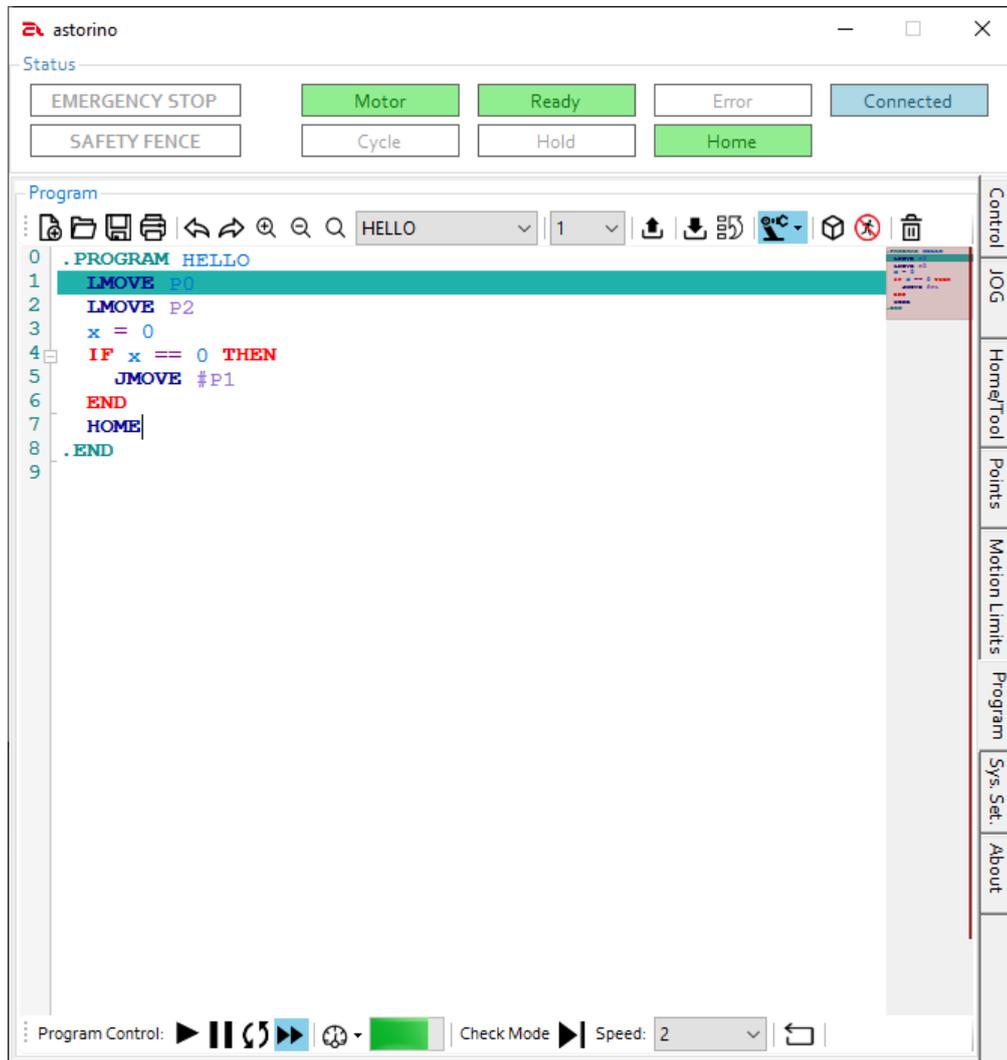
12. Spielt das Programm in die Steuereinheit des Roboters ein (PC ⇌ Astorino)
13. Lädt das ausgewählte Programm aus der Steuereinheit des Roboters herunter
14. Stellt das ausgewählte Programm als Startprogramm ein *
15. Wechsel des Betriebsmodus TEACH / REPEAT (manuell / automatisch)
16. Visualisierungsfenster eröffnen
17. Aktiviert das DryRun-Modus (ohne Roboterbewegung)
18. Löscht das markierte Programm



19. Programm starten oder stoppen
20. HOLD – hält das gestartete Programm an
21. Repeat Continues – aktiviert das Modus Programmschleife
22. Step Continues – aktiviert das Modus Einzelschritt
23. Monitor speed - Geschwindigkeitsüberwachung
24. Check mode – zeigt das aktuell aktive Schritt- oder Dauermodus an.
Taste Next Step – erlaubt zu weiteren Schritten im Schrittmodus zu übergehen. Es funktioniert auch als Taste GO – führt das aktuell markierte Programmschritt aus,
25. Einstellung und Änderung der Geschwindigkeit im TEACH-Modus

ASTORINO Betriebsanleitung

* Nachdem Astorino eingeschaltet wurde, wird das als Startprogramm definierte Programm in den Arbeitsspeicher der Robotersteuerung geladen und ist unmittelbar bereit, in Gang gesetzt zu werden.



Ein Beispielprogramm mit dem Namen *hello*

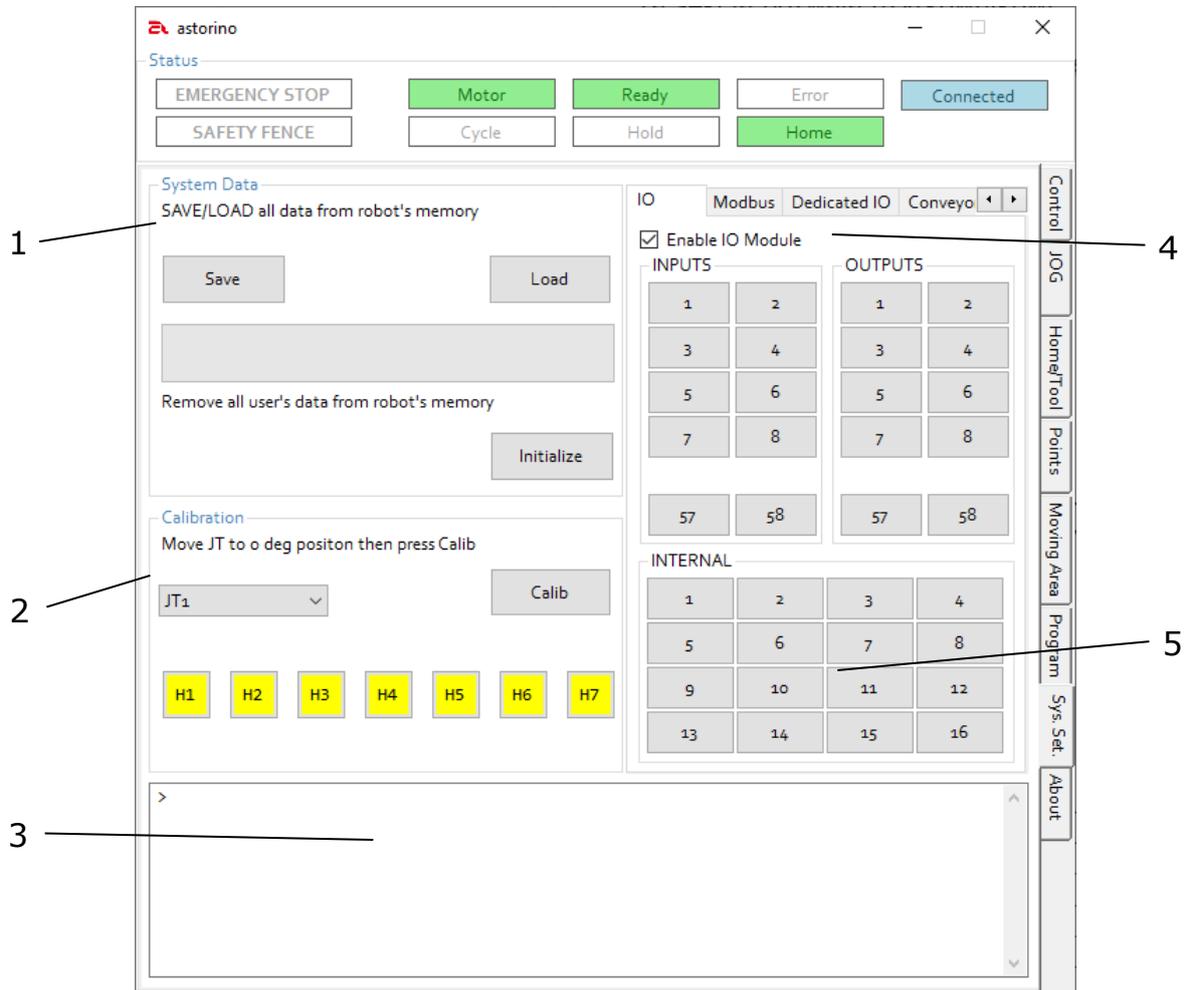
Wird das Programm ausgeführt, bewegt sich der Roboter zu dem Punkt **P1** in gerader Linie (LinearMOVE).

Nach Erreichen des Punktes bewegt er sich auf geradem Pfad zu **P2**.
Der Variablen *x* wurde jetzt der Wert 0 zugeschrieben.

Die IF-Schleife prüft, ob *x* den Wert 0 hat.

In diesem Fall wird der Befehl HOME ausgeführt und der Roboter geht in die Haus-Position über.

15.9 Karte System Settings / System-Einstellungen



System Data	Alle Daten des Roboters sind zu speichern/ zu laden oder der Roboterspeicher ist zu initialisieren.
Calibration	Die Kalibrierung des Roboterarms ist nur dann erforderlich, wenn die Daten auf der SD-Karte entfernt werden, die Karte beschädigt und durch eine neue ersetzt wird oder wenn der Roboter demontiert wird.
Terminal	Nutzerbefehle und Anzeigen der Daten.
IO, Dedicated IO, Conveyor...	Eingangs- und Ausgangsstatus (I/O), Eingang/Ausgang und andere Einstellungen konfigurieren.
INTERNAL	Steuerung der internen Signale.

15.10 Bereich Calibration / Kalibrierung



Diese Sektion erlaubt dem Nutzer den Roboter zu kalibrieren und auch magnetische Sensoren, die an allen Achsen installiert sind, zu prüfen. Wenn die Taste mit Hx-Bezeichnung, wo x von 1 bis 7 beträgt, gelb leuchtet, sieht der Magnetsensor den Magnet.

15.11 Feld Terminal



Der Terminal dient zum Anzeigen der Informationen aus dem Roboter, aber auch dazu, dem Roboter Befehle zu geben.

Alle Bewegungsbefehle, wie LMOVE, HOME usw. müssen das Wort "DO" davor haben und der Roboter muss BEREIT und im REPEAT-Modus sein. Zum Beispiel "DO LMOVE P1"

Sie können auch den Terminal zur Ablesung der variablen Werte (zum Beispiel "PRINT x"), Einteichen von Punkten (zum Beispiel HERE P1), Einstellen von Variablen (zum Beispiel x = 10) und so weiter einsetzen.

Hier ist die Befehlsliste des Terminals:

CPUTEMP	zeigt die Prozessortemperatur an
FREE	zeigt die verfügbare RAM-Speicherkapazität in % an
ERESET	resettet den Fehler
ZPOWER ON	schaltet MOTOREN ein
ZPOWER OFF	schaltet MOTOREN ab
HOLD	stellt das aktuell gestartete Programm ein
CONTINUE	setzt das eingestellte Programm fort
ZZERO x	beginnt die bestimmte x-Achse auf null zu setzen
HALT	stellt das aktuell in Betrieb gesetzte Programm ein
EXECUTE x	setzt das aktuell ausgewählte Programm in Gang, wenn der vorgegebene Name das ausgewählte Programm (x) in Gang setzt
PRIME x	wählt das ausgewählte Programm nach dem Namen aus
STOP	hält das aktuell in Betrieb genommene Programm (Zyklus) an
REP_ONCE ON/OFF	Einschalten und Abschalten der Funktion REPEAT ONCE
STP_ONCE ON/OFF	Einschalten und Abschalten STEP ONCE
STPNEXT	setzt den weiteren Schritt im Schrittmodus frei

ASTORINO Betriebsanleitung

15.11.1 Bereich Status und Konfigurierung

15.11.1.1 IO

In diesem Fenster kann man den Zustand der EINGÄNGE von Signalen sehen und die AUSGÄNGE einschalten/abschalten, indem man entsprechende Auswahlfelder anklickt.

Wird das Feld **Enable IO Module** ausgewählt, wird das Modul Eingang/Ausgang aktiviert. Wenn das IO-Modul beschädigt ist, wird dieses Feld automatisch wieder leer.

Die EINGÄNGE 57,58 und AUSGÄNGE 57,58 sind für die Version B des Roboters reserviert und befinden sich am Arm JT3

[VORSICHT]

Wird diese Funktion ohne das Anschlussmodul I/O an die Hauptplatte des Roboters aktiviert, kommt es zu einem Fehler!

INTERNAL – hier wird der Status der internen Signale geprüft oder erzwungen. (**ON** , OFF)

ASTORINO Betriebsanleitung

15.11.1.2 MODBUS

IO Modbus Dedicated IO Conveyo

Fieldbus Inputs

9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56

Fieldbus Outputs

9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56

In diesem Fenster kann man den Zustand der Eingänge FIELDBUS sehen und die FIELDBUS-Ausgänge einschalten/abschalten, wenn man entsprechende Prüfungstasten anklickt. Wenn der Eingang oder der Ausgang eingeschaltet ist, leuchtet die Taste gelb.

15.11.1.3 Clamp

IO Modbus Clamp Dedicated IO

Handling Clamp Signal Definition 

OUTPUT Signal for OFF

OUTPUT Signal for ON

In diesem Feld kann man die Einstellungen Handling Clamp ändern.

Die Clamp-Einstellungen beeinflussen die Befehle OPENI und CLOSEI.

ASTORINO Betriebsanleitung

15.11.1.4 Dedizierte Signale

Ansicht und Konfigurierung der dedizierten Roboter-Signale. Die Signale haben eine festgelegte, bestimmte Funktion oder Anleitung.

Dedizierte Eingänge sind spezielle Eingangssignale, wie "Reset" oder "Cycle Stop".

Dedizierte Ausgänge sind spezielle Ausgangssignale, wie "Motor ON" und "Error".

MZH (MOTOR ON -> ZEROING -> HOME) – das ist eine spezielle Sequenz, die zur Einhausung des Roboters mit einem Bit vom Anfangszustand nach Einschalten der Speisung dient.

15.11.1.5 Erkennen von Kollisionen

B – die Version des Roboters ist mit einem Beschleunigungsmesser zum Erkennen von Kollisionen ausgestattet.

Hier kann der Nutzer die Schwellen der Kollisionserkennung ändern.

Das Auswahlfeld [4G Mode] ermöglicht die Hochniveau-Schwellen einzuschalten/abzuschalten

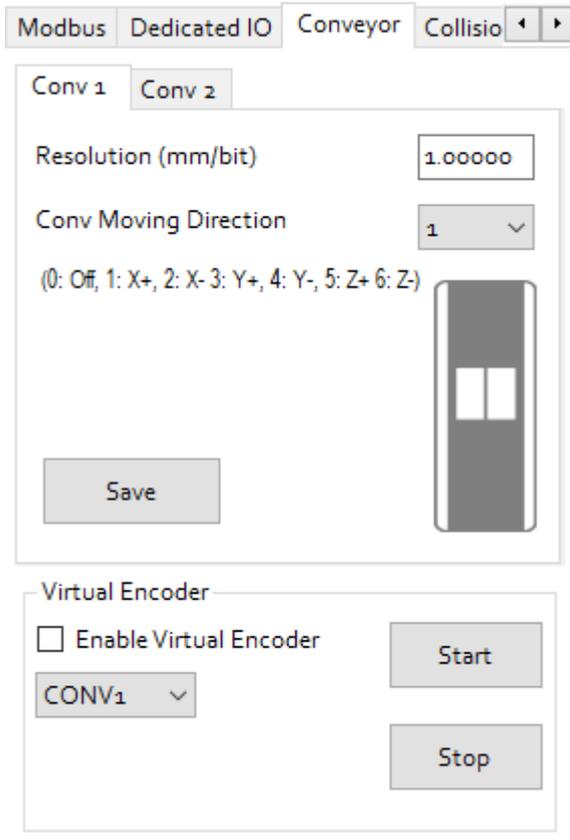
Die [Save]-Taste speichert die Schwellen in den Roboterspeicher

Die [Calib ON/OFF]-Taste beginnt oder stoppt die automatische Kalibrierung der Schwellen.

*um die Einstellungen zu ändern, ist im Terminal der Befehl „Z_USER 3“ einzutragen

ASTORINO Betriebsanleitung

15.11.1.6 Förderband

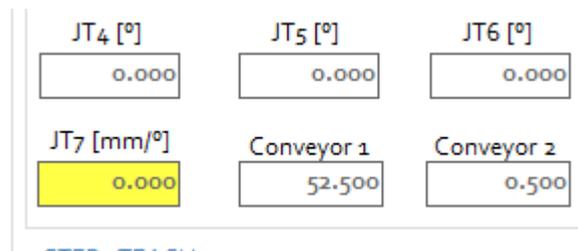


The screenshot shows a control interface with tabs for 'Modbus', 'Dedicated IO', 'Conveyor', and 'Collisio'. Under the 'Conveyor' tab, there are sub-tabs for 'Conv 1' and 'Conv 2'. The 'Conv 2' sub-tab is active. The settings include:

- Resolution (mm/bit): 1.00000
- Conv Moving Direction: 1 (with a dropdown arrow)
- Legend: (0: Off, 1: X+, 2: X- 3: Y+, 4: Y-, 5: Z+ 6: Z-)
- A 'Save' button is located at the bottom left.
- Below the main settings is a 'Virtual Encoder' section with:
 - Enable Virtual Encoder
 - A dropdown menu currently set to 'CONV1'
 - 'Start' and 'Stop' buttons.

Hier kann man die Bewegungsrichtung des Förderbands einstellen. Wir stellen die Richtung gemäß einer Achse des BASE-Systems des Roboters und auch die Auflösung mm/bit ein

Man kann auch virtuellen Encoder starten. Das erlaubt die Anwendungen zu simulieren, welche die Bandverfolgung nutzen. Wir wählen aus der Liste aus, welches Förderband (CONV1 oder CONV2) wir in Betrieb nehmen wollen, wir schalten die Checkbox (Enable Virtual Encoder) ein und demnächst mit den Tasten [Start] oder [Stop] schalten wir den virtuellen Encoder ein oder aus.



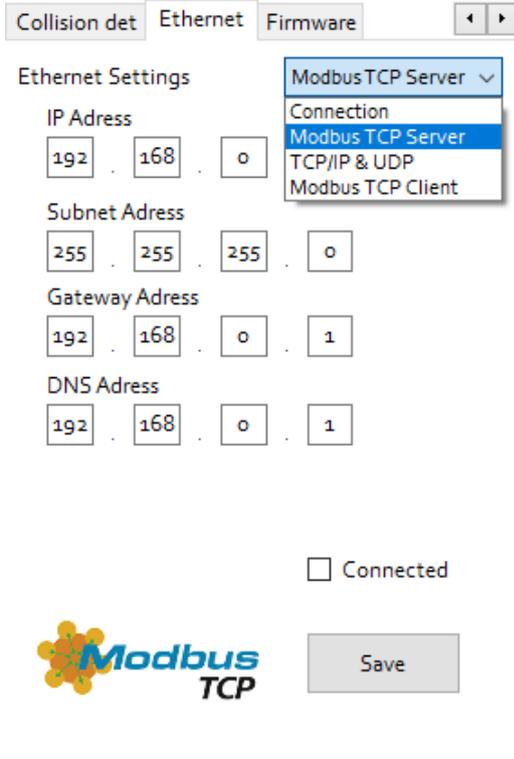
The screenshot shows a JOG display with the following data:

JT4 [°]	JT5 [°]	JT6 [°]
0.000	0.000	0.000
JT7 [mm/°]	Conveyor 1	Conveyor 2
0.000	52.500	0.500

Die Werte sowohl des virtuellen als auch des physischen Encoders werden im Reiter JOG angezeigt.

ASTORINO Betriebsanleitung

15.11.1.7 Ethernet



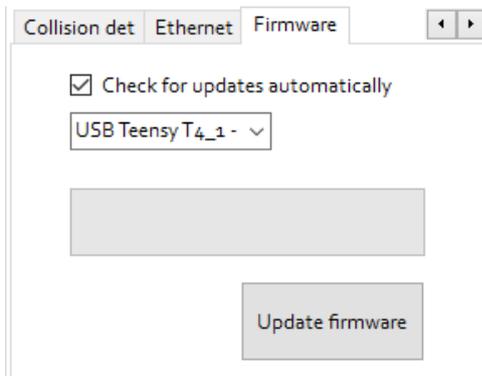
In diesem Bereich kann man die Einstellungen der Ethernet-Kommunikation ändern.

Die Funktion des Ethernet-Portes kann man einstellen für:

- die Verbindung mit der Astorino-Software
- den Modbus TCP Server
- TCP/IP oder UDP
- den Modbus TCP Client

Die [Save]-Taste speichert Änderungen im Controller-Speicher ein. Nach der Speicherung ist ein Neustart des Roboters erforderlich.

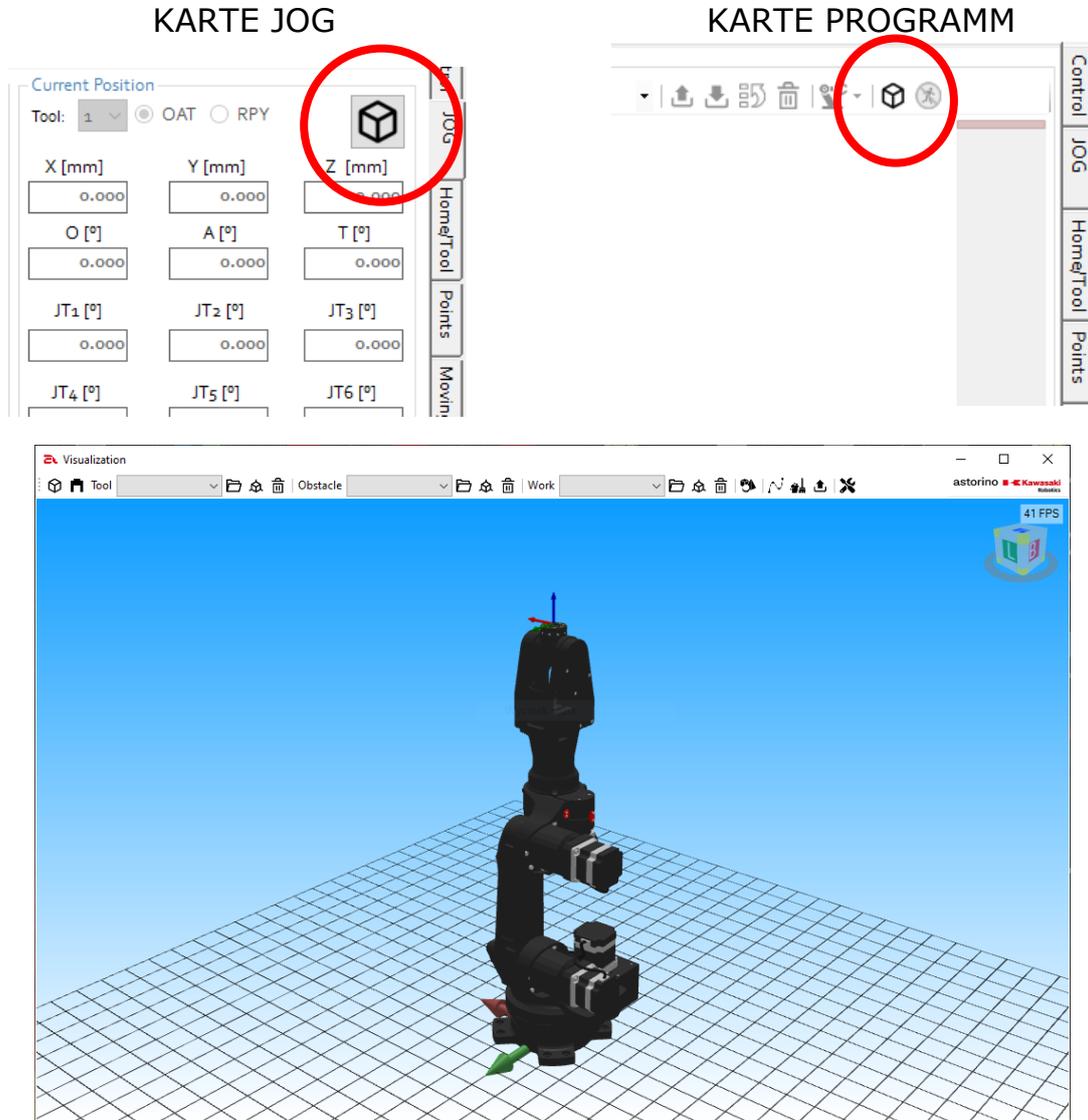
15.11.1.8 Firmware



In diesem Unterreiter können Sie die Software Astorino aktualisieren und auch einstellen, ob die Software prüfen soll, ob es am Server Aktualisierungen gibt.

15.12 Visualisierungsfenster

Um das Visualisierungsfenster aufzumachen und in Echtzeit die Arbeit des Astorino- Roboters zu sehen, ist eine von diesen zwei Tasten zu klicken



15.12.1 Bedienung des Visualisierungsfensters

Das Visualisierungsfenster erlaubt die 3D-Objekte zur Szene mit dem Roboter hinzuzufügen. Das Programm bedient die Dateien von stl-Typ und erlaubt dreidimensionale Grundformen hinzuzufügen. Jedes von Objekten kann man als eines der drei Objekttypen hinzuzufügen:

- Obstacle – Objekte von diesem Typ sind statische Objekte der Szene
- Work – Objekte von diesem Typ können durch den Roboter übertragen werden
- Tool – Objekte von diesem Typ bewegen sich immer gemäß dem Roboterflansch.

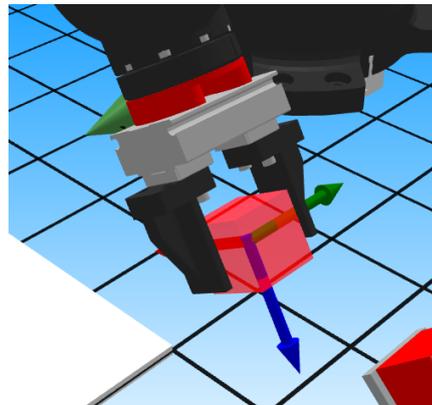
Das Menü des Visualisierungsfensters besteht aus folgenden Bestandteilen:

ASTORINO Betriebsanleitung

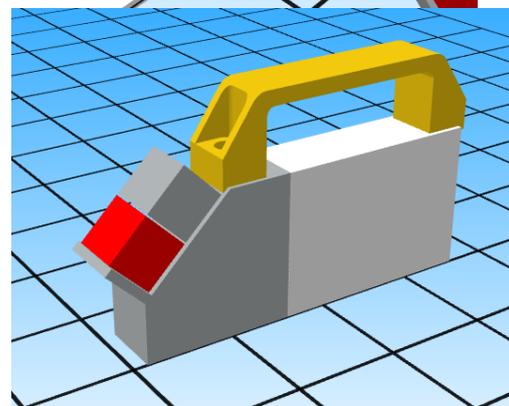
	schaltet die Ansicht des Roboter-Arbeitsraums (Working Space) ein oder aus
	schaltet das 3D-Modell des Standardgreifers ein oder aus
	enthält die Liste der Objekte der Klassen: Tool, Obstacle oder Work
	macht die stl-Datei auf und liest sie als eine der Objekte der Klassen: Tool, Obstacle oder Work ein
	schaltet das Menü der Modifikation von Objekten ein, erlaubt die Änderung der Objektposition oder den Wechsel seiner Farbe
	löscht das aktuell gekennzeichnete Objekt auf der Dropdown-Liste
	schaltet das Menü des Generators von einfachen 3D-Formen ein
	schaltet das Generieren der Visualisierungen von Robotertrajektorien ein
	schaltet die Visualisierung von Robotertrajektorien aus und reinigt diese
	speichert in die .traj-Datei Punkte der Visualisierungen von Robotertrajektorien ein
	schaltet das Menü der Einstellungen von Visualisierungsfenstern ein

15.12.2 Objektklassen

Work – die Objekte der Work-Klasse können durch den Roboter übertragen werden. Damit das Objekt aufgegriffen werden kann, muss sich der TCP-Punkt innerhalb des aufzugreifenden Objektes befinden und das Steuersignal muss im hohen Zustand sein.

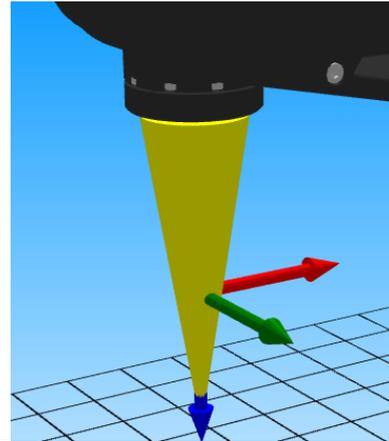


Obstacle – Objekte der Obstacle-Klasse sind statische Bestandteile der Visualisierung. Sie erlauben die Visualisierungsszene aufzubauen, sind ein visueller Aspekt und potentielle Hindernisse anzuzeigen.



ASTORINO Betriebsanleitung

Tool – Objekte der Tool-Klasse sind Objekte, die am Roboterflansch dauerhaft befestigt sind. Durch diese Objekte kann man eigene Werkzeuge schaffen, die am Roboterflansch montiert sind.



15.12.3 Generator einfacher Formen

Der Generator einfacher dreidimensionalen Formen erlaubt folgende Elemente zu generieren:

- Kubus (Würfel),
- Quader,
- Kegel,
- Zylinder,
- Sphäre,
- Pyramide,
- Rohr

Objektname, wenn dieser nicht eingetragen ist, wird dem Objekt ein weiterer freier Name automatisch zugeschrieben.

Objektgröße, abhängig von der Figur sind von 1 bis 3 Parameter anzugeben.

Position, unter welcher die Figur geschaffen werden soll.

Auswahl der Klasse vom zu schaffenden Objekt: Tool, Obstacle, Work.

Hinzufügung des Objektes zur Visualisierung

Auswahl der Objektfarbe

Generatormenü schließen

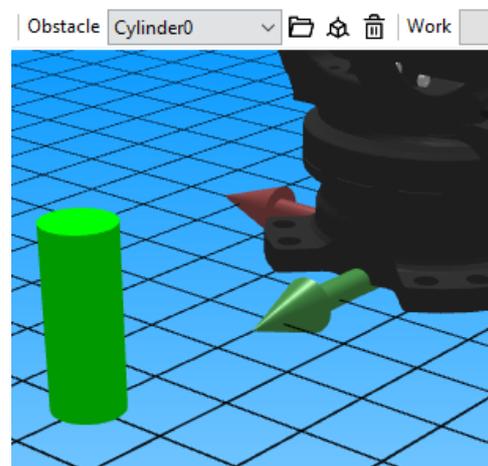
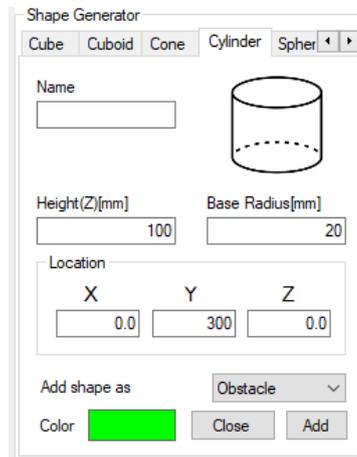
ASTORINO Betriebsanleitung

Beispiel

Der Visualisierung ist ein Zylinder der Obstacle-Klasse mit den Parametern hinzuzufügen:

- 100 mm hoch,
- Basisradius 20mm,
- Farbe grün
- Anfangsposition (0,300,0 [x,y,z])
- beliebiger Name

Damit ein solches Objekt hinzugefügt werden kann, sind folgende Daten im Generatormenü hinzuzufügen und mit der [ADD]-Taste zu bestätigen. Das Objekt wird hinzugefügt.



15.12.4 Menü der Objektmodifikation

Move - Cylinder0 ← Name der aktuell modifizierten Figur.

X: 0.00 Y: 300.00 Z: 0.00
 ← Schieber und Textfelder, die die Position des Objektes auf der Visualisierung ändern lassen.

Rx: 0.00 Ry: 0.00 Rz: 0.00
 ← Einstellung der Auflösung der vorgenannten Schieber

Close Apply
 ← Schließung des Modifikationsfensters.

Increment
 Linear 0.5
 Rotate 0.5
 ← Bestätigung der Änderungen der Objektpositionen.

Color
 ← Bereich der schnellen Farbwahl, das Anklicken mit der Maustaste auf die konkrete Farbe erlaubt diese als Farbe des modifizierten Objektes zu wählen

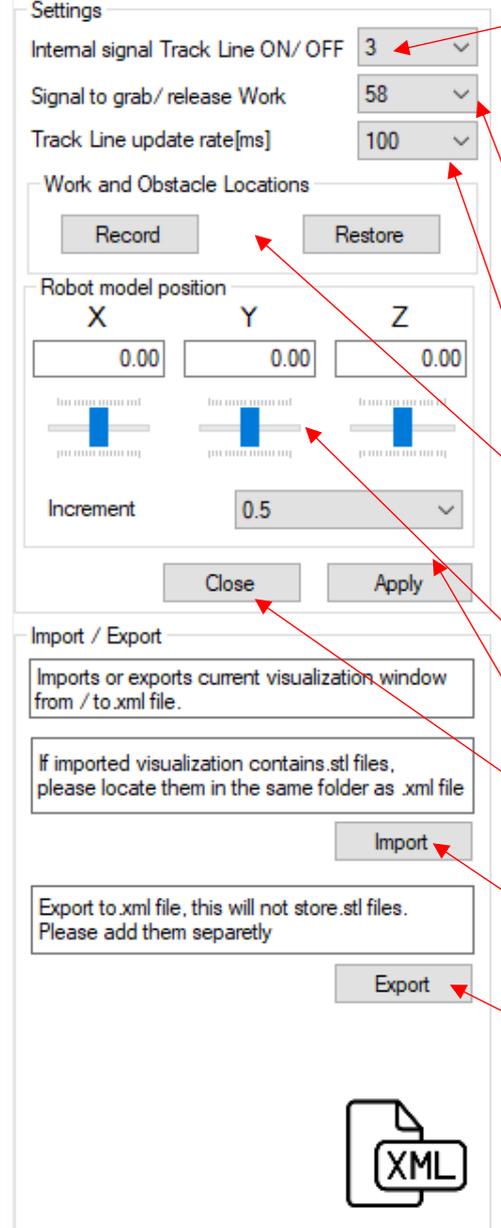
Color Preview
 ← Ansichtsfenster der gewählten Farbe.

Opacity 100.0 % ← Einstellung der Undurchsichtigkeit des Objektes.

Selected Color
 ← Die aktuell gewählte Farbe, das Anklicken dieses Bereiches macht das Fenster der fortgeschrittenen Farbwahl auf

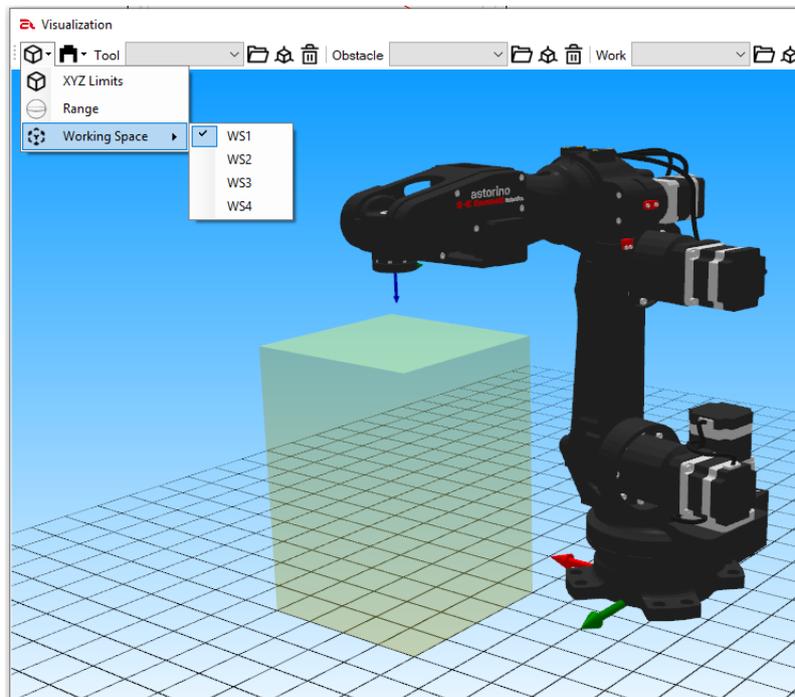
Apply
 ← Bestätigung der Änderungen.

15.12.5 Menü Visualisierungseinstellungen

	<p>Internes Signal (aus dem Pool 2001-2016), welches das Generieren der Trajektorie-Punkte auf der Visualisierung ein- oder abschaltet.</p> <p>Eingangssignal, welches das Greifen von Objekten der Work-Klasse in der Visualisierung durch den Roboter steuert.</p> <p>Zeit, in welcher ein weiterer Punkt der Trajektorievizualisierung geschaffen wird.</p> <p>Speicherung und Wiederherstellung der Positionen von Elementen, die sich auf der Visualisierung befinden.</p> <p>Änderung der Roboterposition auf der Visualisierung.</p> <p>Bestätigung der Änderungen.</p> <p>Schließung der Menü Einstellungen.</p> <p>Import der Visualisierung aus der xml-Datei.</p> <p>Export der Visualisierung in die xml-Datei. Der Export speichert die wiederhergestellten stl-Dateien nicht, sondern nur ihre Namen. Die Dateien sind separat zu kopieren.</p>
--	---

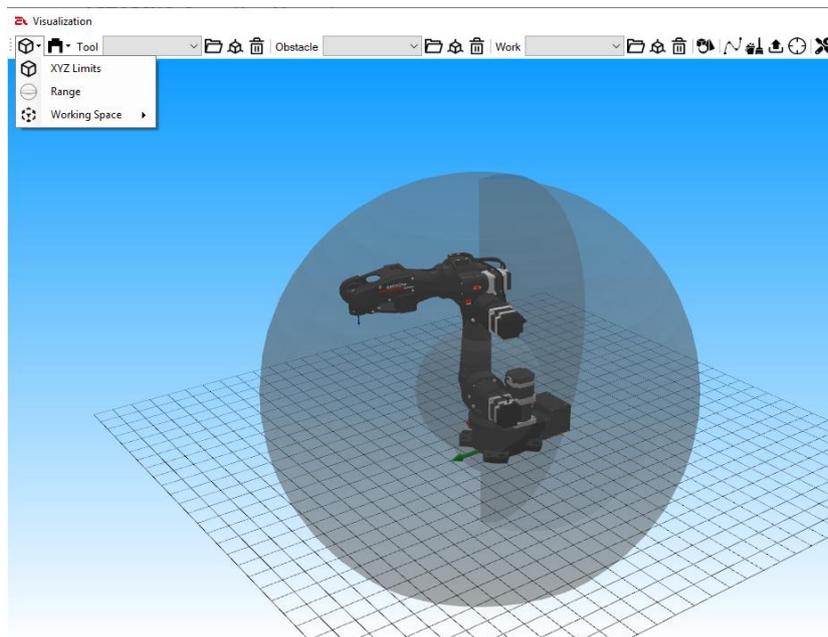
15.12.6 Working Space-Visualisierung

Das Visualisierungsfenster erlaubt alle vier Arbeitsräume anzuzeigen. Wählen Sie, welche von ihnen der Szene hinzugefügt werden soll und demnächst [WS1.. 4] klicken. Ein erneuter Klick hat zur Folge, dass das 3D-Modell des Arbeitsraums versteckt wird.



15.12.7 Visualisierung der Arbeitsreichweite des Roboters

Das Visualisierungsfenster erlaubt die Arbeitsreichweite des Roboters anzuzeigen. Die [Range]- Taste klicken, um ein 3D-Modell des Arbeitsbereiches hinzuzufügen oder zu entfernen.



15.13 Karte About / Informationen

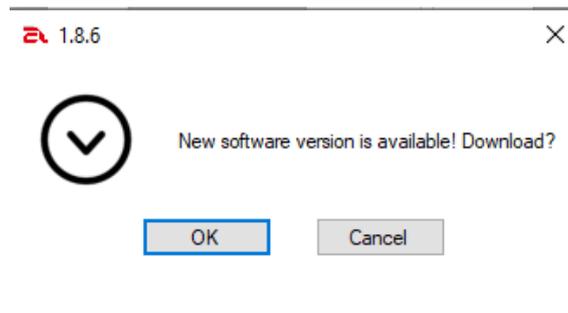


Dieser Reiter zeigt die aktuelle Version der Astorino-Software und der kompatiblen Firmware mit der aktuellen Astorino- Software an.

15.14 Aktualisierung der Firmware

15.14.1 Grundlegende Informationen

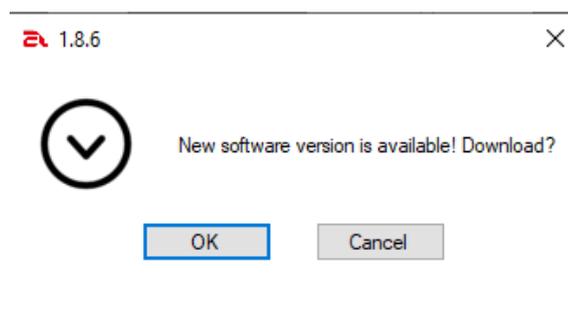
Die Astorino-Software prüft nach Inbetriebnahme automatisch, ob eine neue Version verfügbar ist und wenn ja, teilt sie dies dem Nutzer mit.



Wird die [OK]-Taste geklickt, hat dies zur Folge, dass eine neue Version zu der vom Nutzer festgelegten Lokalisierung auf die Festplatte heruntergeladen wird.

Demnächst hat der Nutzer die alte Version am Computer zu deinstallieren und die neue zu installieren.

Nach der Verbindung mit dem Roboter prüft die Astorino-Software, ob die Software des Roboters aktuell ist. Wenn die neue Firmware verfügbar ist, teilt dies die Anwendung dem Nutzer mit.



Das Anklicken der [OK]-Taste bewirkt das Herunterladen auf die neue vom Nutzer festgelegte Lokalisierung auf der Festplatte.

Hier können Sie die neueste Software-Version vom FTP-Server Kawasaki Robotics herunterladen: <https://ftp.kawasakiRoboter.de/Software/Astorino/>

oder die technische Hilfe kontaktieren:
Astorino@astor.com.pl / Tech-Support@kawasakiRoboter.de

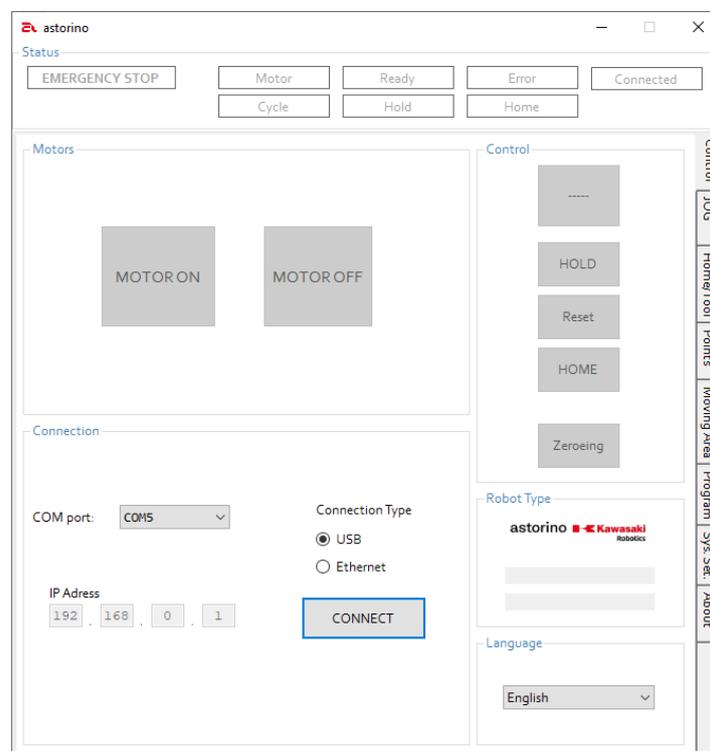
15.14.2 Aktualisierungsverfahren

Um die Firmware zu aktualisieren, ist die Astorino-Software in Gang zu setzen.

[VORSICHT]

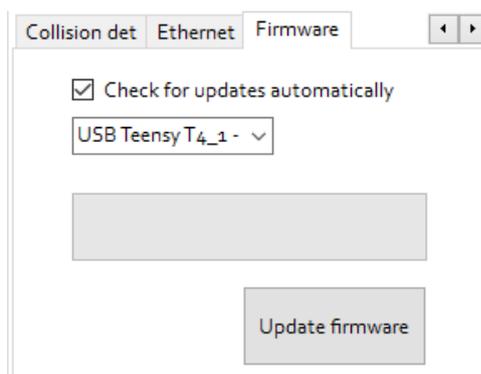
Vergewissern Sie sich, dass die Software an den Roboter **nicht** angeschlossen ist. Die Motoren müssen abgeschaltet sein!

Die Unterbrechung des Vorgangs kann den Prozessor beschädigen, den Roboter während des Aktualisierungsvorgangs nicht abschalten!



Zu [System Setting] wechseln und dort zum Bereich IO-Konfiguration.

Das Pfeil-Symbol nach rechts klicken, bis der Unterreiter [Firmware] erscheint.

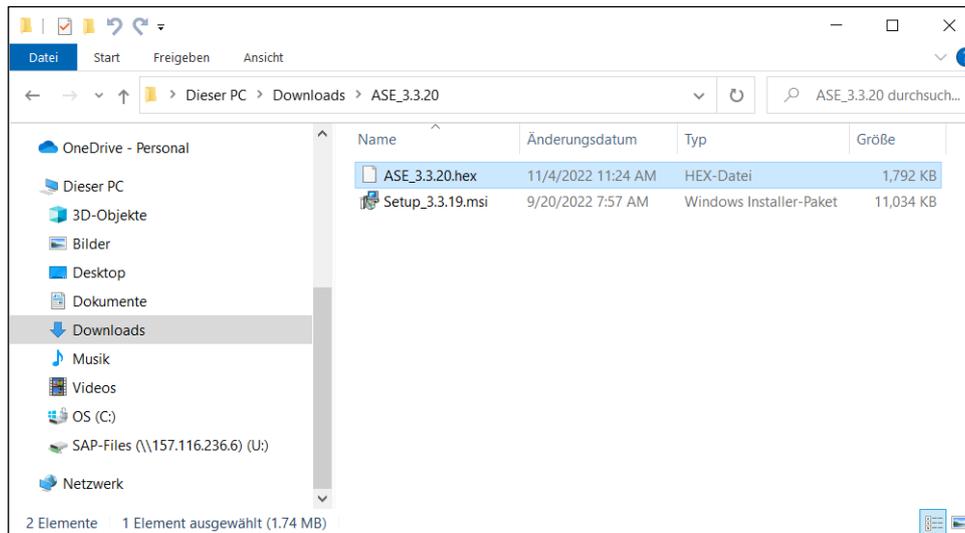


Die Taste [Update firmware] klicken, um das Fenster Dateiauswahl aufzumachen.

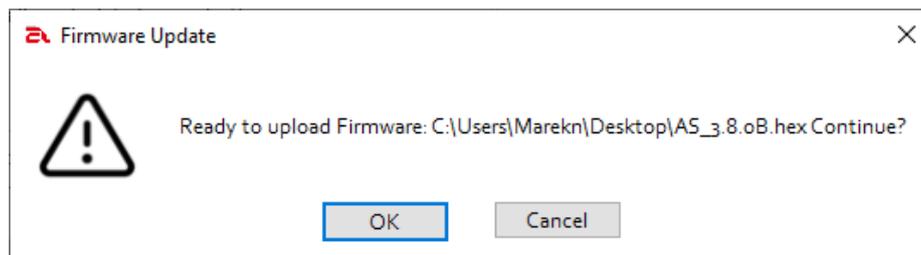
Die Datei *.hex, welche die neue Software enthält, auswählen.

ASTORINO Betriebsanleitung

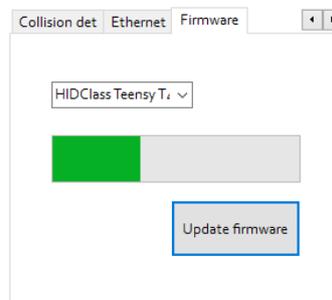
Dateiauswahlfenster:



Bestätigen, dass Sie die neue Software in den Roboterspeicher übersenden wollen:

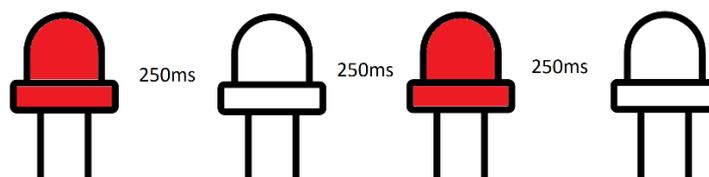


Es wird eine Aktualisierung der Firmware durchgeführt.



Nachdem die Firmware installiert wurde, beobachten Sie die rote LED-Diode (Error) am Robotersockel.

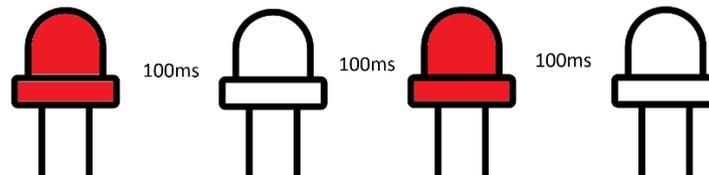
Beginnt die Diode langsam zu blinken (etwa zweimal pro Sekunde), ist der



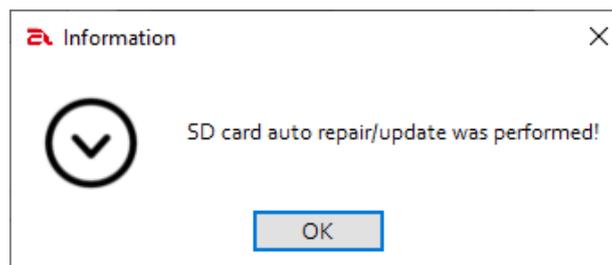
ASTORINO Betriebsanleitung

Roboter abzuschalten und wieder einzuschalten. Das ist ein Fehler der SD-Karte im Robotersockel, der Prozessor konnte die Karte nach der Aktualisierung der Firmware nicht wieder starten. Das Speisung-Reset löst das Problem.

Beginnt die LED-Diode schnell zu blinken (etwa fünfmal pro Sekunde),



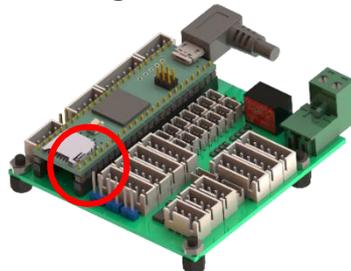
bedeutet das, dass der Roboter die Daten auf der SD-Karte aktualisieren muss. Der Vorgang wird selbsttätig durchgeführt. Nach dem abgeschlossenen Vorgang schaltet die rote LED-Diode ab und nach dem Anschließen an Computer erscheint die Mitteilung, dass der Datenreparatur-/Datenaktualisierungsvorgang auf der SD-Karte abgeschlossen ist.



15.14.3 Datenwiederherstellung nach einer Aktualisierungsstörung

Wurde der Prozess unterbrochen, ist es möglich, dass der Prozessor nicht funktioniert und das Windows-System den Roboter nicht erkennt. Sie können den Prozessor zu den Werkeinstellungen resetten, indem die weiße Reset-Taste am Prozessor **13s bis 17s** lang gedrückt gehalten wird und demnächst beginnt die rote LED-Diode an der Prozesseinheit zu blinken.

Nachdem der Hard Reset-Vorgang durchgeführt worden ist, wird die orangene LED-Diode am Prozessor langsam blinken und die rote LED-Diode (Error) am Robotersockel beginnt ebenfalls langsam zu blinken.



Die weiße Taste befindet sich auf der Prozessorplatte innerhalb des Robotersockels. Um auf sie Zugriff zu erhalten, ist der hintere Oberdeckel abzuschrauben und abzunehmen.

ASTORINO Betriebsanleitung



Seien Sie dabei sehr vorsichtig, wenn Sie das machen! Verwenden Sie keine Metallgegenstände innerhalb des Robotersockels, wenn die Werkstellungen des Prozessors wiederhergestellt werden!

15.15 AS-Sprache

Astorino kann man unter Anwendung der Grundversion der Kawasaki AS-Sprache, die bei allen Industrierobotern von Kawasaki Robotics zum Einsatz kommt, programmieren. Um mehr über diese Sprache zu erfahren, setzen Sie sich bitte mit der Dokumentation „Anleitung der AS-Sprache“ in Kenntnis.

Die nachfolgende Anleitung beschreibt kurz die grundlegenden Funktionen, welche zur Programmierung des Roboterbetriebs angewendet werden.

Aktuelle Liste der unterstützten Befehle und Funktionen:

(x,y,z stehen für die Werte - z. B. SPEED 100 ALWAYSp für Punkte oder Namen der Punkte - z. B. JMOVE P10)

Name	Beschreibung
ACCEL x	Beschleunigung des Roboters in % für den Befehl nächste Bewegung
ACCEL x ALWAYS	Beschleunigung des Roboters in %
ALIGN	Ausgleichen der Z-Achse des Werkzeugs zu der nächsten BASE-Achse
C1MOVE p	legt den mittelbaren Punkt der Kreisinterpolation fest
C2MOVE p	verschiebt den Roboter zum Punkt p in der Kreisinterpolation, indem durch den im Befehl C1MOVE p bestimmten Punkt durchgegangen wird; vor dem C2MOVE-Befehl ist der Befehl C1MOVE anzuwenden
CVCOOPJT x	legt die Zusammenarbeit zwischen dem Förderband Nummer 1 oder 2 fest
CVDELAY x	der Roboter hält die aktuelle Position des Förderbands über die Zeit x aufrecht
CVLAPPRO p,x	bewegt sich in Richtung Z des Werkzeugs auf der jeweiligen Entfernung x vom Punkt p linear mit Verfolgung des Förderbands
CVLDEPART x	verschiebt den Roboter von der laufenden Position in der festgelegten Entfernung x von der laufenden Position entlang der Z-Achse des Werkzeugs mit Verfolgung des Förderbands

ASTORINO Betriebsanleitung

CVLMOVE p	lineare Bewegung zum Punkt p mit Verfolgung des Förderbands
CVRESET x	resettet die Daten des Förderbands zum x-Wert
CVWAIT x	wartet, bis das Förderband den x-Wert erreicht
DECEL x	Bremsen in % für den Befehl nächste Bewegung
DECEL x ALWAYS	Bremsen in %
DISTANCE(p,p)	berechnet die Entfernung zwischen zwei Punkten
DLYSIG x,y	aktiviert das Signal X (1–58 oder Int. 2001–2016) nach Ablauf der Zeit Y in Sekunden
DRIVE x,y,z	bewegt die einzelne Achse um x- Achse, y- Winkel, z -Geschwindigkeit
DRAW x,y,z	lineare Bewegung gegenüber x,y,z gemäß BASE
\$DECODE(x,y)	Die Funktion sucht das Trennzeichen y im Zuge x und sondert sämtliche Zeichen, die sich vor dem Trennzeichen befinden, aus. Diese Zeichen werden erneut als eine Zeichenkette ausgeführt und gleichzeitig aus der originalen Zeichenkette entfernt!
\$ENCODE(x)	wechselt die Zahl in eine Zeichenkette
ERESET	Fehler-Reset
EXISTCOM	Bereitschaftszustand der Kommunikationsdaten HOST (Serial)
HERE p	aktuelle Position des Roboters zum Punkt P speichern
HOME	verschiebt den Roboter zur HOME-Position
INRANGE(p)	prüft, ob sich der Punkt in der Reichweite des Roboterarms befindet
JAPPRO p,x	bewegt sich Punkt-zu-Punkt in der Z-Richtung des Werkzeugs in einer gewissen Entfernung x vom Punkt p
JUMP p,x	Sonderbefehl: JUMP zur p-Position, wo x ein Handgelenk oder ein kartesischer Punkt ist, x entspricht der Hubhöhe.
JMOVE p	Bewegung des Roboters Punkt-zu-Punkt zur p-Position (joint), wo p ein Verbindungs- oder ein kartesischer Punkt ist
LMOVE p	lineare Bewegung zum p-Punkt

ASTORINO Betriebsanleitung

LAPPRO p,x	bewegt sich in Richtung Z des Werkzeugs für die jeweilige Entfernung x vom p-Punkt linear
LDEPART x	verschiebt den Roboter von der laufenden Position in bestimmter Entfernung x vom laufenden Verkaufspunkt entlang der Z-Achse des Werkzeugs
POINT p	bildet die Variable x des Punktes
PRINT x	Eintragung der Daten/des Textes am Terminal
PULSE x,y	aktiviert das Signal X (1-58 oder Int. 2001-2016) für die Zeit Y (Sek.)
SEND x	Übersenden der Daten an HOST (Serial)
SHIFT(p BY x,y,z)	bildet einen neuen Punkt aufgrund der Verlagerung von p Beispiel: POINT TST = SHIFT(P1 DURCH 10,0,0)
SIG(x)	prüft den Signalzustand x — gibt den Wert TRUE oder FALSE zurück Beispiel: IF SIG(2001) == TRUE THEN
SIGNAL x	aktiviert das Signal X (1-58 oder Int. 2001-2016)
SIGNAL -x	deaktiviert das Signal X (1-58 oder Int. 2001-2016)
SPEED x	die Roboter-Geschwindigkeit in % für den Befehl nächste Bewegung
SPEED x ALWAYS	Roboter-Geschwindigkeit in %
SPEED x MM/S	Roboter-Geschwindigkeit in mm/s (max. 250 mm/s) für den Befehl nächste Bewegung
SPEED x MM/S ALWAYS	Roboter-Geschwindigkeit in mm/s (max. 250 mm/s)
SWAIT x	stellt das Programm bis zum hohen Niveau des Signals X (1-58 oder 2001-2016) ein
SWAIT -x	stellt das Programm bis zum tiefen Niveau des Signals X (1-8 oder 2001-2016) ein
TDRAW x,y,z	lineare Bewegung gegenüber x,y,z gemäß TOOL
TOOL p	Auswahl der Werkzeugdaten aus Umwandlungen der p-Punkte
TOOL x	Auswahl eines der TOOL-Systeme (x = 1,2,3)
TWAIT x	stellt das Programm für x Sekunden ein

ASTORINO Betriebsanleitung

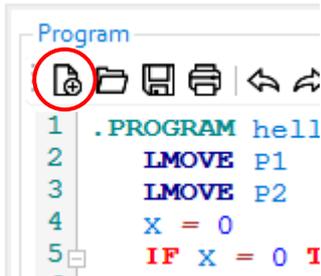
TYPE x	Aufführen der Daten/ des Textes am Terminal
X = CVPOS	Ablesung der Daten des Förderbands 1 an die Variable x
X = CVPOS2	Ablesung der Daten des Förderbands 2 an die Variable X
\$X = RECEIVE	Ablesung der HOST-Daten (Serial) vom Puffer an die Variable \$X
Y = VAL(\$X)	wechselt den Kettenwert in eine Zahl um

- Bedingte Anweisungen:
 - IF THEN ... ELSE ... END
 - IF THEN END
 - CASE ... OF .. VALUE ... ANY... END
 -
- Schleifen:
 - FOR ... TO ... END
 - DO... UNTIL
 - WHILE ... END
- mathematische Ausdrücke und Funktionen:
 - +, -, *, /, ^, MOD
 - SIN, COS, ATAN, FABS

15.16 Programmieren

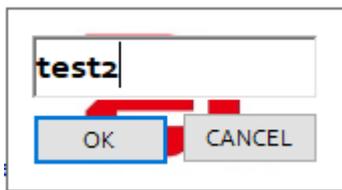
15.16.1 Erstellung eines neuen Programms

Der Bereich **Program** befindet sich auf der Karte **Programs**.
Ikone links klicken, um ein neues Programm zu erstellen:

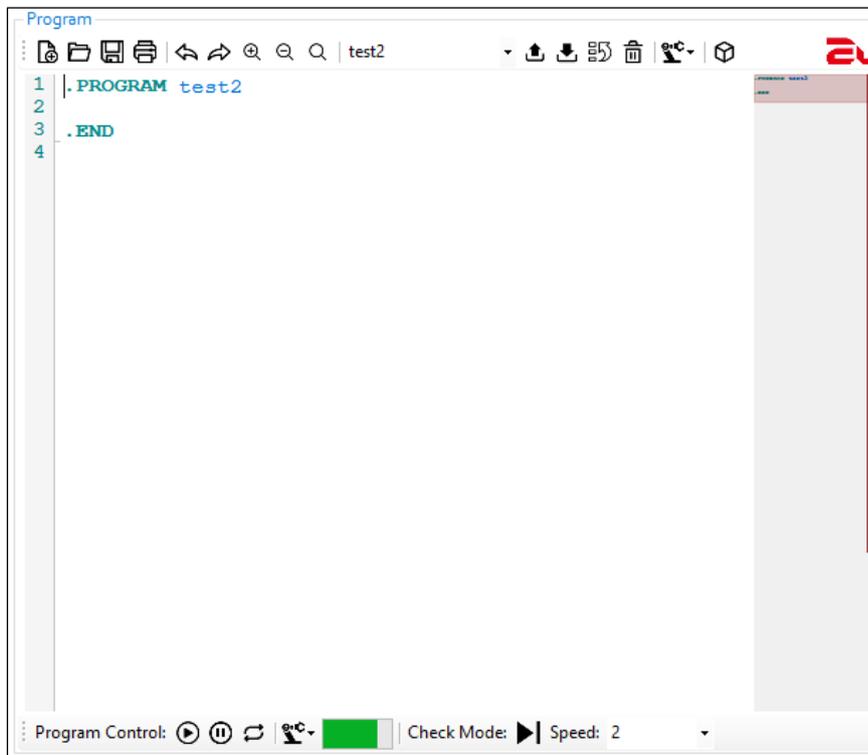


Ein neues Fenster erscheint.

Den Namen des Programms eingeben und **[OK]** klicken:



Die Software bildet eine neue Programmschablone:



15.16.2 Eigenes Programm schreiben

```

Program
┌──────────┴──────────┐
| 1 .PROGRAM test2      |
| 2   SPEED 60          |
| 3   SPEED 100 MM/S   |
| 4   i = 0             |
| 5   n = 5             |
| 6   FOR i = 0 TO n    |
| 7     POINT tst = SHIFT(P1 BY 10*i,0,0) |
| 8     LAPPRO tst, 50  |
| 9     LMOVE tst       |
|10     TWAIT 1        |
|11   END              |
|12 .END               |
|13                    |
└──────────┬──────────┘

```

15.16.3 Programm beim Roboter laden

Die Taste **[Senden]** beginnt gelb zu blinken. Die Taste klicken, um das Programm in den Roboterspeicher einzuspielen:

```

Program
┌──────────┴──────────┐
| 1 .PROGRAM test2      |
| 2   SPEED 60          |
| 3   SPEED 100 MM/S   |
| 4   i = 0             |
| 5   n = 5             |
| 6   FOR i = 0 TO n    |
| 7     POINT tst = SHIFT(P1 BY 10*i,0,0) |
| 8     LAPPRO tst, 50  |
| 9     LMOVE tst       |
|10     TWAIT 1        |
└──────────┬──────────┘

```

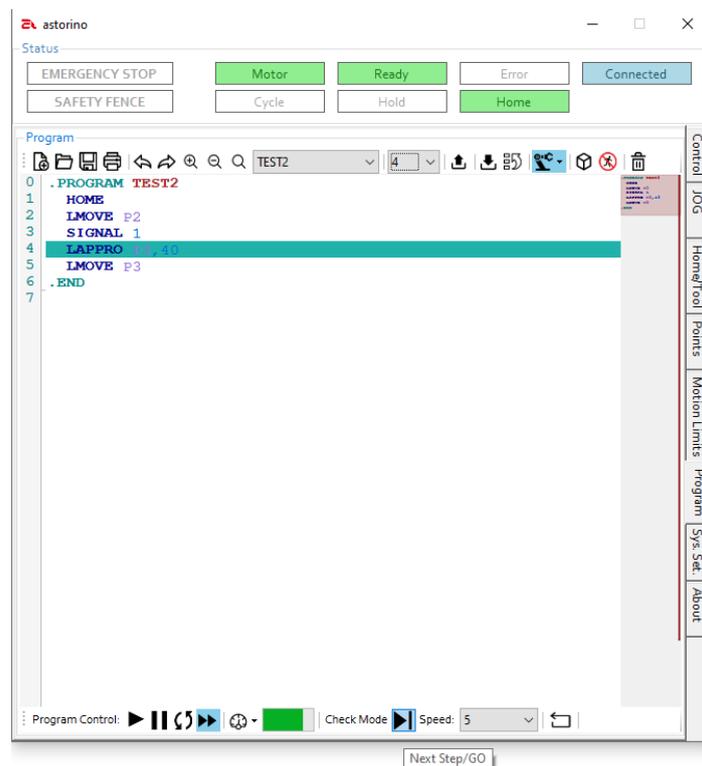


15.16.4 Eingangsetzen der aktuell ausgewählten Zeile

Um die aktuell markierte Linie in Gang zu setzen, ist die Taste Next Step/GO zu drücken und gedrückt zu halten sowie gleichzeitig die linke CTRL-Taste auf der Tastatur zu drücken und gedrückt zu halten. Der Roboter muss im In-Teach-Modus sein.



Wird die CTRL- oder GO-Taste losgelassen, wird der Roboter gestoppt.



WARNHINWEIS!

Die Inbetriebnahme des Codes mithilfe der GO-Taste bewirkt nur die Ausführung anderer Befehle als Steuerung mit dem Programmfluss. Solche Befehle wie IF, FOR, WHILE etc. liefern einen Fehler.

15.16.5 Programm aktivieren

! WARNHINWEIS!

Bevor das Programm gestartet wird, vergewissern Sie sich, dass der Roboter an keinen Gegenstand stößt. Haben Sie keine Sicherheit zu dem geschriebenen Programm, starten Sie es zuerst im DryRun-Modus!

```

0 .PROGRAM JMP
1 SPEED 250 MM/S ALWAYS
2 LMOVE P0
3 accuracy 100 ALWAYS
4 SIGNAL 2001
5

```

Ikone der Programmsteuerung [Cycle Start] ► „auf der Leiste wiedergeben“, um das Programm von der aktuell markierten Zeile in Gang zu setzen:

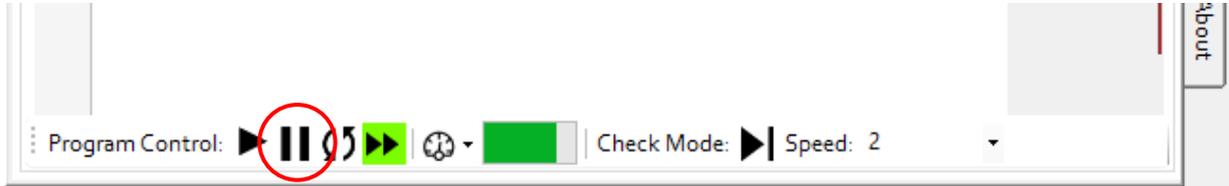


! WARNHINWEIS!

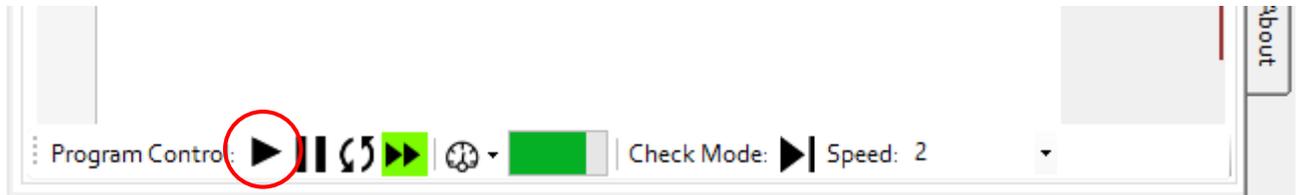
Das Ingangsetzen des Programms innerhalb der Schleife oder Bedingung (FOR, IF, WHILE etc.) liefert einen Programmfehler.

15.16.6 Programm anhalten

Um die Durchführung des Programms anzuhalten, ist zuerst der Roboterbetrieb mit der Taste [Einstellen/HOLD] einzustellen



Demnächst, nachdem der Roboter angehalten wurde, ist die Ikone der Programmsteuerung [Cycle Start] ► „auf der Leiste wiedergeben“ zu klicken, um das Programm einzustellen:



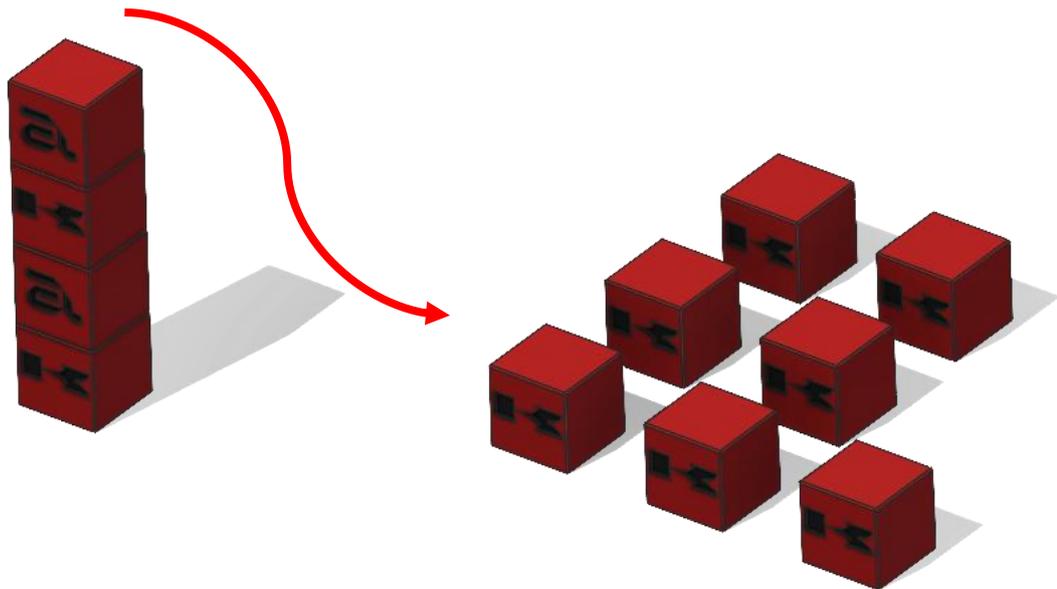
16 Beispielprogramme

16.1 Pick&Place – Palettierung-Beispiel

Dieses Programm bezieht Würfel aus einem Einzelturm und demnächst platziert es diese nach Anzahl der Reihen, der Spalten und der Ebenen.

Der Nutzer kann anpassen:

- die Größe des Gegenstands (des Würfels)
- den Abstand zwischen den Würfeln,
- die Anzahl der Reihen, der Spalten und der Ebenen,



ASTORINO Betriebsanleitung

```

. PROGRAM PAL1
;----- Init -----
deltaX = 60 ;distance between workpieces X
deltaY = 60 ;distance between workpieces Y
deltaZ = 30 ;layer height
numLev = 2
numRow = 1
numCol = 2
numPcs = numLev*numCol*numRow ;pieces count
height = 25 ;height of a workpiece (25 mm)
;----- variable init -----
x = 0
y = 0
z = 1
SIGNAL -1
speed 100 mm/s always
POINT place = p2
POINT pick = P1
POINT pick = SHIFT(p1 BY 0,0,numPcs*height)
;P1 on the table, pick shifted by number of pieces in Z
HOME
LAPPRO pick, 40
;----- Pal-----
FOR z = 0 TO (numLev-1)
    FOR y = 0 TO (numRow-1) ; rows in Y
        FOR x = 0 TO (numCol-1) ;col in X
            POINT pick = SHIFT(pick BY 0,0,-height);calc new pick pose
            JAPPRO pick,40
            speed 20 mm/s
            LMOVE pick
            TWAIT 0.5
            SIGNAL 1 ;close the gripper
            TWAIT 0.5
            LDEPART 50
            LMOVE P3
            POINT place = p2
            POINT place = SHIFT(p2 BY deltax*x,deltay*y,deltaz*z)
            LAPPRO place,30
            speed 20 mm/s
            LMOVE place
            TWAIT 0.5
            SIGNAL -1 ;open the gripper
            TWAIT 0.5
            LDEPART 30
            LMOVE P3
        END
    END
END
.END

```

16.2 Beispielprogramm Eingang/Ausgang

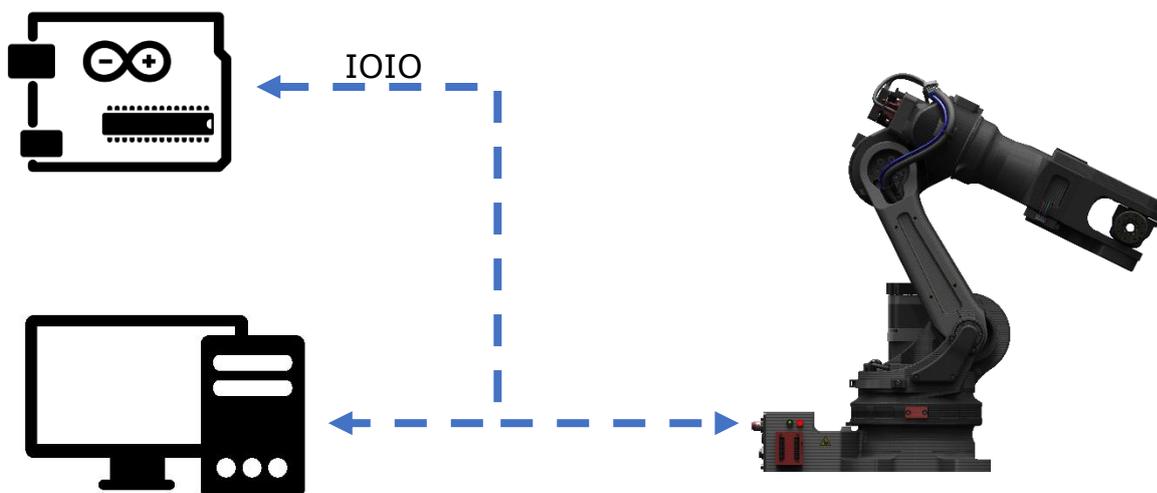
Dieses Beispielprogramm zeigt, wie die Signale auf vielen Wegen zu nutzen sind.

```

PROGRAM IO
; ----- IO example program
; ----- Roboter reads and sets IOs
sensor = 1002 ;sets variable
SWAIT 2001 ;wait until internal 1 signal is on
SIGNAL 8 ;sets 8 output HIGH
IF SIG(sensor) == TRUE THEN
    ;checks if sensor(2 input) is high
    SIGNAL 2002 ; sets 2 internal HIGH
ELSE
    IF SIG(1001) == FALSE THEN
        SIGNAL -8 ;sets 1 output LOW
    END
END
BITS 1,4 = 12
;changes 12 to 4bit binary and sets that on out puts from 1
data = BITS(1004,4) ;read binary data from inputs
;4 bit from 4th output and changes that to decimal
PRINT data
.END
    
```

16.3 Beispielprogramm für die serielle Datenübertragung

An diesem Beispiel wurde gezeigt, wie die serielle Datenübertragung zu nutzen ist. Das Programm kann die Daten zwischen dem Astorino-Roboter und einem PC-Computer (z. B. Matlab oder SerialTerminal) oder einem Mikrokontroller (z. B. Arduino oder ESP32) austauschen.



Plug-in M8 - (Pins: 1-GND, 2 - 5V [Vision], 3-TX, 4- RX)

ASTORINO Betriebsanleitung

```

.PROGRAM SERIAL
; ----- Serial communication example program
; ----- Roboter command frame form Serial Port
; ----- frames: P/ or L/x/y/z/
; ----- From X,Y,Z point is created
; ----- Sends current location if frame is P/
SPEED 150 MM/S ALWAYS
$$_FRAME = "XYZ"
$$_FRAME2 = "JT"
WHILE EXISTCOM == FALSE DO
    TWAIT 0.1
END
$TEMP = RECEIVE
$COMMAND = $DECODE ($TEMP, "/" )
PRINT $COMMAND
;RECEIVE DATA FROM SERIAL AND CREATE A POINT
IF $COMMAND == "L" THEN
    $VAL1 = $DECODE ($TEMP, "/" )
    $VAL2 = $DECODE ($TEMP, "/" )
    $VAL3 = $DECODE ($TEMP, "/" )
    DATAX = VAL ($VAL1)
    DATAY = VAL ($VAL2)
    DATAZ = VAL ($VAL3)
    POINT TEST = TRANS (DATAX, DATAY, DATAZ, 0, 0, 0)
    POINT/OAT TEST = P0
    LMOVE TEST
    SEND "OK"
END
;SEND CURRENT LOCATION TO SERIAL PORT
IF $COMMAND == "P" THEN
    HERE TEMP
    HERE #TEMP
    DECOMPOSE TAB[0] = TEMP
    DECOMPOSE TAB2[0] = #TEMP
    FOR I = 0 TO 5
        TAB2[I] = TAB2[I]*180/PI
        $$_FRAME = $$_FRAME + $ENCODE (TAB[I]) + "/"
        $$_FRAME2 = $$_FRAME2 + $ENCODE (TAB2[I]) + "/"
    END
    SEND $$_FRAME
    SEND $$_FRAME2
END
.END
    
```

WARNHINWEIS

Die serielle Datenübertragung funktioniert mit der Spannung 3.3V, bitte kompatible Elektronik mit der Spannung 3.3V oder Spannungswandler einsetzen.

Die Spannung 5V kann das CPU-Hauptsystem beschädigen!

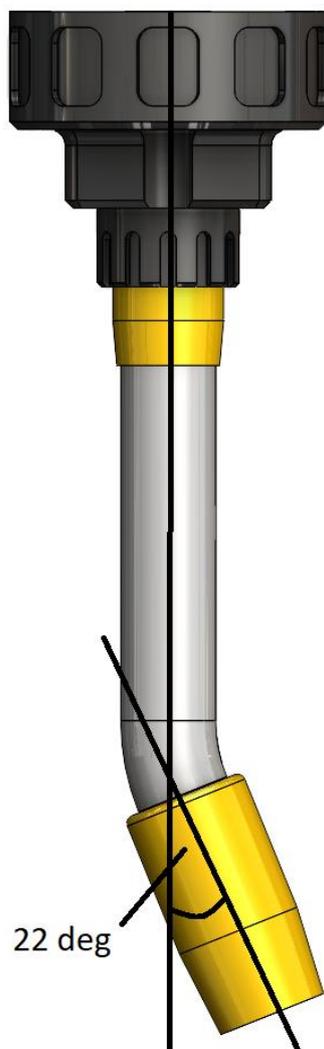
17 Daten des Werkzeugs

17.1 Daten des Werkzeugs von bekannten Abmessungen

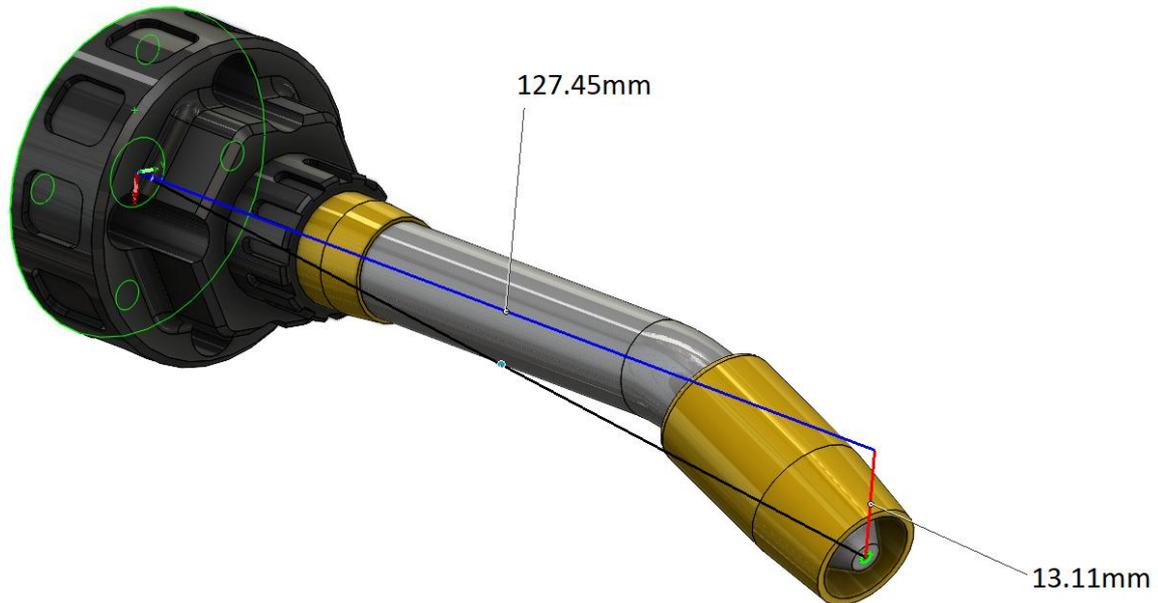
Die Abmessungen des Werkzeugs (TOOL DATA) werden im Grundkoordinatensystem (BASE) des Roboters angegeben.

Abmessungen des Werkzeugs aus bestehenden CAD-Daten eingeben oder diese selbständig messen.

Das nachstehende Beispiel eines Brenners zeigt, wie man kann selbständig solche Daten des Werkzeugs bestimmen kann:



Der Brenner ist unter einem Winkel von 22 Grad eingestellt.



In diesem Fall beträgt die Brennerlänge von der Oberfläche Flansch (Achse 6) bis zur Endung der Düse 127,45 mm in Richtung Z.
(Vorsicht! Nun arbeiten Sie mit Koordinaten des Werkzeugs)

Der Brenner verursacht die Verschiebung in Richtung Y um 13,11 mm. Der Bezugspunkt ist immer die Oberfläche des Werkzeugflansches. Das Koordinatensystem hat seinen Anfang in der Mitte der Flanschoberfläche, wo die Richtung Z = Richtung des Werkzeugdupels im Raum.

In diesem Beispiel sollen folgende Werte eingepflegt werden:

X [mm]	0. 0	
T [mm]	13. 11	
Z[mm]	127. 45	
Rx: -22,0	[O: 90. 0]	Rx ist die Drehung um die Achse X
Ry: 0.0	[O: 22. 0]	Ry ist die Drehung um die Achse Y
Rz: 0,0	[T: -90. 0]	Rz ist die Drehung um die Achse Z

17.2 Automatische Konfiguration des Werkzeugs (Koordinaten)

In diesem Kapitel wurden Verfahren der Funktion einer automatischen Registrierung von Koordinatenwerten des Werkzeugs beschrieben.

WARNHINWEIS

Die automatische Werkzeugkonfiguration ist eine Art vom Bewegungsvorgang des Roboters im TEACH-Modus. Seine Anwendung ist auf das Personal beschränkt, welches eine Sonderschulung abgeschlossen und Qualifikationen zum Einteachen oder zum Überwachen des Roboterbetriebs hat.

17.2.1 Übersicht der Funktionen der automatischen Werkzeugkonfiguration

Während der Bedienung des Roboters kann man am Handgelenkflansch verschiedene Werkzeuge mit verschiedenen Formen (Greifer usw.) montieren. Wenn die Daten des Werkzeugs nicht richtig gemessen wurden, kann sich die Trajektorie der Roboterbewegung von dem eingeteachten Pfad unterscheiden und sämtliche Fehler oder Störungen können sich vergrößern.

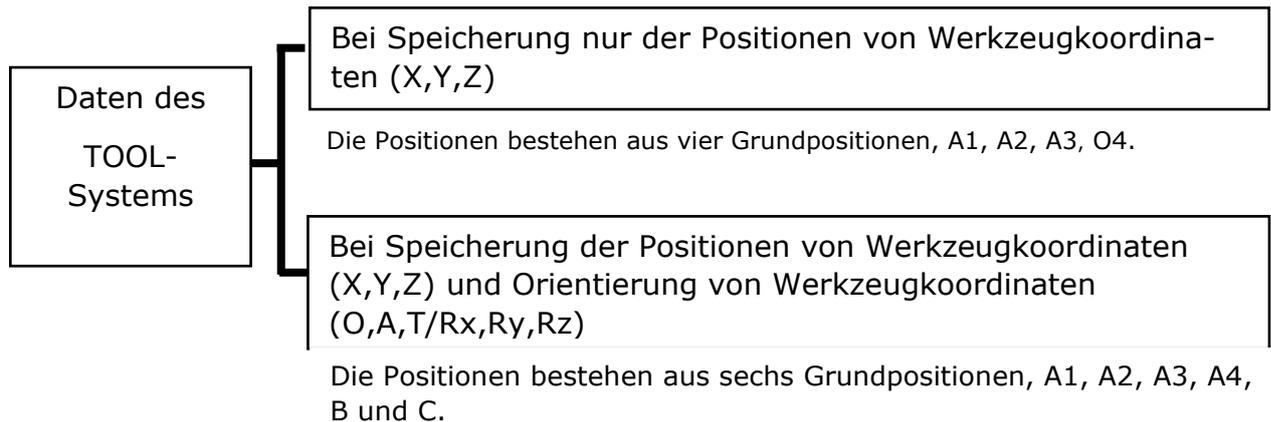
Mit anderen Worten sind die jeweiligen Werkzeuge für die richtige Roboterbedienung unerlässlich. Im Allgemeinen werden jeweilige Werkzeuge mithilfe von Zahlenwerten eingepflegt oder berechnet, es ist jedoch zu bedenken, dass die Messung der Lage und der Orientierung von Werkzeugdaten nicht genau sein kann oder sie kann zum Erwerb der Erfahrung beim Einteachen von Punkten lange brauchen.

Diese Funktion ermöglicht eine automatische Konfiguration der Werte von Werkzeugtransformation durch Einteachen von mehreren Punkten im Raum, ohne dass es erforderlich ist, die Werte der Werkzeugdaten mithilfe von numerischen Daten einzupflegen. Dazu braucht man eine Messvorrichtung mit spitzem Ende, zum Beispiel eine große Schafschraube und ein spitzer Kegel am Werkzeug.



17.2.2 Erforderliche Daten zur automatischen Konfiguration der Werkzeugkoordinaten

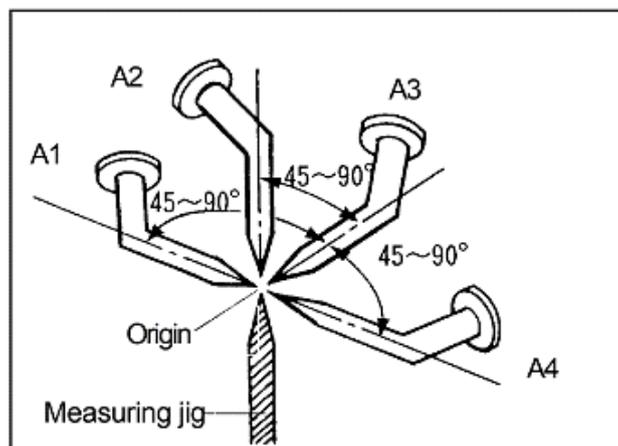
Bei Nutzung der Funktion der automatischen Konfiguration des Werkzeugs wird der folgende Datensatz der Verlegung gemäß dem Stand der Werkzeugdaten aufbewahrt. Die Messung der Daten einer Pose erfolgt durch das Zielen in einen Einteachpunkt von 4 oder 6 verschiedenen Positionen der Werkzeuge, wie es unten beschrieben wurde.



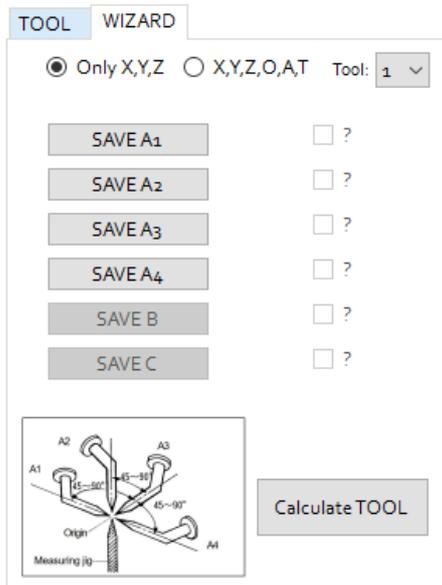
17.2.3 Einteachen von vier Grundpositionen

Die Vier-Punkte-Methode der TCP-Berechnungen ermöglicht Verschiebungen der TOOL-Daten in Dimensionen X, Y, Z zu finden.

Wie es auf nachfolgender Zeichnung gezeigt wurde, sollen Sie 4 Grundpositionen (A1, A2, A3, A4) mit denselben Daten der Positionen, aber mit verschiedenen Orientierungsdaten einteachen, indem es auf denselben Punkt des Endstücks am Messinstrument gezielt wird. Vergewissern Sie sich, dass die Winkel zwischen jeweiligen Orientierungen im Bereich von 45° bis 90° enthalten sind. Die Stirnfläche des Handgelenkflansches soll eine andere Ebene für jede Punktorientierung haben. Jede Basisposition ist so einzuteachen, damit sich die Koordinaten des Werkzeugs und der Anfang des Messinstrumentes miteinander berühren.



ASTORINO Betriebsanleitung

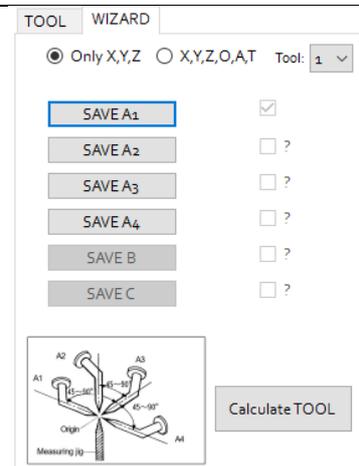


[Only X, Y,Z] im Reiter HOME/Tool auswählen

Aus der Liste ist die Nummer des Werkzeugs, das eingeteacht werden soll, auszuwählen. Sie können 1,2 oder 3 auswählen.

Den Roboter ins Einteach-Modus umschalten und zur Position wie unten übergehen (das ist ein Beispiel, die tatsächlichen Positionen können anders sein). Nachdem die Position erreicht wird, ist SAVE Ax zu drücken, wo x 1,2,3 oder 4 beträgt

A1 einteachen



ASTORINO Betriebsanleitung

A2 einteichen



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3 ?

SAVE A4 ?

SAVE B ?

SAVE C ?

Calculate TOOL

A3 einteichen



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3

SAVE A4 ?

SAVE B ?

SAVE C ?

Calculate TOOL

A4 einteichen



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3

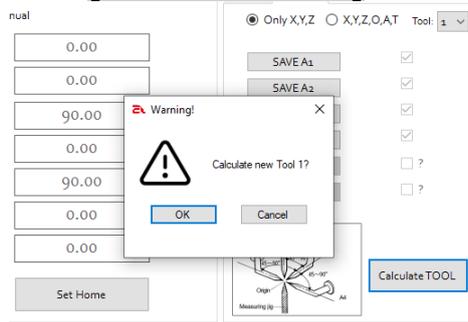
SAVE A4

SAVE B ?

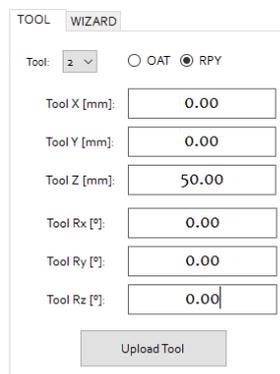
SAVE C ?

Calculate TOOL

Taste [Calculate TOOL] drücken

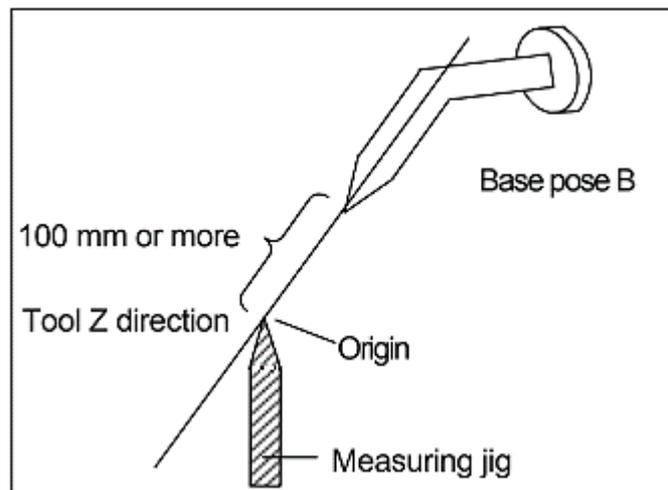


Die neuen Berechnungsdaten des Werkzeugs werden auf der SD-Karte gespeichert und auf der TOOL-Karte angezeigt.



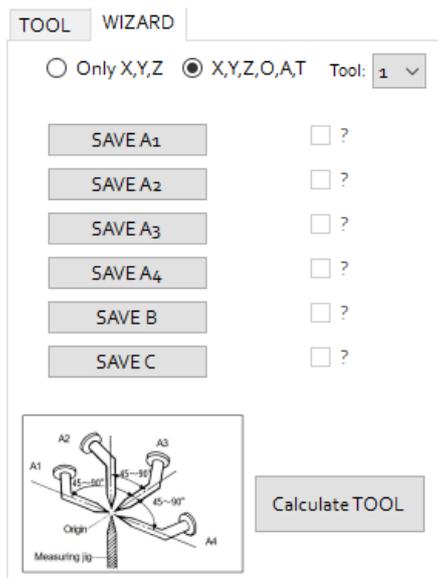
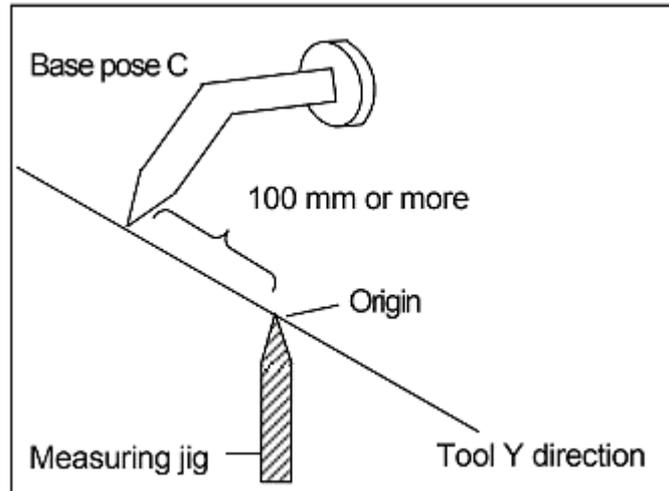
17.2.4 Einteachen von sechs Grundpositionen

Im Falle der Grundposition B soll diese so eingeteacht werden, dass sie sich zwischen dem Anfang des Messinstrumentes und der Position, die um 100 mm oder mehr von TCP (Mittelpunkt des Werkzeugs) entfernt ist, in erwünschter Richtung -Z des Werkzeugs, wie nachfolgend dargestellt wurde, berührt.



ASTORINO Betriebsanleitung

Im Falle der Grundposition C soll diese so eingeteacht werden, dass sie sich zwischen dem Anfang des Messinstrumentes und der Position, die um 100 mm oder mehr von TCP in erwünschter Richtung +Y des Werkzeugs entfernt ist, wie nachfolgend dargestellt wurde, berührt.



Im Reiter HOME/Tool (Werkzeug) ist **[X,Y,Z,O,A,T]** auszuwählen

Aus der Liste ist die Nummer des Werkzeugs, das eingeteacht werden soll, auszuwählen. Sie können 1,2 oder 3 auswählen

Den Roboter ins Einteacht-Modus umschalten und zur Position wie unten übergehen (das ist ein Beispiel, die tatsächlichen Positionen können anders sein). Nachdem die Position erreicht wurde, ist SAVE Ax zu drücken, wo x 1,2,3 oder 4 beträgt.

ASTORINO Betriebsanleitung

A1 einteichen

TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2 ?

SAVE A3 ?

SAVE A4 ?

SAVE B ?

SAVE C ?

Calculate TOOL

A2 einteichen

TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3 ?

SAVE A4 ?

SAVE B ?

SAVE C ?

Calculate TOOL

A3 einteichen

TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3

SAVE A4 ?

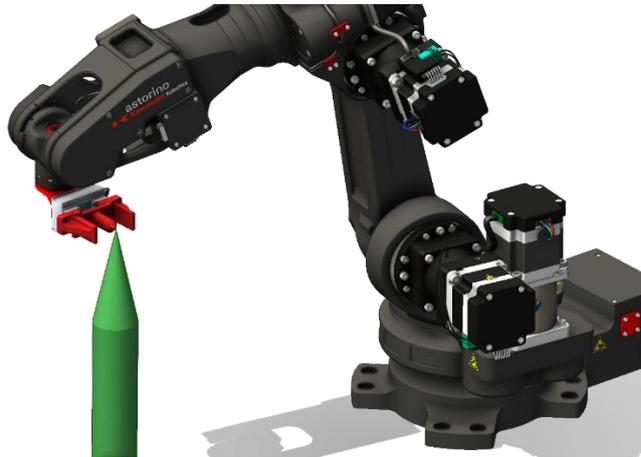
SAVE B ?

SAVE C ?

Calculate TOOL

ASTORINO Betriebsanleitung

A4 einteichen



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3

SAVE A4

SAVE B ?

SAVE C ?

Calculate TOOL

B einteichen



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3

SAVE A4

SAVE B

SAVE C ?

Calculate TOOL

C einteichen



TOOL WIZARD

Only X,Y,Z X,Y,Z,O,A,T Tool: 1

SAVE A1

SAVE A2

SAVE A3

SAVE A4

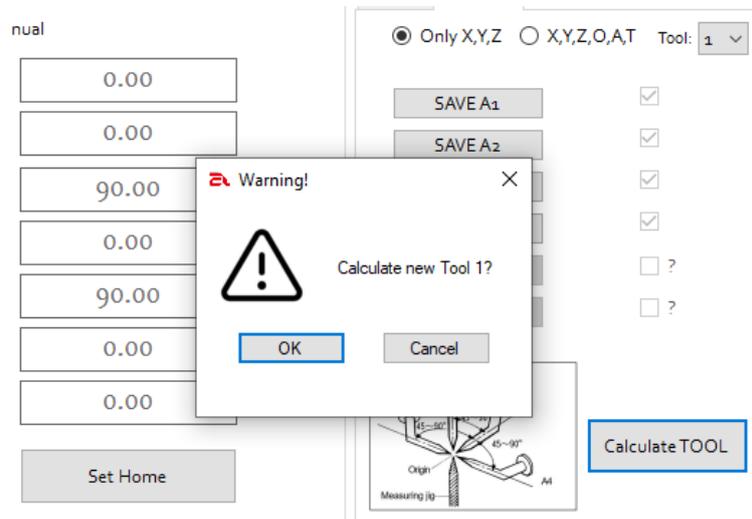
SAVE B

SAVE C

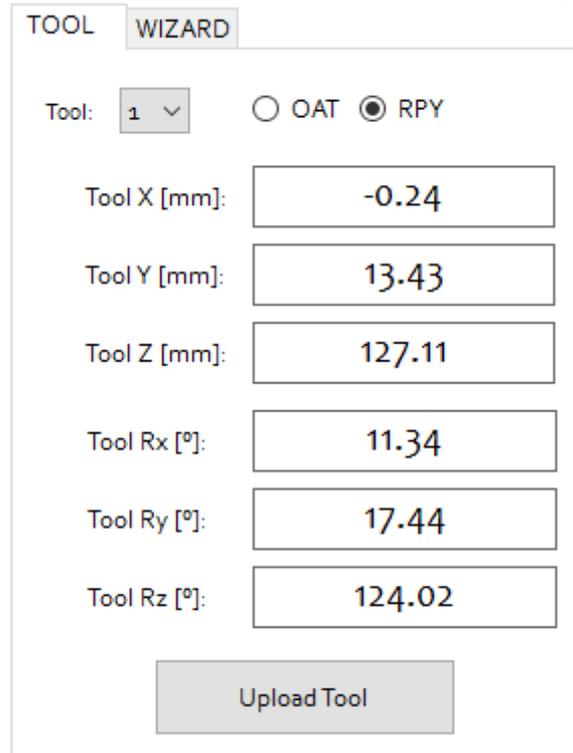
Calculate TOOL

ASTORINO Betriebsanleitung

Taste [WERKZEUG BERECHNEN] drücken



Die neuen Berechnungsdaten des Werkzeugs werden auf der SD-Karte gespeichert und auf der TOOL-Karte angezeigt.



18 Automatische Konfiguration des WORK-Systems (KOORDINATEN)

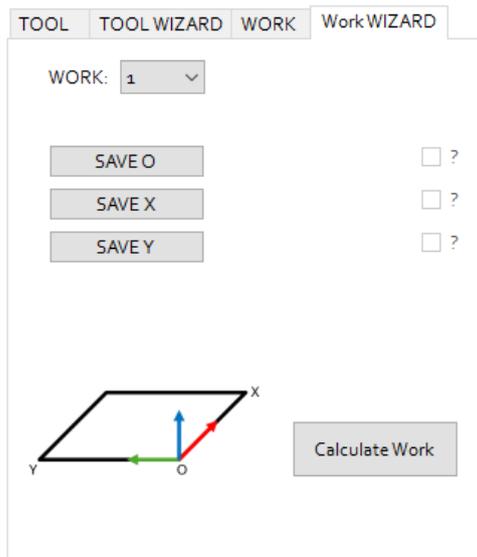
In diesem Kapitel wurden Verfahren der Funktion von automatischer Registrierung der Werte von Koordinaten des WORK-Systems beschrieben



WARNHINWEIS

Die automatische Konfiguration des WORK-Systems ist eine Art des Bewegungsvorgangs vom Roboter im TEACH-Modus. Seine Anwendung ist auf das Personal beschränkt, welches eine Sonderschulung abgeschlossen und Qualifikationen zum Einteachen oder zum Überwachen des Roboterbetriebs hat.

18.1.1 Einteachen von drei Grundpunkten

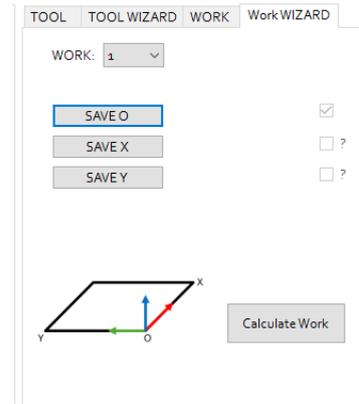
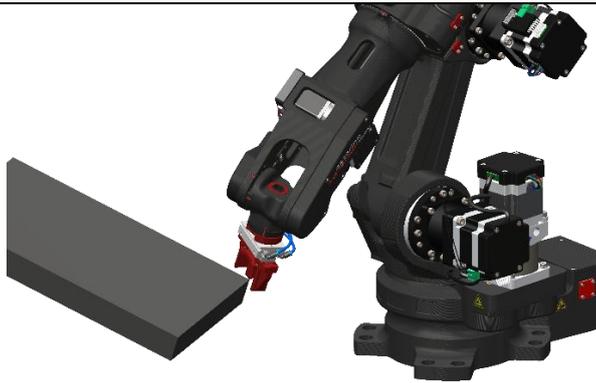


Nummer des WORK-Systems, welches berechnet wird, auswählen. 1 oder 2 kann ausgewählt werden.

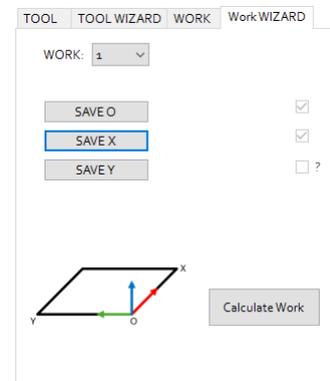
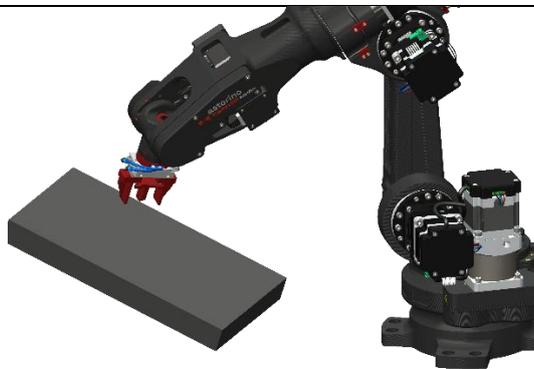
Den Roboter ins Einteach-Modus umschalten und zur Position wie unten übergehen (das ist ein Beispiel, die tatsächlichen Positionen können anders sein). Nachdem jede Position erreicht wurde, ist die Taste SPEICHERN O, X oder Y zu drücken. Der Punkt O definiert den Anfang des Koordinatensystems, der Punkt X definiert die Richtung der neuen Achse X, der Punkt Y definiert die Richtung der neuen Achse Y.

ASTORINO Betriebsanleitung

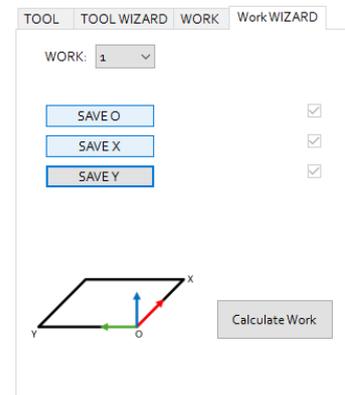
O einteichen.



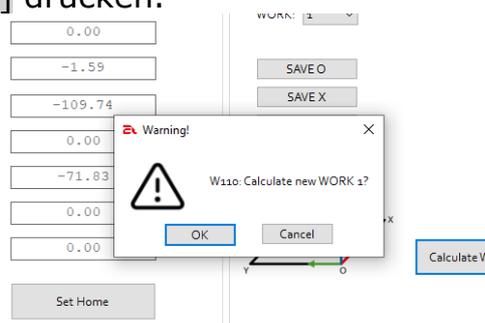
X einteichen.



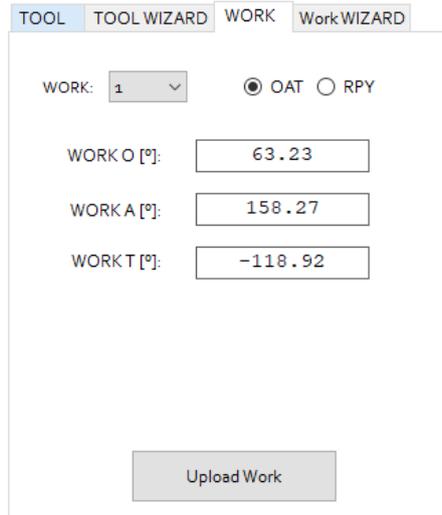
Y einteichen.



Taste [Calculate Work] drücken.



Die neu berechneten Daten des WORK-Systems werden auf der SD-Karte gespeichert und im WORK-Reiter angezeigt.



19 Autokalibrierung der Kollisionserkennung

[VORSICHT]

Die Kalibrierung wird für ein bestimmtes Programm ausgeführt, ein Programmwechsel kann die Wiederholung dieses Vorgangs erfordern!

Um die Schwellen der Erkennung von Kollisionen automatisch zu kalibrieren, gehen Sie zu Kollision über, Nutzer-Niveau zu 3 wechseln.

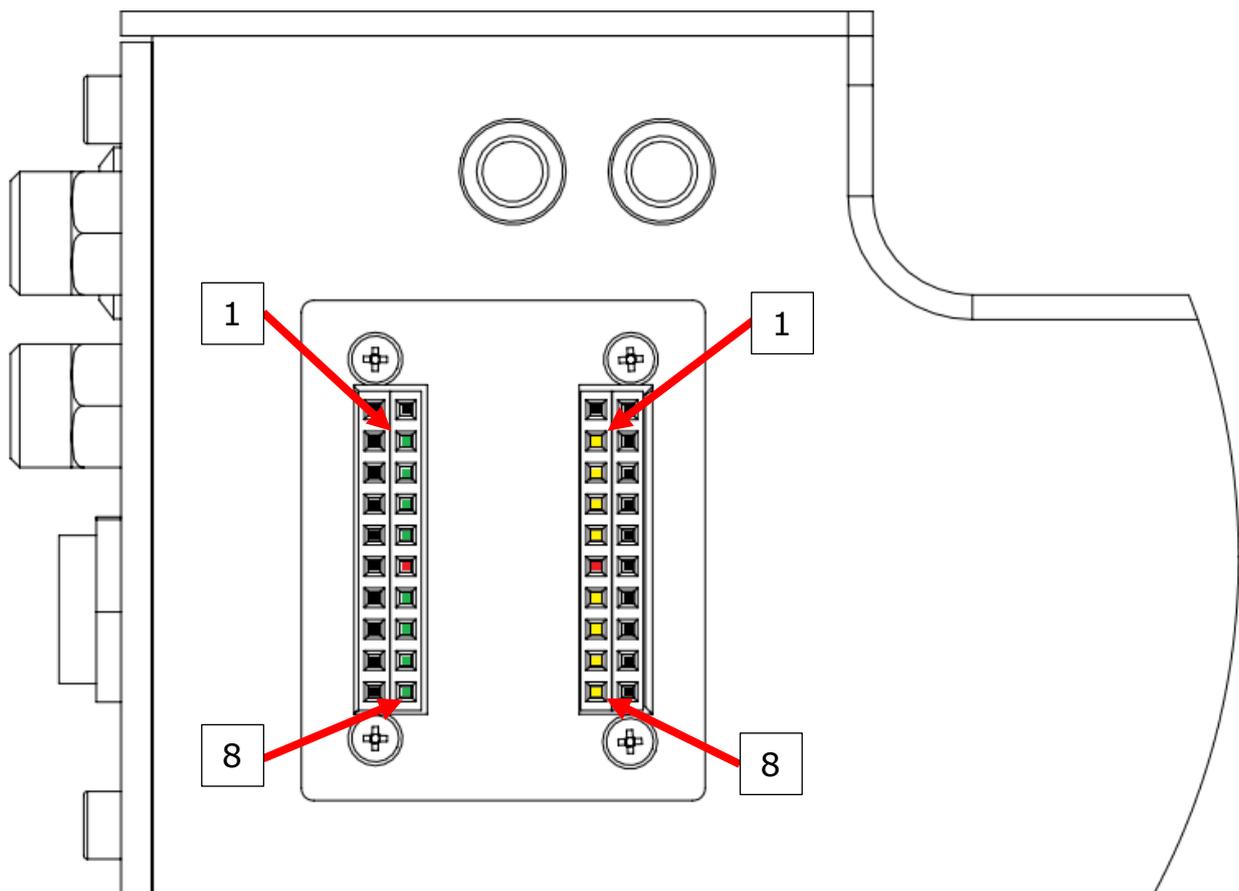
Verfahren Sie gemäß nachfolgenden Schritten:

- den Roboter in Teach- oder Repeat-Modus umschalten (die Kalibrierung wird für das zuletzt ausgewählte Modus ausgeführt),
- Programm auswählen, für welches Sie den Sensor kalibrieren wollen,
- zur Karte Collision detection übergehen und demnächst [Calibration ON/OFF] klicken
- das ausgewählte Programm für mehrere Zyklen in Gang setzen,
- das Programm anhalten (Cycle off),
- zum Reiter Collision detection übergehen und [Calibration ON/OFF] drücken,
- neue Schwellen werden im Roboterspeicher gespeichert.

20 Eingänge/Ausgänge – 3,3V

Astorino hat 8 Eingänge und 8 Ausgänge, die auf die Spannung 3,3V DC gestützt sind.

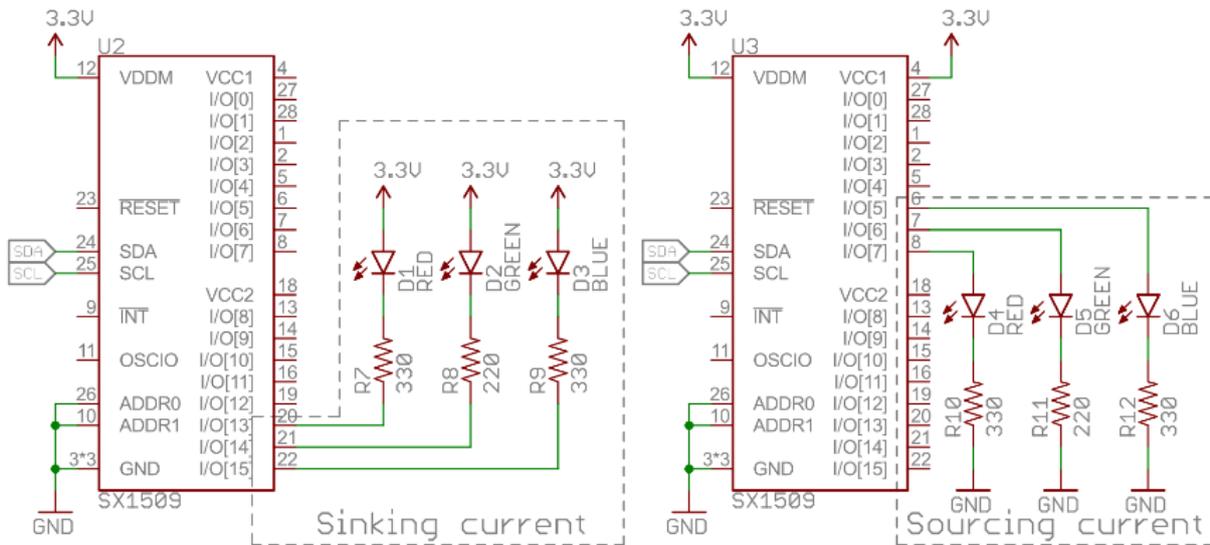
Farben	Funktion
■	Ausgang
■	Eingang
■	3,3 V/DC
■	0V (GND)



ASTORINO Betriebsanleitung

Das System arbeitet normalerweise als PNP-Schaltung. PNP bezeichnet positive Betriebsspannung (die vor allem in Europa und Nordamerika zum Einsatz kommt).

Im Zusammenhang damit schaltet das Modul das positive Potential auf seinen Ausgang um.



Den Vorgang kann man zu NPN wechseln, indem folgende Befehle im Terminal verwendet werden:

- Z_OUTSOURCE 1 – PNP
- Z_OUTSOURCE 0 – NPN
- Z_INPULL 1 – aktiviert das Anheben der Eingänge zu 3,3V
- Z_INPULL 0 - deaktiviert das Anheben der Eingänge zu 3,3V

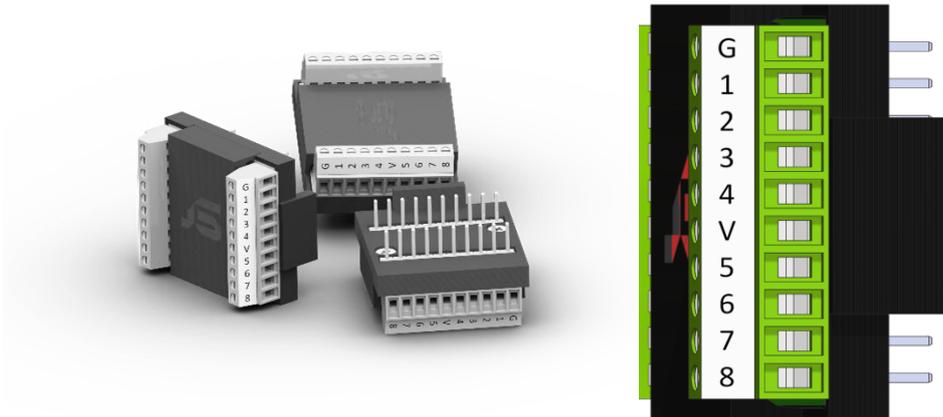


WARNHINWEIS!

Jeder AUSGANG liefert 8 mA Strom. Bitte das Limit nicht überschreiten, da die Hauptplatte dadurch beschädigt werden kann.

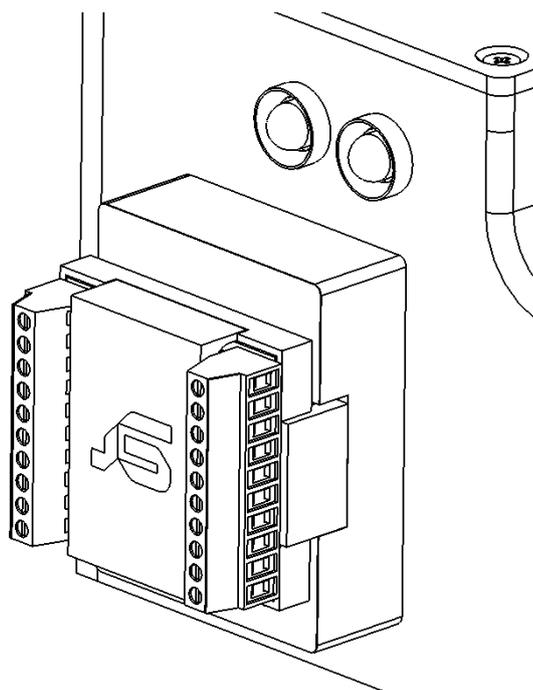
20.1 I/O 3.3V-Adapter

Der IO 3.3V-Adapter ist eine Option und wird gesondert verkauft. Der Adapter ermöglicht Außengeräte mit Verschraubungen leicht zu verbinden.



20.1.1 I/O 3.3V-Adapter – Montage

Der Adapter wird in die Standardeingänge/-ausgänge im Robotersockel eingesteckt.



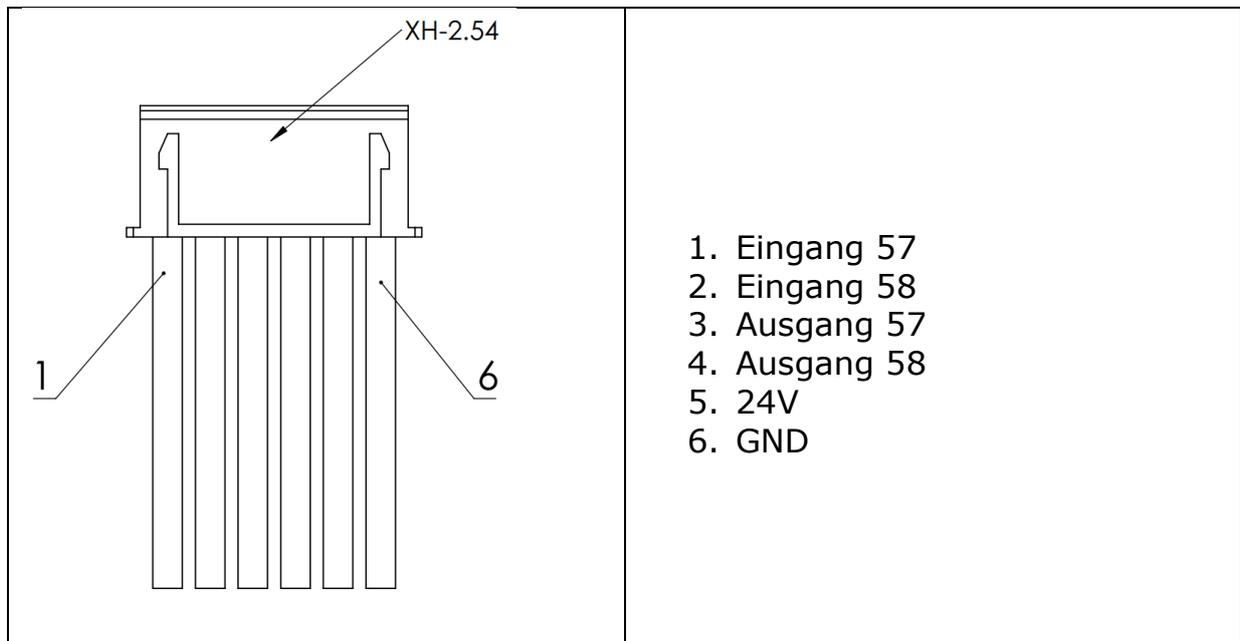
21 Ausgänge und Eingänge am Roboterarm

Die Version B des Astorino-Roboters ist mit zwei Eingängen 24V und zwei Ausgängen 24V (PNP) am JT3-Arm ausgestattet.

In der Astorino-Software und im AS-System sind ihre Nummern 57 und 58 für die Ausgänge und 1057, 1058 für die Eingänge.

Der eingesetzte Steckverbinder ist weiblich XH-2.54 6 Pin.

Setzen Sie sich bitte mit der nachfolgenden Tabelle zur Verbindung der EINGÄNGE/AUSGÄNGE in Kenntnis

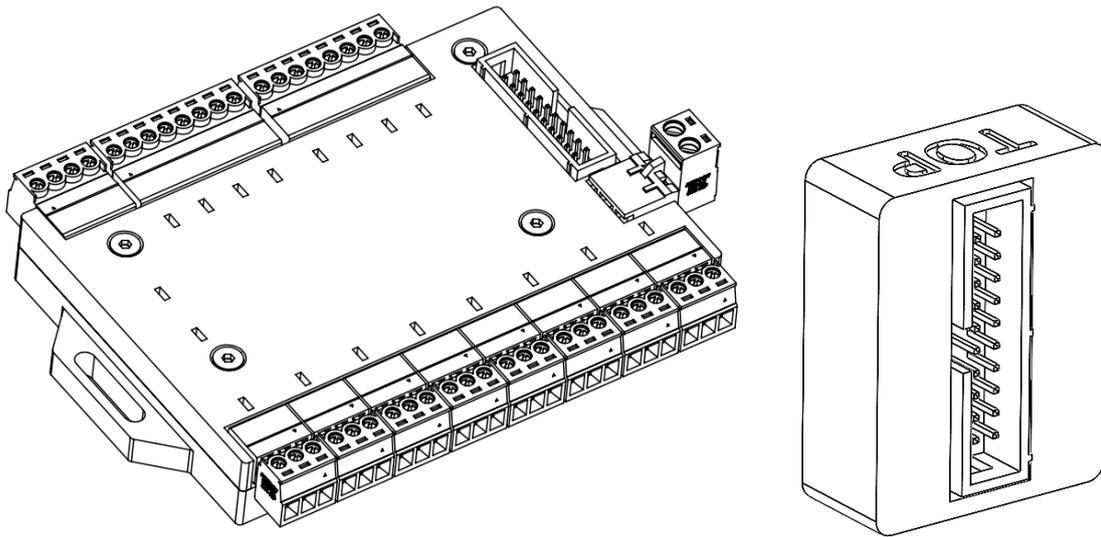


WARNHINWEIS!

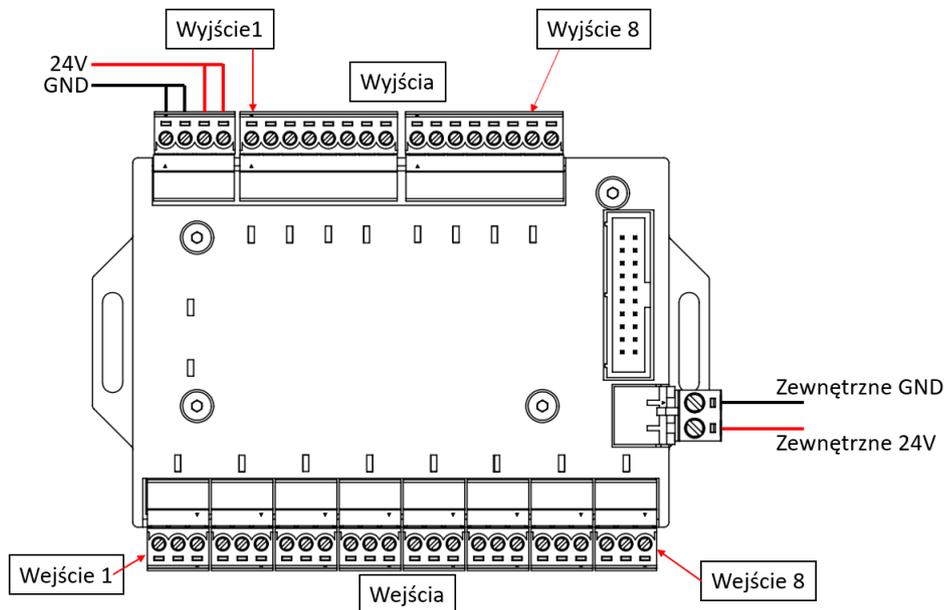
Jeder AUSGANG liefert 300 mA Strom. Bitte das Limit nicht überschreiten, da die Hauptplatte dadurch beschädigt werden kann.

22 I/O 24V-Modul

Das Modul I/O 24V ist als Option erhältlich und wird unabhängig vom Astorino-Roboter verkauft. **Das Modul erweitert die Anzahl der Eingänge/Ausgänge, es ist eine Lasche für den Standard-IO 3.3V**



Eingang 1 Ausgänge Eingang 8



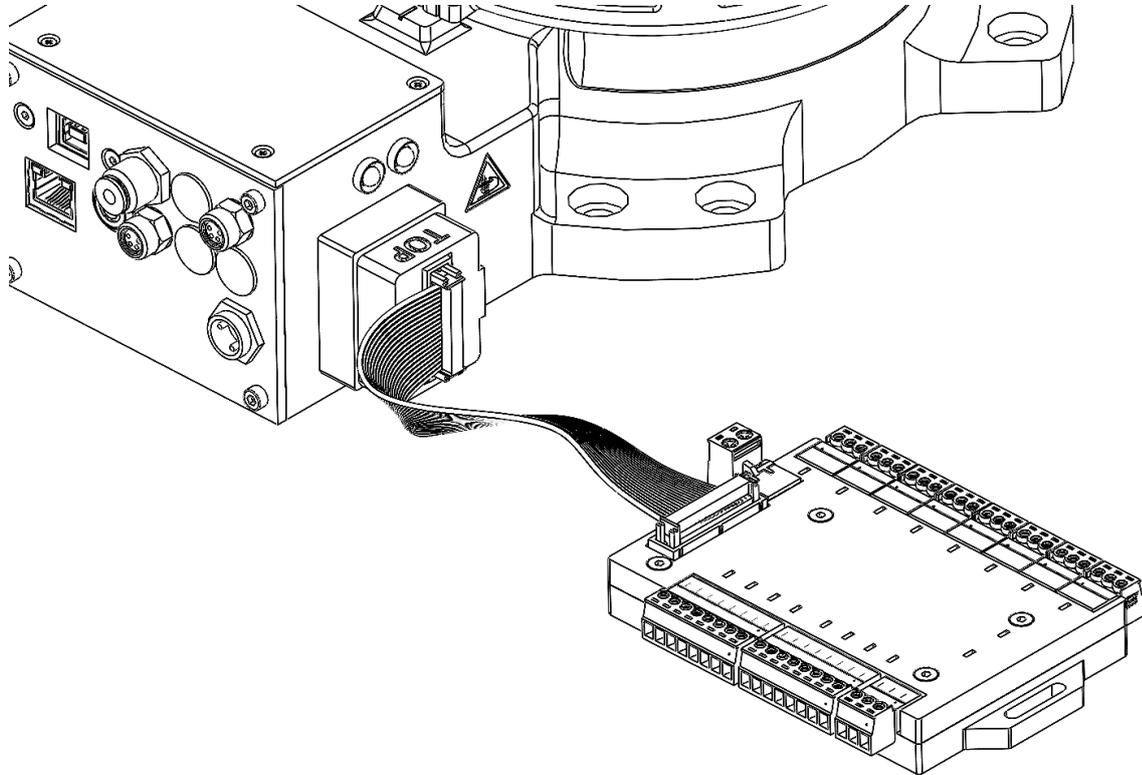
Eingang 1 Ausgänge Eingang 8

Außen-GND
Außen 24V

Man kann insgesamt 8 Eingänge 24V und 8 Ausgänge 24V anschließen. Jeder Ausgang liefert einen Strom von 300mA (ca. 7,2W).

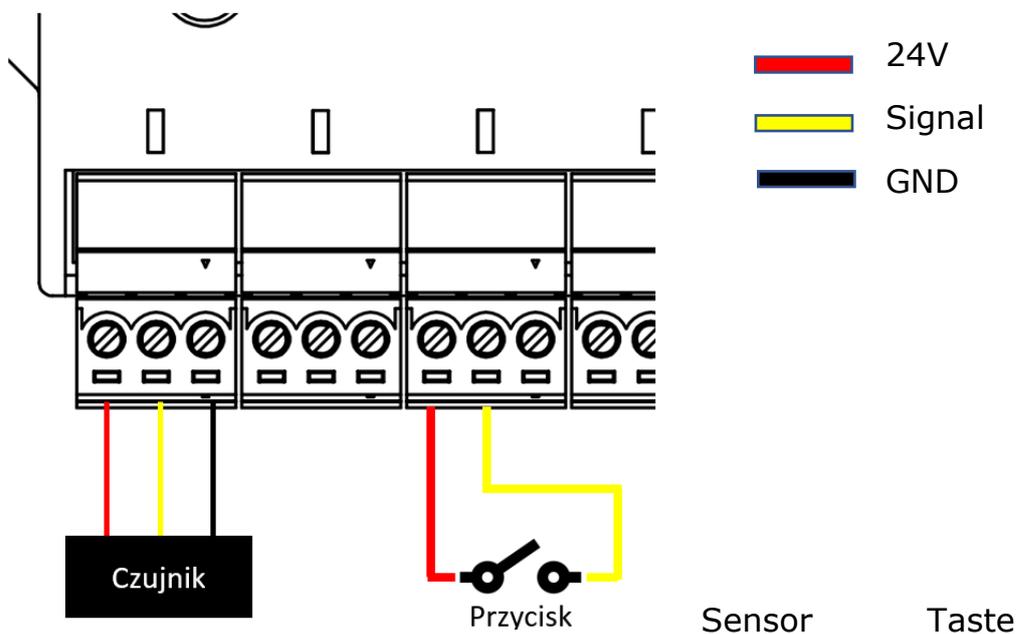
22.1 Anschluss an den Roboter

Das Modul wird über ein mitgeliefertes 30cm lange Flachbandkabel angeschlossen.



22.2 Anschließen der Eingänge

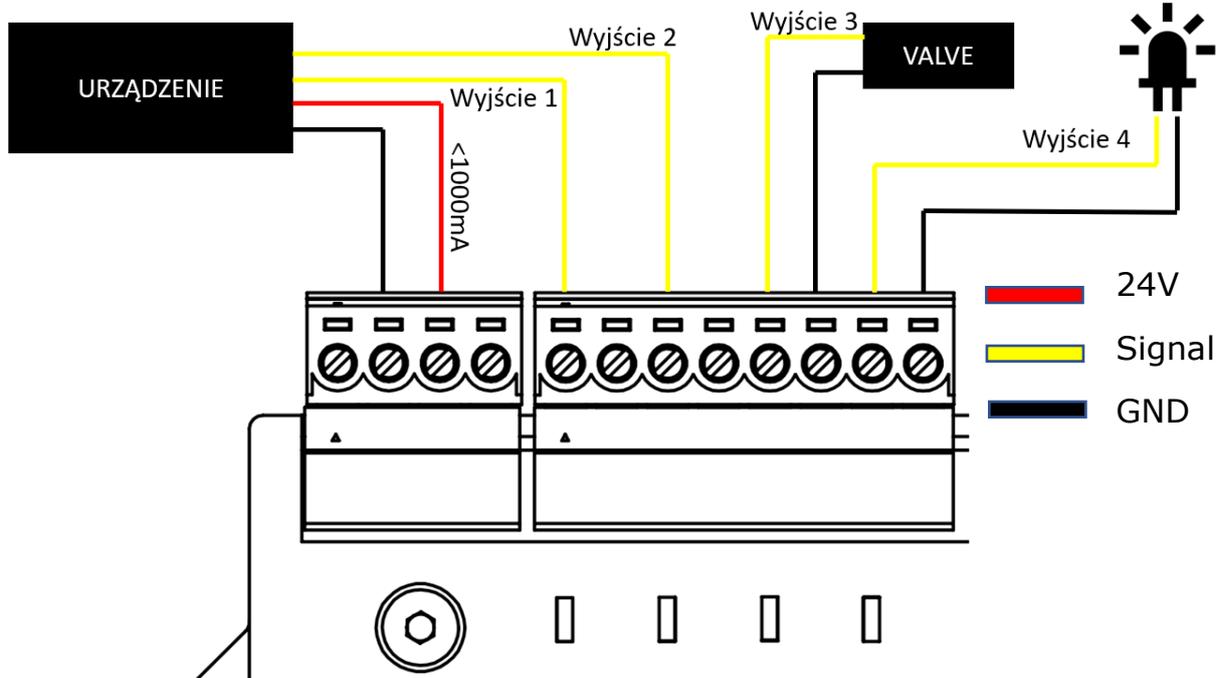
Ein Beispielschluss der Außengeräte an die Robotereingänge



22.3 Anschließen der Ausgänge

Ein Beispielanschluss der Außengeräte an die RoboterAusgänge.

GERÄT Eingang 1 Eingang 2 Eingang 3 Eingang 4



23 TCP-MODBUS

Das Modbus ist ein Datenübertragungsprotokoll, welches ursprünglich durch Modicon (zurzeit Schneider Electric) 1979 zum Einsatz mit programmierbaren SPS-Steuerungen (PLC) veröffentlicht wurde. Modbus ist de facto ein Standardkommunikationsprotokoll geworden und ist ein allgemein erhältliches Mittel zum Anschließen von industriellen elektronischen Geräten.



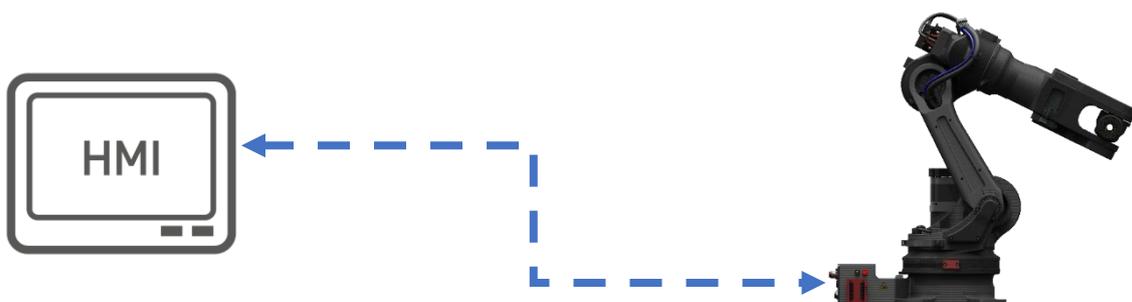
WARNHINWEIS!

Funktioniert Astorino als ein Gerät vom Typ Client im Modbus-Netz, werden die Register während des Roboterbetriebs nicht aufgefrischt!

23.1 Betriebsarten des Modbus-Netzes

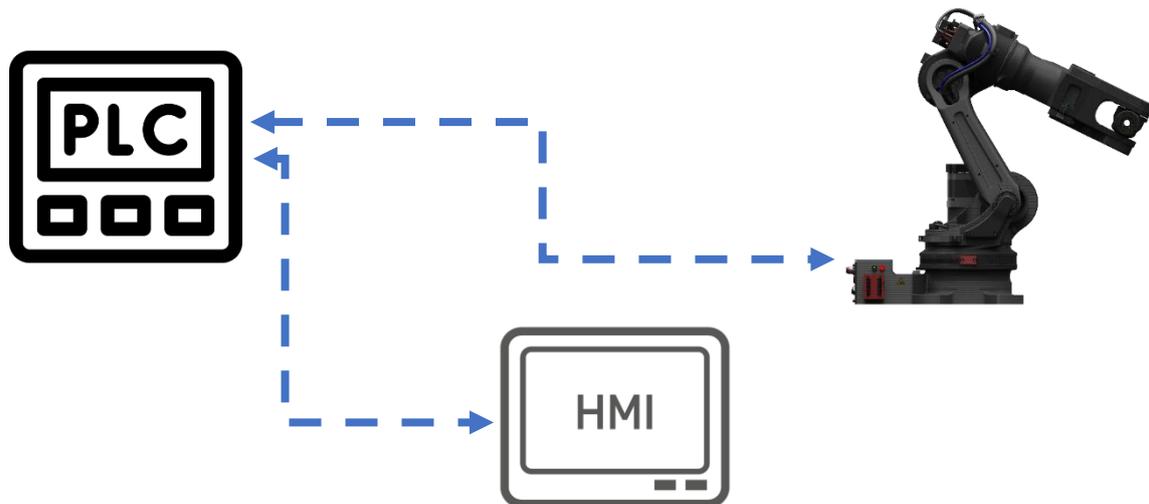
Astorino kann in zwei Netzbetriebsarten funktionieren. Der Roboter kann als Server oder als Client eingestellt werden. Das Signalpool bleibt gleich.

- Modbus-Server



ASTORINO Betriebsanleitung

- Modbus-Client



23.2 Modbus-Objektypen bei Astorino

Die folgenden Objekttypen können durch den Modbus-Server zum Modbus-Client-Gerät bedient werden. Die Adressen sind für die originale Modicon-Spezifikation repräsentativ. Nach derzeitigem Standard kann die Adresse 0 - 65535 betragen mit dem Objekttyp, welcher durch den Befehl zur Ableseung oder Speicherung des Registers identifiziert wird. Der Astorino-Roboter kann 3x Input Register 3x Holding Register ablesen und speichern, was zusätzliche 56 Eingänge und 56 Ausgänge ergibt.

Der Astorino-Roboter nutzt den Standard-PORT: 502

Wenn Astorino als Server arbeitet

Objekttyp	Astorino-Funktion	Größe	Adressraum
Input Register	Ausgänge	16 Bits	30001 - 30003
Holding Register	Eingänge	16 Bits	40001 - 40003

Wenn Astorino als Client arbeitet

Objekttyp	Astorino-Funktion	Größe	Adressraum
Input Register	Ausgänge	16 Bits	30001 - 30003
Holding Register	Eingänge	16 Bits	40001 - 40003

23.3 Konfiguration des Ethernet-Ports

Netzadressen gemäß Konfiguration PLC/HMI und Ethernet-Einstellungen für Modbus TCP Server oder Client einstellen.

Astorino als Modbus TCP Server

Astorino als Modbus TCP Client

Collision det Ethernet **Firmware** ◀ ▶

Ethernet Settings **ModbusTCP Server** ▼

IP Address
 . . .

Subnet Address
 . . .

Gateway Address
 . . .

DNS Address
 . . .

Modbus TCP port: 502 Connected

 **Save**

Collision det Ethernet **Firmware** ◀ ▶

Ethernet Settings **ModbusTCP Client** ▼

IP Address
 . . .

Subnet Address
 . . .

Gateway Address
 . . .

DNS Address
 . . .

Modbus Server Address:
 . . .

Modbus TCP port: 502 Connected

 **Save**

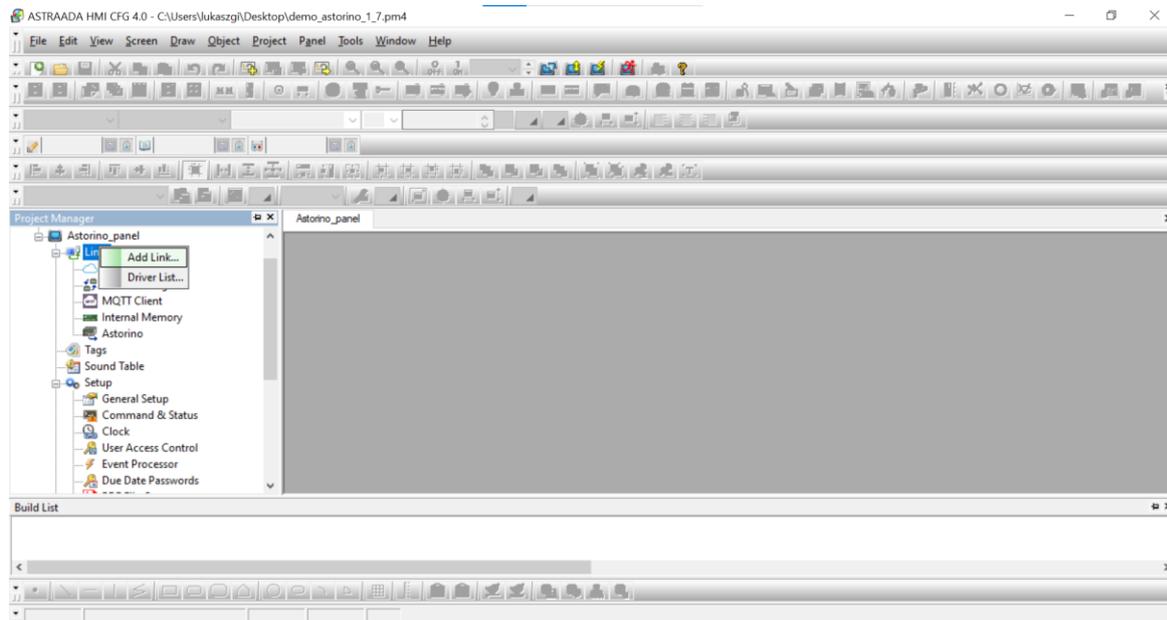
ASTORINO Betriebsanleitung

23.4 ASTRAADA HMI-Panel – Beispiel

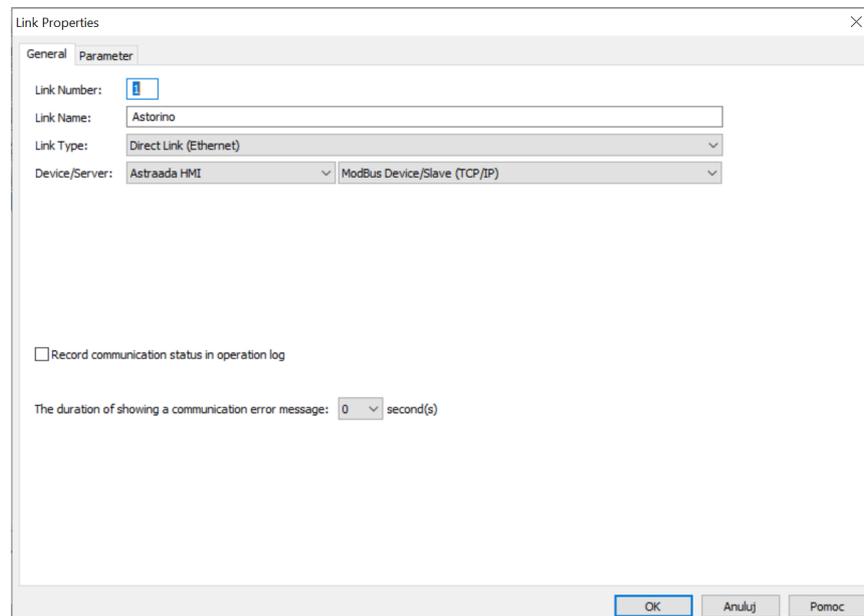
Dieses Beispiel zeigt nur eine Konfiguration des Kommunikationsprotokolls Modbus TCP an HMI ASTRAADA-Panels. Mehr Informationen kann man in ASTRAADA HMI-Handbüchern finden. Das Beispiel nutzt den Astorino-Roboter als Kommunikationsserver.

Das Programm ASTRAADA HMI CFG aufmachen und entsprechendes HMI-Panel bei Optionen einstellen.

1. Link hinzufügen

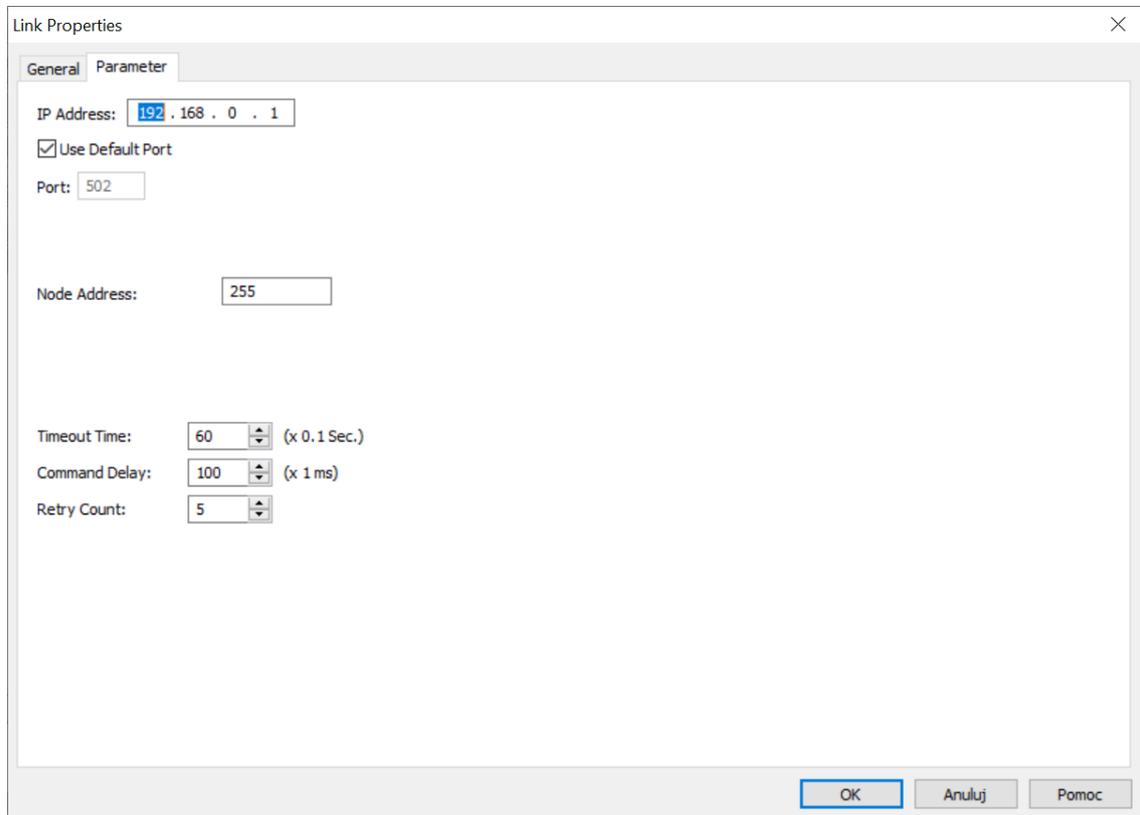


2. Verbindung konfigurieren, ModBus-Gerät (Device/Slave TCP/IP) hinzufügen.



ASTORINO Betriebsanleitung

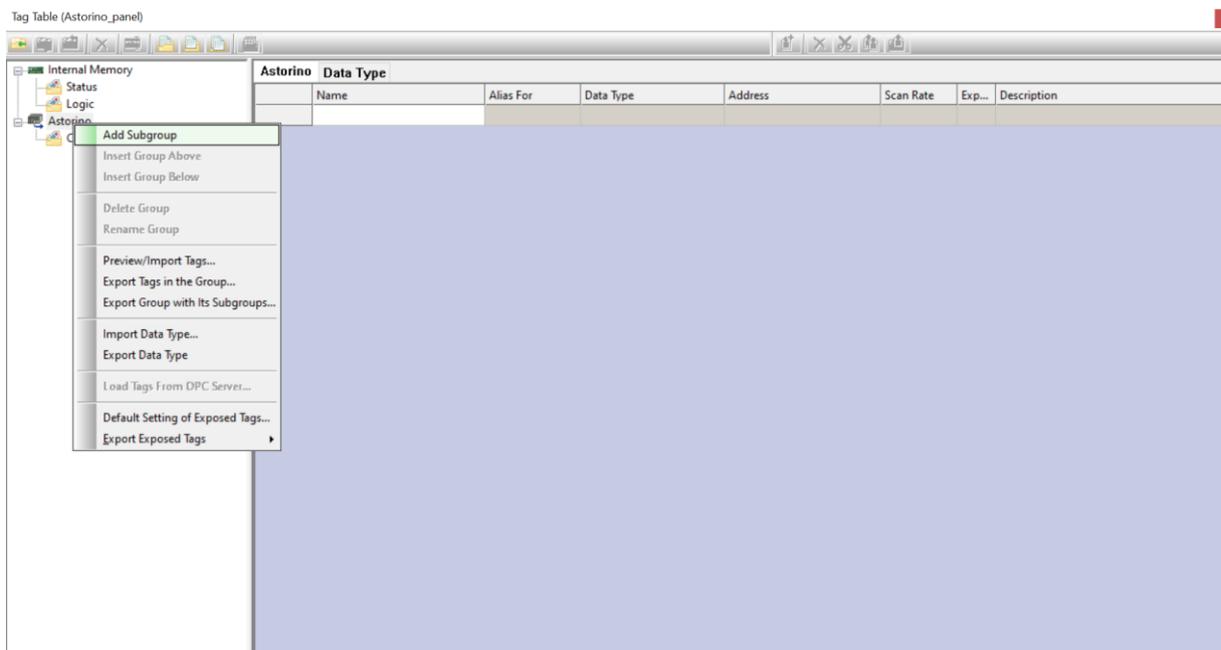
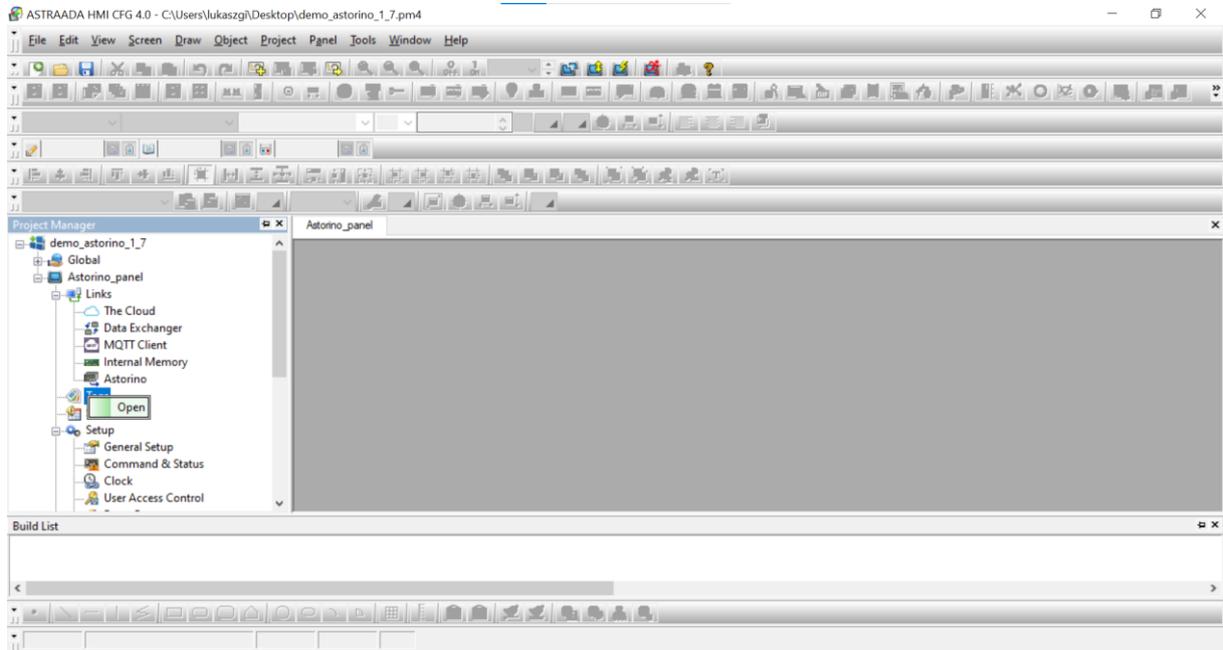
3. Die IP-Adresse, das Zeitlimit, die Anzahl der wiederholten Versuche etc. einstellen. Man empfiehlt die Einstellung der Anzahl von wiederholten Versuche für mindestens 3 und die Überschreitungen des Zeitlimits für mindestens 3 Sekunden.



ASTORINO Betriebsanleitung

4. Datentyp einstellen

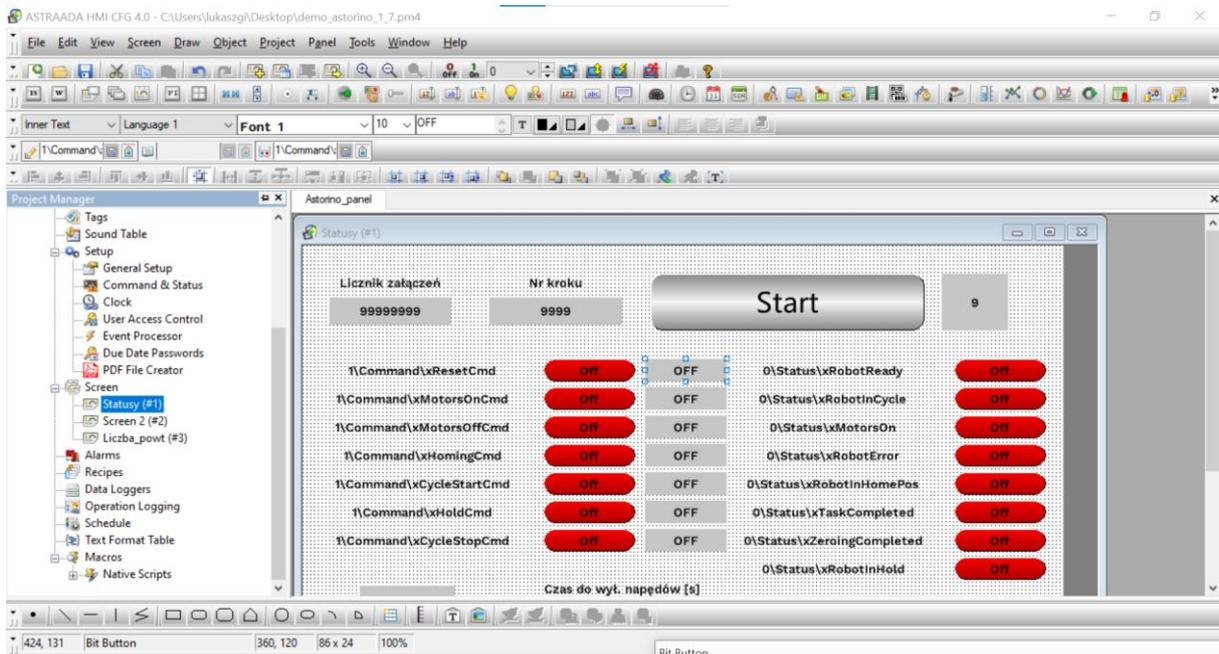
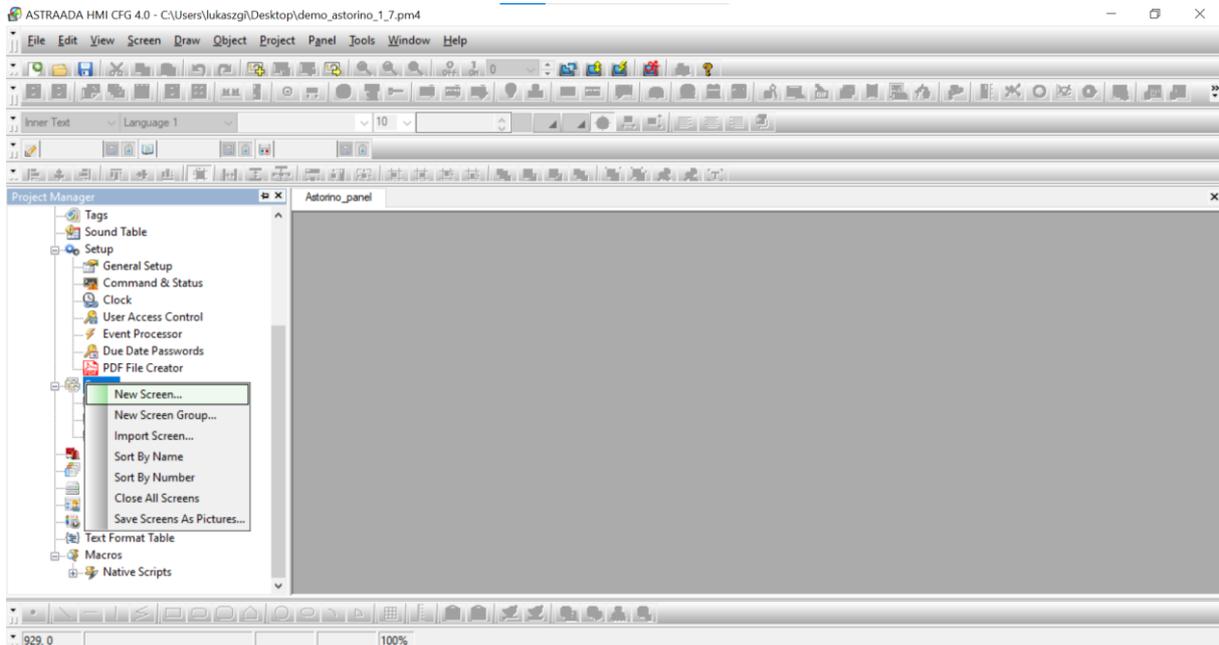
Bei diesem Beispiel werden Eingänge und Ausgänge als normal und dediziert eingestellt.



Command	Data Type						
	Name	Alias For	Data Type	Address	Scan Rate	Exp...	Description
1	xResetCmd		Bit	40001.0	Normal	No	Error reset command on the robot
2	xMotorsOnCmd		Bit	40001.1	Normal	No	Motors on command on the robot
3	xMotorsOffCmd		Bit	40001.2	Normal	No	Motors off command on the robot
4	xHomingCmd		Bit	40001.3	Normal	No	Robot position homing command
5	xCycleStartCmd		Bit	40001.4	Normal	No	Start cycle command on the robot
6	xHoldCmd		Bit	40001.5	Normal	No	Command to put the robot into hold mode
7	xCycleStopCmd		Bit	40001.6	Normal	No	Command to stop the robot's work cycle

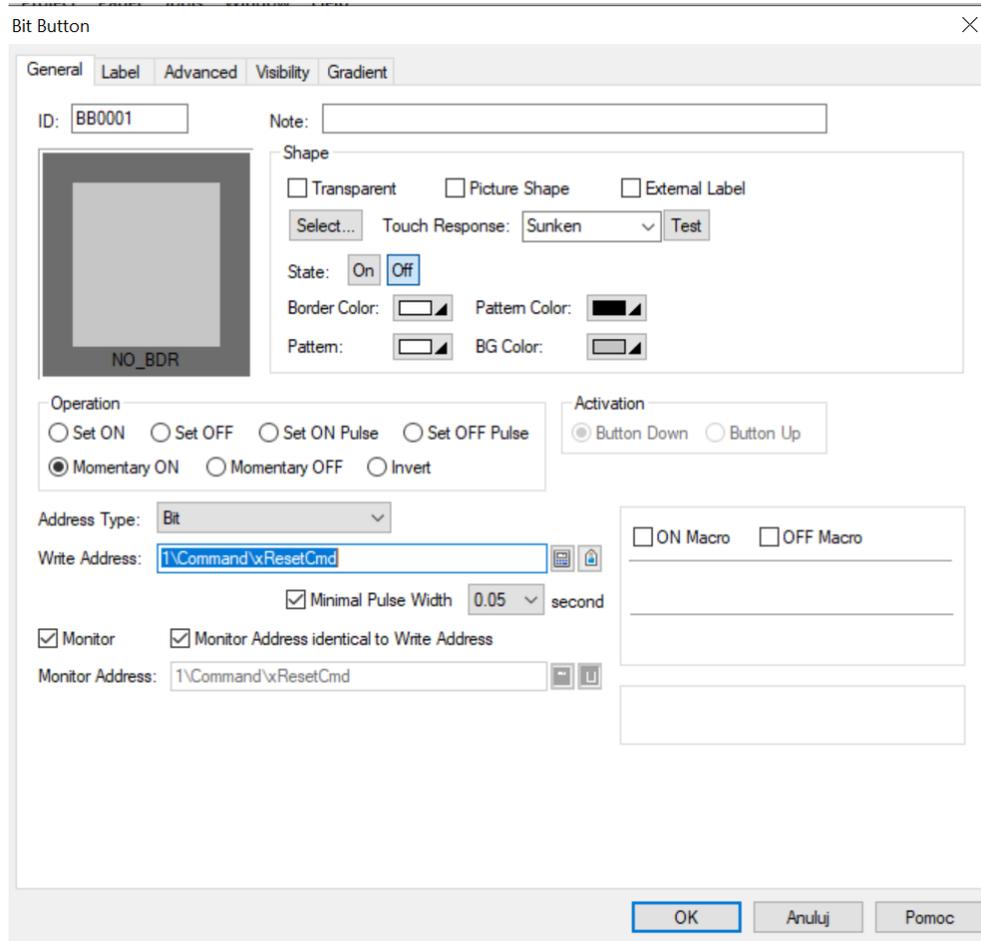
ASTORINO Betriebsanleitung

5. Neuen Bildschirm hinzufügen und seinen Inhalt konfigurieren.



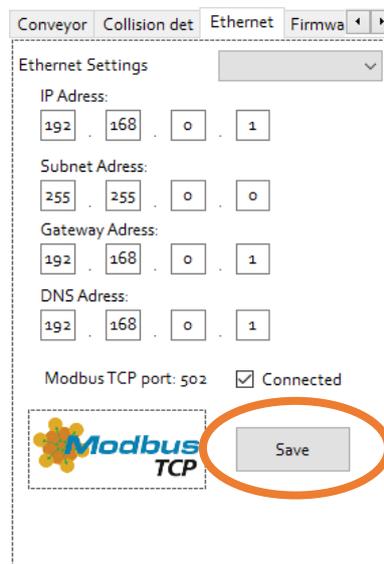
ASTORINO Betriebsanleitung

6. Tasten zur Ablesung/Speicherung jeweiliger Adressen konfigurieren.



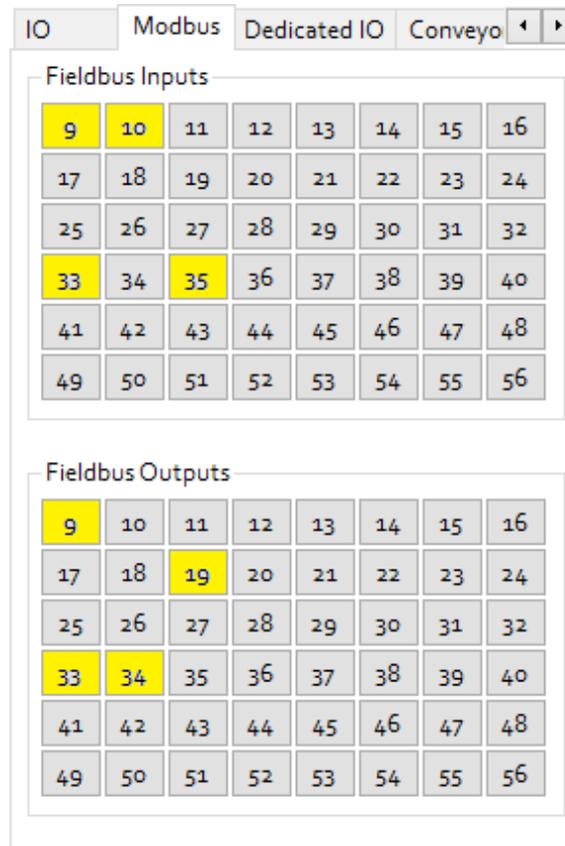
7. Speichern und das Projekt HMI-Panel senden.

Funktioniert die Kommunikation richtig, wird der Status in diesem Menü angezeigt.



ASTORINO Betriebsanleitung

Der Registerzustand wird auf dieser Karte angezeigt. Wenn der Registerbyte eingeschaltet ist, leuchten die Tasten gelb.



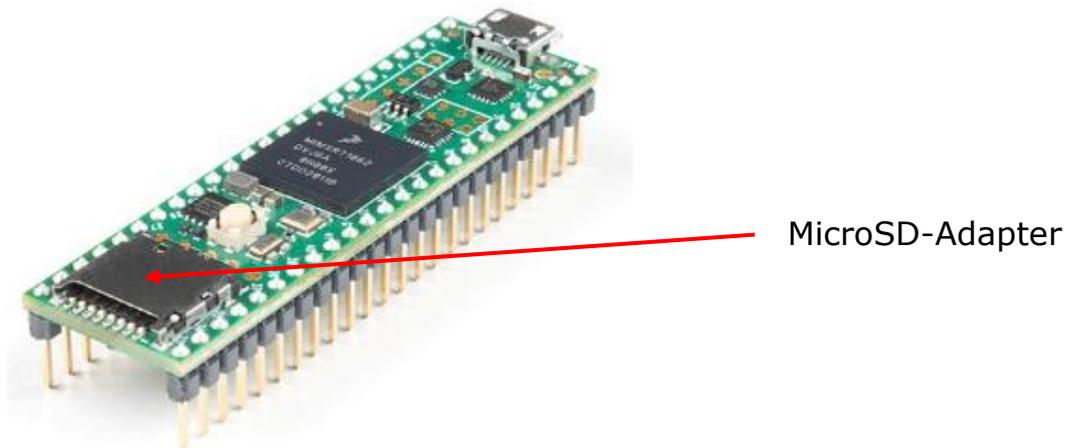
23.5 Nutzung der Modbusregister als wirkliche Werte

Zur Ablesung und Speicherung der Zahlendaten mittels Modbus-Register kann man die **BITS**-Funktion anwenden.

```
.PROGRAM BIT
  BITS 9,16 = 12082 ;sets a number as bits
  x = BITS(1009,16) ;reads a number from bits
.END
```

24 Kalibrierung

Die Kalibrierung ist nach dem Aufbau des Roboters durchzuführen. Nach der Kalibrierung werden die Daten des Auf-null-Setzens auf der Karte microSD, die auf der Hauptplatte des Prozessors im Robotersockel, platziert ist, gespeichert:



Das bedeutet, dass der Roboter nicht jedes Mal erneut kalibriert werden muss, wenn die Einspeisung abgeschaltet ist.

Das Kalibrierungsverfahren ist im Kalibrierungshandbuch beschrieben.

25 Informationen über den Hersteller

Kawasaki Robotics Astorino
BETRIEBSANLEITUNG

06.2024: 6. Ausgabe

Eine Veröffentlichung von: ASTOR und Kawasaki Robotics GmbH

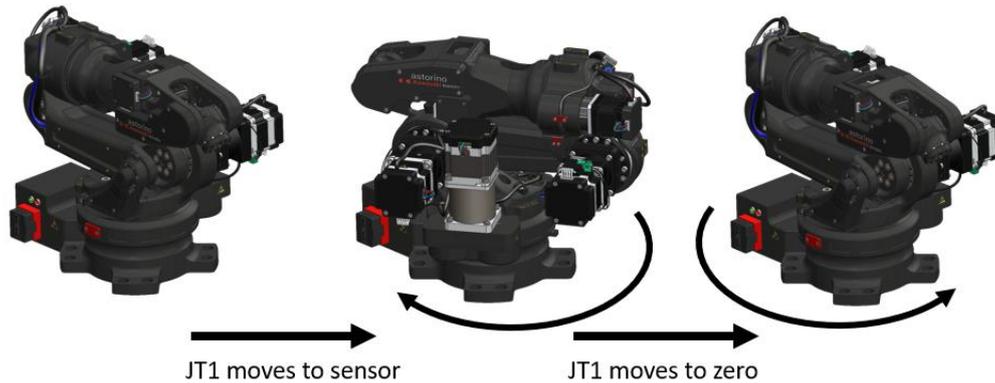
Copyright © 2024 ASTOR & KAWASAKI Robotics GmbH.
Alle Rechte vorbehalten.

Anlage A – Auf null setzen - Standardverfahren

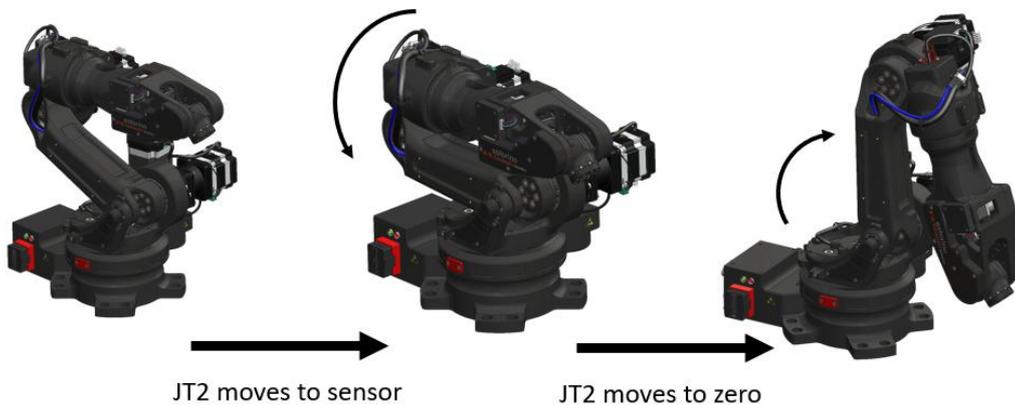
Der Astorino-Roboter ist mit Inkrementaldrehgebern ausgestattet, deswegen müssen alle Achsen nach seiner Einschaltung auf null gesetzt werden.

Dieses Verfahren ist automatisch und in der voreingestellten Konfiguration im Folgenden beschrieben.

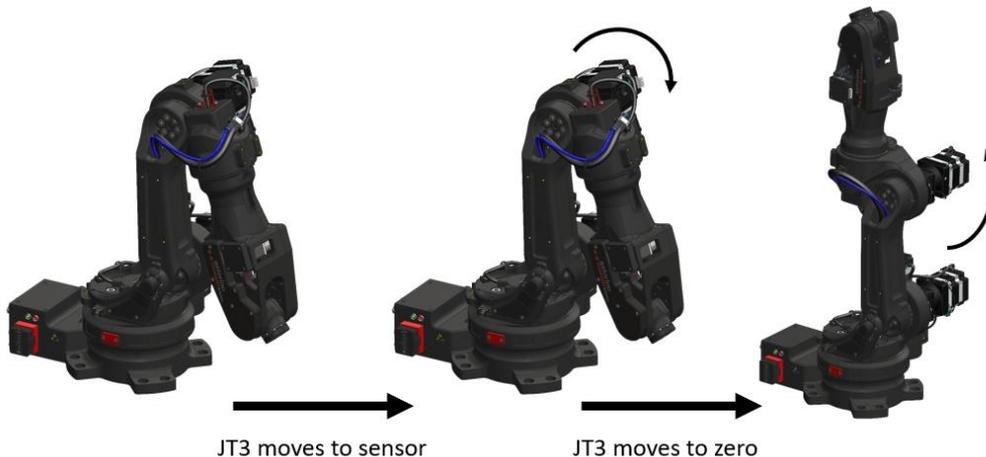
JT1:



JT2:

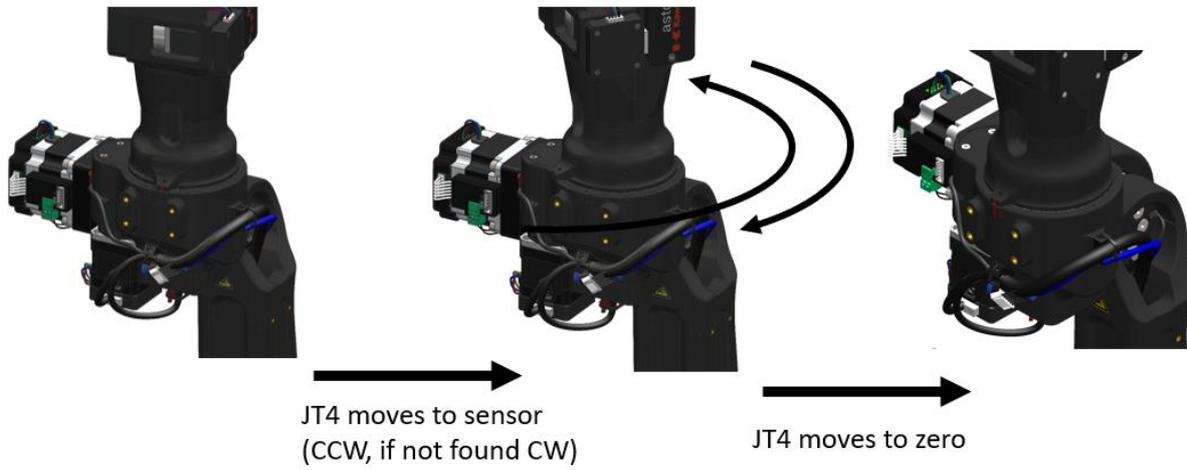


JT3:

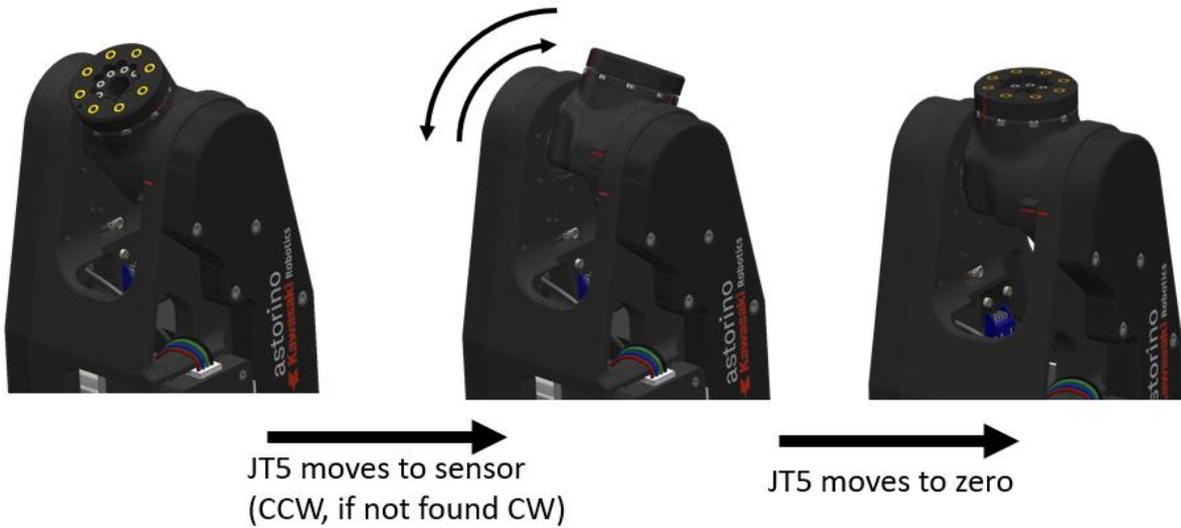


ASTORINO Betriebsanleitung

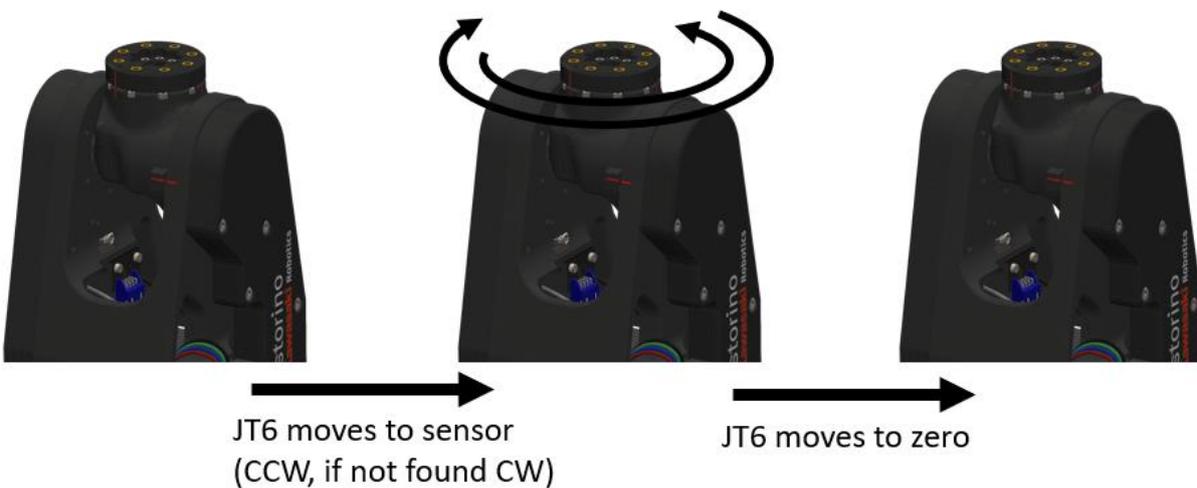
JT4:



JT5:



JT6:



Anlage B – PETG-Material

PETG ist eines der vielseitigsten Materialien zum 3D-Drucken, stark und leicht auszudrucken. Seine Beliebtheit ist in den letzten Jahren gestiegen als eine Alternative für PLA. PETG ist eine Variante von PET, die beim 3D-Druck angewandt wird. Das G am Ende steht für Glycol-modifiziert. Diese Änderung beeinflusst die chemische Struktur, wodurch das Material noch transparenter, weniger empfindlich und leichter zu bearbeiten ist.

PETG hat sehr interessante Eigenschaften und seine ernsthaften Konkurrenten sind PLA und ABS. Haupteigenschaften, die berücksichtigt werden sollen, sind folgende:

1. **Steife:** Das Material ist schwer zu deformieren, darunter durch Dehnen und Biegen. PLA ist steifer als PETG und PETG sowie ABS sind fast genauso steif.
2. **Beständigkeit:** es ist in der Regel schwieriger PETG als PLA und ABS zu brechen. In Anlehnung an technische Eigenschaften, ist PETG nicht nur beständiger als ABS, sondern ist die Haftung zwischen den Schichten höher, was insgesamt eine bessere Beständigkeit ergibt.
3. **Wärmebeständigkeit:** PETG wird bei der Temperatur 80°C weich, wobei PLA beginnt bei einer Temperatur von 50°C zu erweichen. Das ABS hat jedoch die höchste Wärmebeständigkeit, indem es bei der Temperatur 105°C erweicht.
4. **Geruchloses Drucken:** Im Gegensatz zu ABS, erzeugt PETG beim Drucken keinen Geruch.
5. **Recyclingfähig:** Wegen seiner Beliebtheit, haben die meisten Städte die erforderliche Infrastruktur zum Recycling von PETG.

Wie ist PETG zu drucken

Hotend-Temperatur: PETG wird gewöhnlich bei einer Temperatur von 220-250°C gedruckt und kann mittels fast jeden 3D-Druckers gedruckt werden, darunter mit vollständig metallischen Hotends oder mit solchen, die ein internes PTFE-Rohr haben.

1. **Oberflächentemperatur:** Damit PETG gedruckt werden kann, ist es erforderlich, einen beheizten Tisch mit einer Temperatur von 60-90°C zu verwenden. Man empfiehlt auch Leim, wie Papierleim, zur Druckfläche hinzuzufügen.
2. **Geschlossene 3D-Drucker:** Obwohl die Anwendung von geschlossenen 3D-Druckern nicht erforderlich ist, empfehlen wir die Schwankungen der Raumtemperatur zu vermeiden.
3. **Schichtenventilator:** Man empfiehlt einen Schichtenventilator beim Drucken von PETG anzuwenden.
4. **Verziehen:** PETG hat einen verminderten Hitzeschrumpf, es ist also fürs Verziehen nicht anfällig und dadurch entstehen Teile mit guten Maßtoleranzen

Anlage 3 – Anschluss Typ PNP

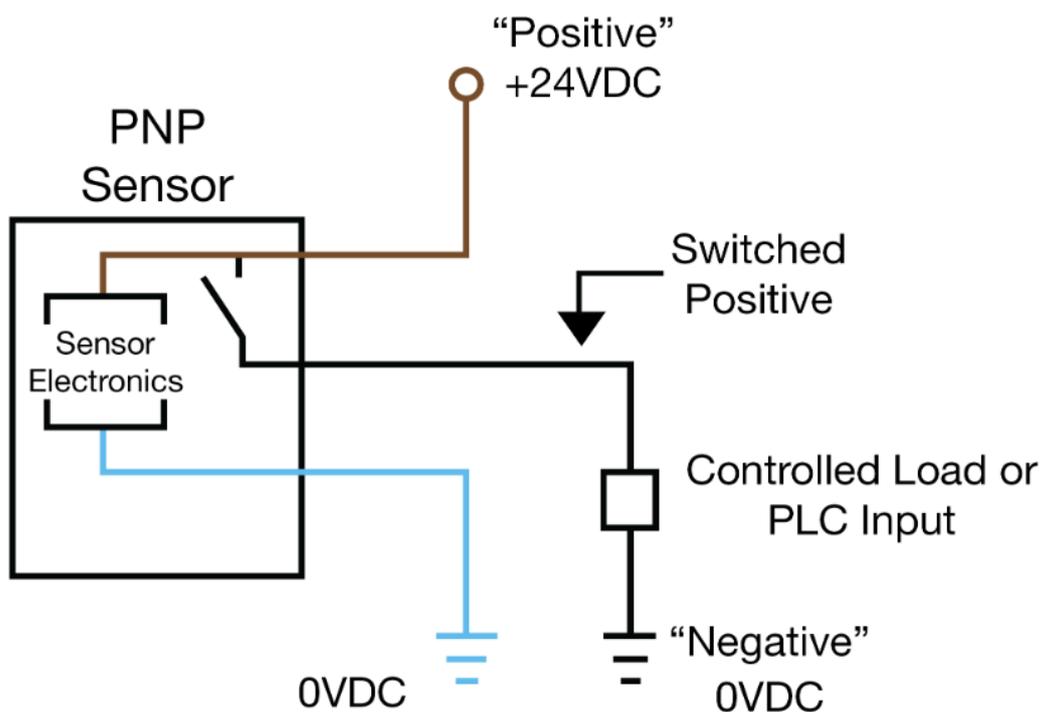
PNP bedeutet Positiv, Negativ, Positiv. Es ist ebenfalls als Sourcing bekannt. Am IO-Modul ist der PNP-Eingang, wenn er nicht gesteuert wird, zum hohen Zustand, z. B. +24V, angehoben.

In Europa wird meistens der Typ des PNP-Eingangs/Ausgangs angewandt, der mit einem PNP-Sensor oder mit einem Stellmotor zum Einsatz kommt. Weniger verbreitet sind heutzutage NPN-Eingangskarten, die in Asien populär waren und einen Sensor von Typ NPN zum richtigen Funktionieren erfordern.

Das ist ein einfacher Weg, sich zu merken, wie ein PNP-Gerät mit 3 Leitungen anzuschließen ist:

PNP = umgeschaltet positiv

"Umgeschaltet" bezieht sich darauf, welche Seite der kontrollierten Last (Relais, kleines Display, SPS-Eingang) elektrisch umgeschaltet wird, oder die Last wird an die negative angeschlossen und die positive wird umgeschaltet (PNP).



Anlage 4 – Teensy 4.1

Teensy 4.1 ist der stärkste Mikrocontroller, kompatibel mit Arduino, der zurzeit auf dem Markt erhältlich ist. Er stützt sich auf NXP i.MX RT1062 ARM Cortex-M7, der mit einer Frequenz von 600MHz arbeitet mit der Möglichkeit des Umtaktens. Vielleicht ist das Beste dabei, dass er mit dem populären Programmierumfeld Arduino IDE und auch mit vielen bestehenden Arduino-Bibliotheken kompatibel ist, wodurch er im Gegensatz zu vielen anderen, erhältlichen fortgeschrittenen Mikrocontrollern sehr einfach zu starten ist.

Das Herz des Mikrocontrollers i.MX RT1060 ist der Prozessorkern ARM Cortex-M7, der viele fortgeschrittene Funktionen in die echte Plattform der Echtzeit-Mikrocontrollern einbringt.

Cortex-M7 ist der doppelte superskalare Prozessor, was bedeutet, dass M7 zwei Befehle pro Uhrzyklus, bei 600MHz ausführen kann! Natürlich gehört die Ausführung von zwei Befehlen gleichzeitig vom Befehl und von den Registern, welche den Kompilator bestellen. Die vorläufigen Vergleichstests haben ergeben, dass der Code C ++, kompiliert durch Arduino IDE, die Tendenz hat, zwei Befehle pro Zyklus mit etwa 40% bis 50% der Zeit während der Ausführung der nummerisch intensiven Arbeit unter Anwendung von ganzen Zahlen und Indexzahlen zu erreichen.



Mehr Informationen kann man auf der Internetseite PJRC finden.

<https://www.pjrc.com/store/teensy41.html>