

LOW  
COST  
AUTOMATION  
by igus®

# igus® Robot Control V15 Benutzerhandbuch



---

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Kontakt . . . . .	6
1.2	Bestimmungsgemäße Verwendung . . . . .	6
1.3	Zielgruppe und Qualifikation . . . . .	6
1.4	Verwendete Symbole . . . . .	7
1.5	Produktsicherheit . . . . .	7
1.6	Vorschriften . . . . .	7
<b>2</b>	<b>Übersicht über das System</b>	<b>8</b>
2.1	Unterstützte Kinematiken . . . . .	8
2.2	Spezifikationen . . . . .	9
2.3	Mechanische Abmessungen . . . . .	10
<b>3</b>	<b>Sicherheitshinweise</b>	<b>11</b>
<b>4</b>	<b>CE-Konformität</b>	<b>12</b>
4.1	CE-Kennzeichnung . . . . .	12
4.2	Integration von Sicherheitskomponenten zur Herstellung der CE-Konformität . . . . .	12
<b>5</b>	<b>Anforderungen</b>	<b>13</b>
5.1	Umgebungsbedingungen . . . . .	13
5.2	Softwareanforderungen . . . . .	13
<b>6</b>	<b>Schnellstart-Anleitung</b>	<b>15</b>
6.1	Einrichten und Anschließen . . . . .	15
6.2	Einschalten . . . . .	16
6.3	Verbinden und Bewegen des Roboters . . . . .	16
6.3.1	Vorbereitung mit dem integrierten Computer . . . . .	16
6.3.2	Erstes Bewegen des Roboters . . . . .	17
<b>7</b>	<b>Installation</b>	<b>20</b>
7.1	Übersicht . . . . .	20
7.2	Motormodul . . . . .	20
7.3	Digital-I/O-Modul . . . . .	21
7.4	Motorbremse . . . . .	22
7.5	Steuerrechner . . . . .	23
7.5.1	Spannungsversorgung . . . . .	23
7.5.2	Busverbindung . . . . .	23
7.5.3	LAN-Anschluss . . . . .	25
7.5.4	USB-Anschluss . . . . .	25
7.5.5	Verbindung Handsteuergerät . . . . .	25
7.6	Handbediengerät . . . . .	26

<b>8</b>	<b>Bewegen eines Roboters mit iRC</b>	<b>27</b>
8.1	Die grafische Benutzeroberfläche . . . . .	27
8.1.1	Simulieren eines Roboters . . . . .	28
8.1.2	Verbinden eines realen Roboters . . . . .	29
8.1.3	Navigation und Bewegen des Roboters in der 3D-Ansicht . . . . .	29
8.1.4	Szenenelemente . . . . .	30
8.1.5	Bewegungsschweif . . . . .	30
8.2	Anschließen des Roboters . . . . .	33
8.2.1	Verbindung zur Hardware . . . . .	33
8.2.2	Den Roboter bewegen . . . . .	34
8.3	Referenzieren des Roboters . . . . .	34
8.3.1	Schrittweise Anleitung der Referenzierung . . . . .	34
8.3.2	Referenzierungsprogramm . . . . .	35
8.4	Bewegen des Roboters . . . . .	36
8.4.1	Gamepad . . . . .	36
8.4.2	Softwaretasten . . . . .	37
8.5	Nicht erreichbare Positionen und Singularitäten . . . . .	38
8.6	Starten von Roboterprogrammen . . . . .	38
8.7	Digitale Ein- und Ausgänge . . . . .	40
8.8	Software-Schnittstellen . . . . .	41
8.8.1	App-Schnittstelle . . . . .	42
8.9	Aktualisieren der Software . . . . .	42
<b>9</b>	<b>Bewegen des Roboters per Handbediengerät</b>	<b>44</b>
9.1	Roboter Zurücksetzen und aktivieren . . . . .	44
9.2	Bewegen des Roboters mit dem 3-Achsen-Joystick . . . . .	45
9.3	Programmauswahl . . . . .	45
9.4	Referenzierung . . . . .	46
9.5	Starten und Stoppen eines Programms . . . . .	47
9.6	Setzen der digitalen Ausgänge und Ablesen der digitalen Eingänge . . . . .	47
9.7	Anzeige von Statusinformationen . . . . .	48
<b>10</b>	<b>Programmierung eines Roboters mit iRC</b>	<b>49</b>
10.1	Programmeditor . . . . .	49
10.1.1	Ändern der Befehlssequenz . . . . .	50
10.1.2	Position nachbessern . . . . .	50
10.1.3	Startbefehl festlegen . . . . .	51
10.2	Roboter- und Logikprogramme . . . . .	51
10.3	Kommentare und Informationen im Programm . . . . .	52
10.3.1	Informationen zum Programm . . . . .	52
10.3.2	Beschreibungen . . . . .	54
10.3.3	Kommentare . . . . .	54
10.4	Bewegung . . . . .	54
10.4.1	Abbruchbedingungen . . . . .	54
10.4.2	Beschleunigung und Glättung . . . . .	55
10.4.3	Achsbewegung . . . . .	55

10.4.4	Lineare Bewegung	56
10.4.5	Achsbewegung zu kartesischer Position	57
10.4.6	Relative Bewegung	58
10.4.7	Kreisbewegung	58
10.4.8	Dauerbewegung	60
10.4.9	Pfad	60
10.4.10	Punktfolge	61
10.5	Koordinatensysteme	63
10.5.1	Benutzerkoordinatensystem kopieren	63
10.6	Greifer und digitale Ein-/Ausgänge	63
10.6.1	Digitale Eingänge	63
10.6.2	Digitale Ausgänge	64
10.6.3	Globale Signale	64
10.6.4	Öffnen/Schließen des Greifers	64
10.7	Programmfluss	64
10.7.1	Bedingungen	64
10.7.2	Stop	66
10.7.3	Pause	66
10.7.4	Warten	66
10.7.5	Wenn-dann-sonst	67
10.7.6	Schleifen	67
10.7.7	SwitchCase	67
10.7.8	Rasterbewegungen / Palettierung	68
10.7.9	Unterprogramme	70
10.8	Variablen und Variablenzugriff	70
10.8.1	Benutzervariablen	71
10.8.2	Systemvariablen	71
10.8.3	Persistente Variablen	72
10.8.4	Zugriff auf Elemente	72
10.8.5	Berechnungen mit Variablen	73
10.8.6	Variablen beobachten	73
10.9	Kamera	73
10.10	Appfunktion	75
10.11	Audio	75
10.11.1	ITon abspielen	76
10.11.2	Sprachausgabe	76
<b>11</b>	<b>Koordinatensysteme</b>	<b>78</b>
11.1	Vordefinierte Koordinatensysteme	78
11.1.1	Das Basiskoordinatensystem	78
11.1.2	Das Werkzeugkoordinatensystem	78
11.2	Benutzerkoordinatensysteme	78
11.2.1	Erstellen eines neuen BKS	78
11.3	Koordinatensysteme wechseln	79

<b>12 Hardwarekonfiguration</b>	<b>82</b>
12.1 Ein-/Ausgänge . . . . .	82
12.1.1 Elektrische Integration . . . . .	82
12.1.2 Software-Konfiguration . . . . .	83
12.1.3 Sensoren und Taster anschließen . . . . .	84
12.1.4 Aktoren anschließen . . . . .	84
12.2 Roboterkonfiguration - Achslängen . . . . .	85
12.3 Motorbremse . . . . .	86
12.3.1 Elektrische Integration . . . . .	86
12.3.2 Software-Konfiguration . . . . .	87
12.4 Zusatzachsen . . . . .	87
12.5 Konfiguration der Motorsteuerungen . . . . .	89
<b>13 Softwarekonfiguration</b>	<b>90</b>
13.1 Benutzeroberfläche . . . . .	91
13.2 Projekt . . . . .	91
13.3 Bremse . . . . .	91
13.4 Externe Achsen . . . . .	91
13.5 Ein-/Ausgänge . . . . .	91
13.6 Programm . . . . .	91
13.7 Mobile Plattform . . . . .	91
13.8 Portal . . . . .	91
13.9 Referenzierung . . . . .	92
13.10 Werkzeug . . . . .	92
13.11 Benutzerkoordinatensysteme . . . . .	93
13.12 Virtuelle Box . . . . .	93
13.13 Schnittstellen . . . . .	94
13.13.1 SPS-Schnittstelle . . . . .	94
13.13.2 Programmwahl über digitale Eingänge . . . . .	95
13.13.3 Modbus . . . . .	95
13.13.4 CRI-Schnittstelle . . . . .	96
13.13.5 Positionsschnittstelle . . . . .	96
13.13.6 App-Schnittstelle . . . . .	97
13.13.7 Kameraschnittstelle . . . . .	97
13.13.8 Netzwerk . . . . .	102
13.13.9 Cloud . . . . .	103
13.13.10 Gamepad . . . . .	104
13.13.11 Unterspannungsversorgung (USV/UPS) . . . . .	105
13.14 Zugriff auf Konfigurationsdateien . . . . .	105
13.14.1 Zugriff über iRC . . . . .	105
13.14.2 SFTP-Zugriff . . . . .	105
13.14.3 Secure Shell Zugriff . . . . .	107
13.15 Weitergehende Konfiguration . . . . .	108

---

<b>14 Modbus</b>	<b>109</b>
14.1 Konfiguration des Modbus-Servers . . . . .	109
14.2 TIA-Portal Bibliothek . . . . .	109
14.2.1 Anlegen des Roboter Datenbausteins . . . . .	109
14.2.2 Einfügen des Roboterkommunikations FB . . . . .	110
14.2.3 Datenzugriff . . . . .	110
14.3 Adresszuordnung . . . . .	111
14.3.1 1 Bit Lesezugriff (Coils und diskrete Eingänge) . . . . .	112
14.3.2 16 Bit Lesezugriff (Eingaberegister) . . . . .	114
14.3.3 16 Bit Lese- und Schreibzugriff (Halteregister) . . . . .	116
14.3.4 Zahlen- und Positionsvariablen . . . . .	117
14.3.5 Bedeutung der Aufzählungswerte . . . . .	118
<b>15 Wartung</b>	<b>121</b>
15.1 Reinigung . . . . .	121
<b>16 Fehlerbehebung</b>	<b>122</b>
16.1 Häufig gestellte Fragen . . . . .	122
16.2 Fehlercodes und Wege zur Behebung . . . . .	122
16.3 Testsoftware Module Control . . . . .	122
16.4 Support-Kontakt . . . . .	123

---

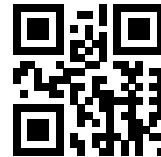
# 1 Einleitung

## 1.1 Kontakt

igus® GmbH  
Spicher Str. 1a  
D-51147 Köln

Tel.: +49(0)2203 / 96498-255  
E-Mail: [ww-robot-control@igus.net](mailto:ww-robot-control@igus.net)

Internet: [www.igus.de](http://www.igus.de)



## 1.2 Bestimmungsgemäße Verwendung

Die bestimmungsgemäße Verwendung des Produktes definiert sich durch die Verwendungen innerhalb der definierten Grenzen, aus den technischen Daten. Insbesondere zu beachten sind hierbei die zulässigen elektrischen Kenngrößen, sowie die definierten zulässigen Umgebungsbedingungen. Diese sind im weiteren Verlauf der Anleitung genauer spezifiziert.

Die bestimmungsgemäße Verwendung für dieses Produkt finden Sie im folgenden Abschnitt 3.

## 1.3 Zielgruppe und Qualifikation

Das Produkt und diese Dokumentation richten sich an technisch geschulte Fachkräfte wie:

- Entwicklungsingenieure
- Anlagenkonstrukteure
- Monteure/Servicekräfte
- Applikationsingenieure

Die Installation, Inbetriebnahme, sowie der Betrieb ist nur durch Fachkräfte erlaubt. Dies sind Personen, welche alle nachfolgenden Anforderungen erfüllen.

- eine entsprechende Ausbildung und Erfahrung im Umgang mit Motoren und deren Steuerung haben
- den Inhalt dieses technischen Handbuchs kennen und verstehen
- die geltenden Vorschriften kennen

## 1.4 Verwendete Symbole

Alle Hinweise in diesem Dokument folgen einer einheitlichen Form und sind gemäß nachfolgenden Klassen gegliedert.



**Der Hinweis WARNUNG macht den Leser auf mögliche gefährliche Situationen aufmerksam.**

Die Missachtung einer Warnung kann **möglicherweise** zu mittelschweren Verletzungen des Benutzers führen.

- Innerhalb einer Warnung beschreibt dies Möglichkeiten zur Vermeidung von Gefahren.



**Dieser Hinweis kennzeichnet mögliche Fehlbedienungen des Produktes.**

Die Missachtung dieses Hinweises kann die Funktionalität des Produktes einschränken.



In dieser Box befinden sich ergänzende Hinweise, sowie Tipps und Tricks.

## 1.5 Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

- RoHS-Richtlinie (2011/65/EU, 2015/863/EU)
- EMV-Richtlinie (2014/30/EU)

## 1.6 Vorschriften

Neben dem vorliegenden technischen Handbuch unterliegt der Betrieb, die Inbetriebnahme den geltenden Ortstypischen Vorschriften, wie z.B.:

- Unfallverhütungsvorschriften
- örtliche Vorschriften zur Arbeitssicherheit

## 2 Übersicht über das System

Die hier beschriebene Robotersteuerung ist Teil eines Robotersystems, welches aus vier Grundkomponenten besteht:

1. **Roboter:** der mechanische Roboterarm, Portal, Deltaroboter, Scara Roboter, etc ...
2. **Robotersteuerung:** modulare Robotersteuerung bestehend aus u.a. Embedded Computer, Schrittmotortreibermodule zum Ansteuern der Roboterachsen und IO-Module.
3. **Robotersteuerungssoftware:** Steuerungssoftware zum Bewegen des Roboters und dem Ausführen von Roboterprogrammen.
4. **Programmierungsumgebung:** Grafische Software zum Einrichten von Roboterprogrammen

Die Robotersteuerung dient zur Steuerung der Bewegung der Motoren des Roboters. Die Robotersteuerungen sind mit 48V Motorspannung oder 24V Motorspannung verfügbar. Die Schrittmotortreibermodule sind für bipolare Schrittmotoren unterschiedlicher Größe und konfigurierbar, bzw. vorkonfiguriert. Zur genauen Positionierung und Regelung werden im Normalfall Quadraturencodersignale eingelesen, die die tatsächliche Motor-, bzw. Achsposition in die Schrittmotortreibermodule rückkoppeln. Zudem werden unterschiedliche Referenzierungsmethoden und Bewegungen unterstützt, um nach einem Kaltstart des Robotersystems die "Ist-Position" des Roboters an die Software zu übermitteln. Unterschiedliche inverse Kinematiken werden in der Software unterstützt, so dass z.B. kontrollierte Linearbewegungen eines Roboterarms genauso einfach zu programmieren sind wie die eines Portalroboters.

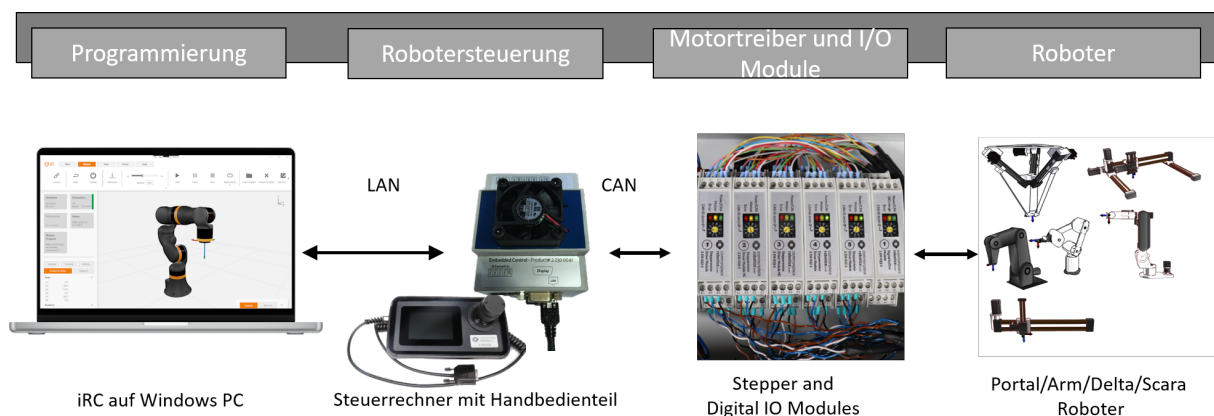


Abbildung 1: **Robotersystem** - Roboter, Robotersteuerung mit integriertem Computer und optionalem Handbediengerät. Separater per LAN mit der Robotersteuerung verbundener Windows PC mit iRC Steuerungssoftware und Programmierungsumgebung. Der Windows PC ist zur Programmierung des Roboters erforderlich. Auch ohne Windows PC lassen sich Roboterprogramme ausführen, da die Robotersteuerung den integrierten Computer als Steuerrechner verwendet.

### 2.1 Unterstützte Kinematiken

Die Steuerungshardware und Software unterstützen eine Vielzahl von Kinematiken, die eine kontrollierte Bewegung des Roboters erlauben. Vorkonfiguriert sind folgende inverse Kinematiken.

- Einzelachsen
- 2-Achs-Deltaroboter
- 3-Achs-Deltaroboter
- 4-Achs-Deltaroboter (3-Achs + Rotation)
- 2-Achs-Portalroboter (X und Y Achse)
- 2-Achs-Portalroboter (Y und Z Achse)
- 3-Achs-Portalroboter (X, Y und Z Achse)
- 3-Achs-Roboterarme
- 4-Achs-Roboterarme
- 5-Achs-Roboterarme
- 6-Achs-Roboterarme
- 3-Achs-SCARA-Roboter
- 4-Achs-SCARA-Roboter

Eine vollständige Liste von unterstützten Robotermodellen findet sich im "Roboter Simulieren" Bereich der iRC . Zudem können in iRC bis zu drei kinematisch unabhängige Zusatzachsen konfiguriert werden.

## 2.2 Spezifikationen

<b>Modulare Steuerung</b>	<b>Eigenschaft</b>
Stromversorgung	24V/5A und entweder 24V/10A oder 48V/10A
Typ	Hutschienenmodule mit 5 poligem Busverbinder
Kommunikation mit Steuerrechner	CAN
Schrittmotor Modul	Zum Betrieb eines bipolaren Schrittmotors Quadraturencoder RS422 12-24V Referenzschalter Eingang
Digitales IO Modul	7 digitale Eingänge (Optokoppler) 7 digitale Ausgänge, Halbleiterrelais, max. 500 mA

Tabelle 2: Modulare Robotersteuerung - Komponenten

Die Eigenschaften des integrierten Computers können Tabelle 3 entnommen werden.

<b>Steuerrechner</b>	<b>Eigenschaft</b>
Typ	Single Board Computer mit Tochtermodul
Stromversorgung	24V/1A
Betriebssystem	Linux-basiert
Software	RobotControl Core Robotersteuerungssoftware
Schnittstellen	CAN Bus (Verbindung zu den Modulen) Ethernet (Verbindung zu Windows PC)

<b>Steuerrechner</b>	<b>Eigenschaft</b>
	RS232 Display Anschluss
	Digitale Eingänge für Zustimmschalter
	DC/DC Wandler 5V/4A

Tabelle 3: Integrierter Computer - Spezifikationen

### 2.3 Mechanische Abmessungen

Tabelle 4 listet die mechanischen Abmessungen der einzelnen Komponenten im Schaltschrank, sowie des Schaltschranks selber auf.

<b>Objekt</b>	<b>L x B x H (mm)</b>
Schaltschrank	600 x 300 x 155
Steuerungsmodul	100 x 22 x 92
Steuerrechner	90 x 72 x 75
Netzteil 24V/10A	120 x 87 x 125
Netzteil 24V/5A	120 x 40 x 125
Netzteil 48V/10A	120 x 87 x 140
Handbedienteil (mit Display)	205 x 110 x 110

Tabelle 4: Modulare Robotersteuerung - Dimensionen

### 3 Sicherheitshinweise



Bedienen Sie den Roboter vorsichtig!

Achten Sie bei der Bedienung eines Roboterarms oder der Inbetriebnahme einer Roboterzelle stets auf die persönliche Sicherheit der Benutzer und anderer Personen! Insbesondere dürfen sich keine Personen oder Hindernisse im Arbeitsbereich des Roboters befinden.

- In der Grundversion enthält das Robotersteuerungspaket keine sicherheitsrelevanten Funktionen. Je nach Anwendung müssen diese möglicherweise hinzugefügt werden. Siehe auch "CE-Kennzeichnung" unten.
- CE-Kennzeichnung: Der Roboterarm und die Robotersteuerung sind ein Teil eines Systems, das in seiner Gesamtheit risikobewertet werden und den geltenden Sicherheitsvorschriften entsprechen muss, um die persönliche Sicherheit zu gewährleisten. Je nach Ergebnis der Bewertung müssen weitere Sicherheitskomponenten integriert werden. Dies sind in der Regel Sicherheitsrelais und Türschalter. Verantwortlich ist der inbetriebnehmende Ingenieur des Systems.
- Die Robotersteuerung enthält ein 24V Netzteil mit bis zu 10A Ausgangsstrom, das selbst Netzspannung (120 V / 240 V) benötigt. Bitte überprüfen Sie das Etikett auf dem Netzteil. Nur qualifiziertes Personal darf das Netzteil an das Netz anschließen und in Betrieb nehmen.
- Arbeiten an der Roboterelektronik sollten nur von qualifiziertem Personal durchgeführt werden. Überprüfen Sie die Richtlinien für elektrostatische Entladung (ESD).
- Trennen Sie die Robotersteuerung immer vom Netz (120 V / 240 V), wenn Sie im Schaltschrank oder an der Elektronik arbeiten, die an die Robotersteuerung angeschlossen ist.
- KEIN Hot-Plugging! Dies kann zu dauerhaften Schäden an den Motormodulen führen. Installieren oder entfernen Sie keine Module oder Steckverbinder (z.B. Handbediengerät, Not-Aus-Schalter, DIO-Module oder externe Relais, Motoranschlüsse...), während die Spannungsversorgung eingeschaltet ist.
- Der Roboterarm muss auf einer robusten Oberfläche aufgestellt und verschraubt oder anderweitig gesichert werden.
- Verwenden und lagern Sie das System nur in einer trockenen und sauberen Umgebung.
- Verwenden Sie das System nur bei Raumtemperatur (15° bis 32°C).
- Die Belüftung des Systems muss ungehindert arbeiten können, um einen ausreichenden Luftstrom zur Kühlung der Schrittmotormodule zu gewährleisten. Neben dem Lüfter der Robotersteuerung müssen mindestens 10 cm Platz vorhanden sein. Der Lüfter muss idealerweise nach oben oder zur Seite (reduzierter Wirkungsgrad) zeigen. Der Lüfter darf nicht nach unten zeigen.
- Sichern Sie wichtige Daten vor der Installation der iRC - igus Robot Control.

## 4 CE-Konformität

### 4.1 CE-Kennzeichnung

Die modulare Robotersteuerung ist mit einer CE-Kennzeichnung gemäß der RoHS-, Niederspannungs- und EMV-Richtlinien versehen. Die vollständige Maschine, d.h. die komplette Roboterzelle, ist nicht durch diese CE-Kennzeichnung abgedeckt. Die vollständige Maschine wird vom Integrator als Gesamtsystem dokumentiert, auf Risiken bewertet und mit einer CE-Kennzeichnung nach Maschinenrichtlinie versehen.



**Erstellung CE-Kennzeichnung notwendig!** Für den Betrieb der vollständigen Maschine muss eine CE-Kennzeichnung nach Maschinenrichtlinie erfolgen! Diese wird durch den Integrator oder Inbetriebnehmer vorgenommen!

### 4.2 Integration von Sicherheitskomponenten zur Herstellung der CE-Konformität

Die einfachste Möglichkeit die für CE-Konformität notwendige Personensicherheit herzustellen ist die Integration von Sicherheitskomponenten mit der entsprechenden PL- oder SIL-Bewertung:

- Sicherheitsrelais oder Sicherheitssteuerung
- NotAus- und Tür-Schalter
- Umhausung

Die modulare Robotersteuerung an sich stellt keine Sicherheitsfunktionen zur Verfügung. Sie trägt keine Bewertung nach PL oder SIL.

In der Steuerung sind zwei Netzteile vorgesehen, eines für die Logik (24V/5A), das andere für die Motorströme (24V/10A oder 48V/10A). Das erste Netzteil sollte kontinuierlich arbeiten, das zweite sollte im NotAus-Fall deaktiviert werden.



**Logikspannung nicht unterbrechen!** Bei NotAus führt die Unterbrechung der Motorspannung zu einem Stop der Motoren. Die Logik-Stromversorgung für die Integrierte Steuerung darf nicht unterbrochen werden. So kann das Programm nach einem Wiederfreigeben des NotAus ohne Hochfahren oder Referenzierung weitergeführt oder neu gestartet werden. Weitere Details finden Sie im Schaltplan der Steuerung.

Für die Unterbrechung der Motorspannung gibt es zwei Möglichkeiten:

- Auslieferungszustand: AC-seitig, also vor dem Netzteil.
- Optional: DC-seitig, also im 24 bzw. 48V-Kreis. Hier müssen Schütze verwendet werden die für hohe DC-Ströme >10A ausgelegt sind. Weiterhin muss zwingend eine Strombegrenzung verwendet werden um zu hohe Ströme beim Laden der Kondensatoren zu verhindern

## 5 Anforderungen

### 5.1 Umgebungsbedingungen

Umgebungsbedingung	Wert
Schutzklasse	IP20
Umgebungstemperatur (Betrieb)	+10...+32°C
Umgebungstemperatur (Lagerung)	-10...+85°C
Luftfeuchtigkeit (nicht kondensierend)	0...90%
Aufstellhöhe über NN (ohne Leistungsbeschränkung)	1500m

Tabelle 5: Umgebungsbedingungen



#### **Kondensation vermeiden!**

Betrieb oder Lagerung außerhalb der spezifizierten Umgebungsbedingungen kann zu Kondensation führen.

- Kondensation kann zu Kurzschlüssen und Zerstörung des Gerätes führen.
- Vermeiden Sie schnelle Temperaturwechsel.
- Bei Lagerung in kalter Umgebung: Akklimatisierung vor Inbetriebnahme (mindestens 2 Stunden bei Raumtemperatur).
- Relative Luftfeuchtigkeit muss stets unter 90% bleiben (nicht kondensierend).



#### **Aggressive Umgebungen nicht zulässig!**

Der Motorcontroller ist nicht für den Einsatz in chemisch belasteten oder aggressiven Umgebungen vorgesehen.

- Keine Verwendung in Umgebungen mit korrosiven Gasen oder Dämpfen.
- Keine Verwendung in Umgebungen mit starker Staubbelastung.
- Keine Verwendung in explosionsgefährdeten Bereichen.
- Bei besonderen Umgebungsanforderungen kontaktieren Sie den Hersteller.

### 5.2 Softwareanforderungen

Zur Verwendung von iRC wird ein Computer mit folgenden Eigenschaften benötigt:

- PC (min. Intel i5 CPU) mit Windows 10 bzw. 11 (64 Bit)
- .NET Framework 4.8.1 oder neuer
- mindestens 500MB freier Speicherplatz
- Grafikkarte (integriert oder dediziert)
  - OpenGL 3.1 oder neuer

- Herstellertreiber (der Standardtreiber von Microsoft wird nicht unterstützt)
- einen freien Ethernet-Port

## 6 Schnellstart-Anleitung

### 6.1 Einrichten und Anschließen



#### **Vor Beginn der Arbeiten**

Zur Vermeidung von Verletzungen, sowie Beschädigungen der Komponenten beachten Sie nachfolgende Hinweise:

- Befolgen Sie die Sicherheitshinweise in Abschnitt 3.
- Trennen Sie den Roboter, bzw. die Steuerung vom Netz. Führen Sie niemals Arbeiten an unter Spannung stehenden Teilen durch. Arbeiten an Schaltschränken sind nur von Elektrofachkräften durchzuführen.
- Kein Hot-Plugging! Vor dem einstecken oder trennen von Modulen/Steckern/Elektrischen Verbindungen, trennen Sie die Steuerung/den Roboter vom Stromnetz.
- Sorgen Sie für einen sicheren Stand des Roboters und der Steuerung.
- Beachten Sie die Anforderungen an die Umgebung 5.1.
- Während der Bewegungen des Roboters dürfen sich keine Personen im Arbeitsbereich des Roboters befinden.

Zum Aufbau und Inbetriebnahme des Roboters gehen Sie in der nachfolgenden Reihenfolge vor:

1. Stellen Sie sicher, dass die Steuerung vom Stromnetz getrennt ist: Ziehen Sie den Netzstecker.
2. Montieren Sie den Roboter auf einer geeigneten Basis. Stellen Sie sicher, dass die Kabel nicht gespannt sind.
3. Führen Sie die Roboterkabel durch das große kreisförmige Loch in den Schaltschrank ein und stecken Sie sie an die Schrittmotormodule. Jeder Motor mit dem dazugehörigen Referenzsensor ist über 4 Steckverbinder mit seinem Schrittmotormodul verbunden. Alle Anschlüsse sind beschriftet und kodiert, um diesen Prozess zu unterstützen. Folgende Kabel gehören jeweils zu einem Motor die Zugehörigkeit zwischen Motor und Kabelsatz ist ebenfalls auf den Steckern numerisch gekennzeichnet:
  - Motorkabel (mit der Bezeichnung Motor)
  - Encoderkabel (2 Anschlüsse mit der Bezeichnung ENC-1 und ENC-2)
  - Referenzsensor (mit der Bezeichnung End-Stop)
4. Verbinden Sie die Schirm und Masseadern der Encoder und Motorkabel, sofern vorhanden.
  - Motorkabel Masse: grün-gelb
  - Encoderschirmung (Masse): schwarz
5. Alle Masseadern der Motor und Encoderkabel werden in den Masseblock neben der Steuerung gesteckt.
6. Sichern Sie die Roboterkabel gegen Spannung, z.B. mit einem Kabelbinder an einem der Löcher im Schaltschrank. Falls vorhanden, stecken Sie das Kabel des Handbediengeräts ein und sichern Sie es über die Schraubverbindung.

7. Der Roboter ist nach Abschluss dieser Schritte bereit zum Einschalten.



Abbildung 2: Drei Motormodule mit je 4 Steckern pro Modul, die zu je einem Motor und dem dazugehörigen Referenzsensor führen. Auch im Bild: ein Digitales I/O Modul (Modul rechts), welches nicht verdrahtet ist.

## 6.2 Einschalten



**Gefahr von elektrischem Schlag!** Vergewissern Sie sich vor dem Inbetriebnehmen der Komponenten über die Ordnungsgemäße Verbindung aller Kabel und Schaltschrankkomponenten. Prüfen Sie alle Kabel auf einen festen Sitz und stellen Sie sicher, dass sich keine Losen Kabel im Schaltschrank befinden.

1. Verbinden Sie den Roboter über das beigegefügte Netzkabel mit Ihrer Stromversorgung.
2. Schalten Sie den Roboter mit dem Ein/Aus-Schalter am Schaltschrank ein.
3. Die grünen Leuchtdioden (LEDs) auf den Modulen leuchten nun, ebenso die meisten roten LEDs und möglicherweise auch einige der gelben LEDs.
4. Nachdem der Bootvorgang des Steuerrechners abgeschlossen ist (ca. 60s nach Einschalten der Steuerung), beginnen die grünen LEDs zu blinken. Dies zeigt Kommunikation mit den Modulen an. Nun ist die Robotersteuerung in Betrieb. Falls verfügbar, können Sie den Roboter jetzt über das optionale Handbediengerät bewegen.

## 6.3 Verbinden und Bewegen des Roboters

### 6.3.1 Vorbereitung mit dem integrierten Computer

Die Nachfolgenden Schritte stellen die Ethernet-Verbindung zwischen Robotersteuerung und Windows-PC her.



Abbildung 3: Integrierter Steuerrechner mit Ethernet-Kabel.

1. Verbinden Sie Ihren PC über ein Ethernet-Kabel mit der Robotersteuerung. Verwenden Sie den Ethernet-Anschluss, der sich direkt neben der USB-Buchse am integrierten Computer (siehe Abb. 3) der Robotersteuerung befindet.
2. Stellen Sie die IP-Adresse des PCs auf: statisch und 192.168.3.1 mit einer Subnetzmaske von 255.255.255.0. Anleitungen zur Änderung der IP-Adresse Ihres Computers finden Sie im Internet unter dem Stichwort "IP Adresse ändern Windows 10" (bzw. Windows 11).

### 6.3.2 Erstes Bewegen des Roboters

1. Installieren Sie die iRC Software auf Ihrem PC.
2. Starten Sie die iRC Software. Beim Start können Sie entweder einen beliebigen Roboter simulieren oder sich direkt mit Ihrem Roboter Verbinden. Drücken Sie "Automatisch verbinden", wenn der Roboter im Auslieferungszustand ist oder geben Sie eine konkrete IP-Adresse ein, falls diese auf dem Roboter geändert wurde (siehe Abb. 4).
3. Sie können den Roboter jetzt aktivieren, indem Sie nachfolgenden Schaltflächen in der angegebenen Reihenfolge drücken (siehe auch Abb. 5):
  - 3.1 "Verbinden"
  - 3.2 "Zurücksetzen"
  - 3.3 "Aktivieren"
4. Jetzt sollte die Status-LED-Leuchte links in der iRC grün sein und der Status "Kein Fehler" anzeigen.

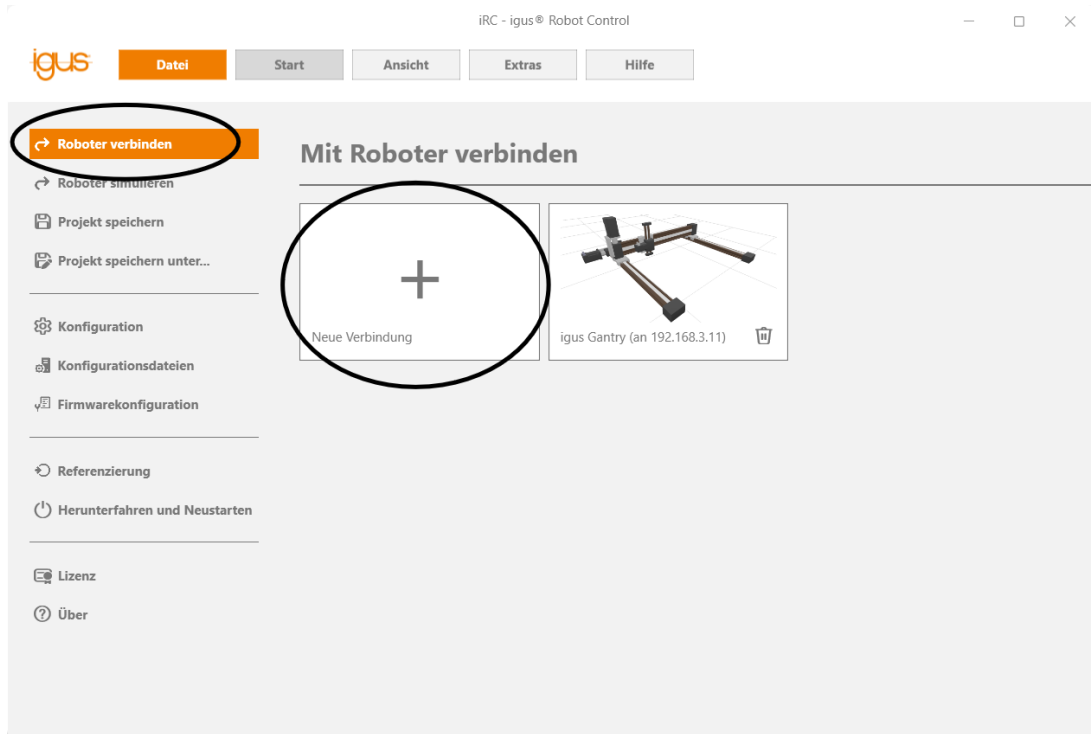


Abbildung 4: Verbinden mit dem Roboter

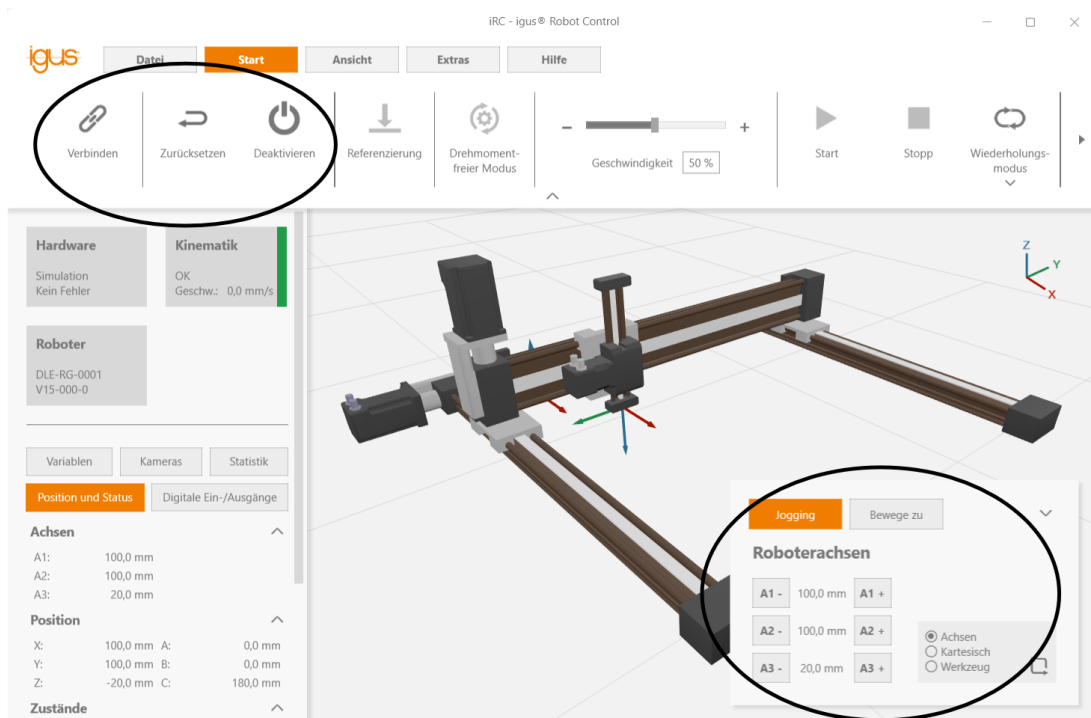


Abbildung 5: Jog-Befehle

- 
5. Sie können nun die Gelenke des Roboters mit Hilfe der Schaltflächen im Bereich "Jogging" bewegen (siehe Abb. 5).



**Kartesische Bewegungen und Programme**

Für die Verwendung von kartesischen Befehlen und das Abspielen von Roboterprogrammen ist eine Referenzierung des Roboters zwingend erforderlich.



**Problembehandlung**

Sollten beim Verbinden oder Aktivieren Probleme auftreten, schauen Sie bitte im Kapitel "Fehlerbehebung" 16.

## 7 Installation

Die detaillierte Anschlussbelegung entnehmen Sie bitte dem Schaltplan der Robotersteuerung. Die Anschlussbelegungen in diesem Kapitel sind ausschließlich zur besseren Veranschaulichung gedacht. Konsultieren Sie grundsätzlich den Schaltplan für die Anschlussbelegung.

### 7.1 Übersicht

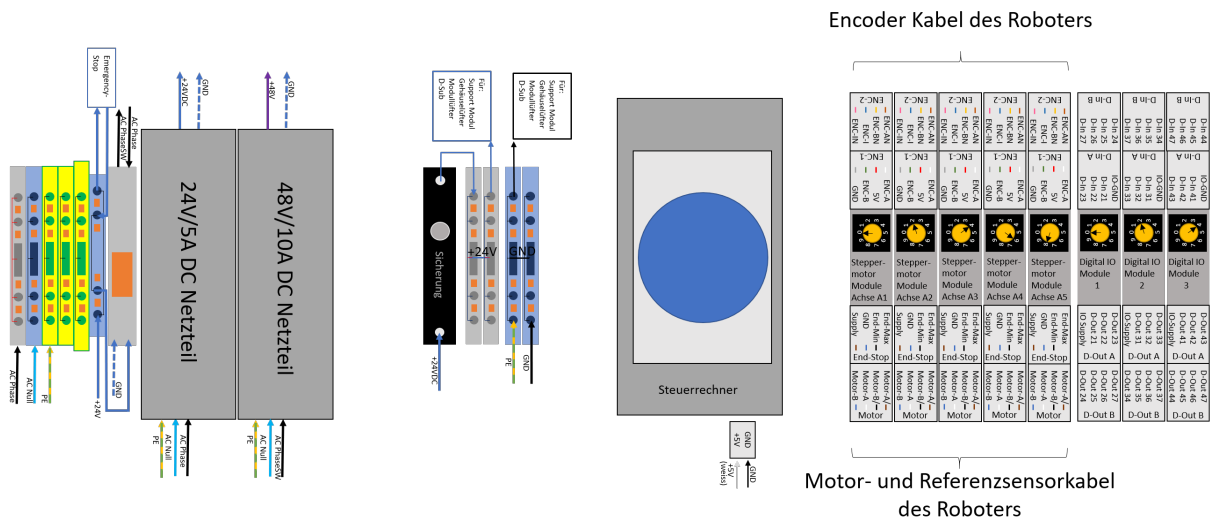


Abbildung 6: **Robotersteuerung:** Hier dargestellt mit 5 Stepper-Modulen und 3 DIO-Modulen. Zusätzlich sind optionale 24V/5A und 48V/10A Netzteile für Logik- und Motorspannung dargestellt. Detaillierte Anschlussbelegung im separaten Schaltplan.

### 7.2 Motormodul

Jedes Schrittmotormodul treibt einen bipolaren Schrittmotor mit Motorencoder an. Die Encodersignale werden von einem RS422-Baustein ausgewertet. Die Signale für jede Achse laufen über drei Kabel: Motorkabel, Encoderkabel und Referenzschalterkabel. Das Motorkabel wird an einen Stecker mit der Bezeichnung "Motor", das Encoderkabel an zwei Stecker mit der Bezeichnung "ENC-1" und "ENC-2" und das Referenzschalterkabel an den Stecker mit der Bezeichnung "End-Stop" angeschlossen.

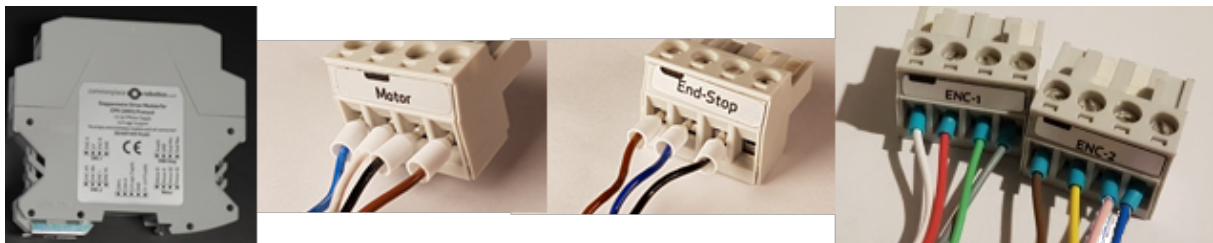


Abbildung 7: Motor Modul mit Steckverbindern



Die hier beschriebene Zuweisung bezieht sich auf die normale Konfiguration für Stepper-Motoren (Nema 17-24). Andere Motoren (Nema 34, BLDC Closed-Loop) können andere Aderfarben oder Pin-Belegungen haben. Eine allgemeine Beschreibung finden sie hier:

[https://wiki.cpr-robots.com/index.php/Wiring\\_IRC\\_DIN\\_Rail\\_Modules](https://wiki.cpr-robots.com/index.php/Wiring_IRC_DIN_Rail_Modules)



Motorstecker ("Motor"): Verbindet einen bipolaren Schrittmotor:

- Pin 1 (links): B (blau)
- Pin 2: A (weiss)
- Pin 3: B/ (schwarz)
- Pin 4: A/ (braun)

Encoderstecker 1 (ENC-1): Verbindet einen Quadratur-Encoder mit einem RS422-Baustein.

- Pin 1 (links): A (weiss)
- Pin 2: 5 V DC (rot)
- Pin 3: B (grün)
- Pin 4: 0 V (grau)

Encoderstecker 2 (ENC-2): Verbindet einen Quadratur-Encoder mit einem RS422-Baustein.

- Pin 1 (links): A-N (braun)
- Pin 2: B-N (gelb)
- Pin 3: index (rose)
- Pin 4: index-N (blau)

Alle acht Adern (Encoder-Anschlüsse 1 und 2) müssen angeschlossen werden, um den Encoder auszulesen.

End-Stop-Stecker: Wird an einen End- oder Referenzschalter angeschlossen.

- Pin 1 (links): 24 V (braun)
- Pin 2: Masse (GND) (blau)
- Pin 3: Signal (schwarz)
- Pin 4: nicht verbinden



Bei einigen Portalrobotern sind Motoren am gegenüber liegenden Achsende montiert. Der Referenzschalter sollte, wenn nicht anders in der Aufbauanleitung beschrieben, trotzdem am Nullpunkt (wie in der Simulation gezeigt) montiert sein. Wenn beispielsweise der Motor der Y-Achse rechts montiert ist sollte der Referenzschalter links montiert sein. Das stellt sicher, dass der Nullpunkt und die eingestellten Achslimits korrekt gesetzt werden.

### 7.3 Digital-I/O-Modul

Das DIO-Modul bietet Ein- und Ausgangskanäle, z.B. zur Ansteuerung eines Greiferventils. Die Ausgänge können bis zu 500 mA schalten. Die Eingänge verwenden Optokoppler und sind kompatibel zu Eingangsspannungen zwischen 12 und 24 V.

Ein von den Ausgangsrelais geschalteter Stromkreis darf keine größeren Kondensatoren enthalten. Wenn der Strom 500 mA übersteigt, können die Halbleiterrelais beschädigt werden.

Aus Sicherheitsgründen sind die Ein- und Ausgänge des DIO-Moduls galvanisch getrennt. Das bedeutet, dass die Schaltkreise der Ein- und Ausgänge nicht mit den internen Schaltkreisen des Controllers verbunden sind. Es ist daher notwendig, eine Versorgungsspannung für die Ausgänge und eine Masseleitung für die Eingänge anzuschließen. Hierfür können die in der Robotersteuerung verfügbaren 24 V, aber auch eine unabhängige externe Spannungsquelle verwendet werden. In der Software sind die Ein- und Ausgänge des ersten DIO-Moduls mit den Nummern 21-27, des zweiten DIO-Moduls (falls eingebaut) mit den Nummern 31-37 und des dritten mit 41-47 nummeriert.

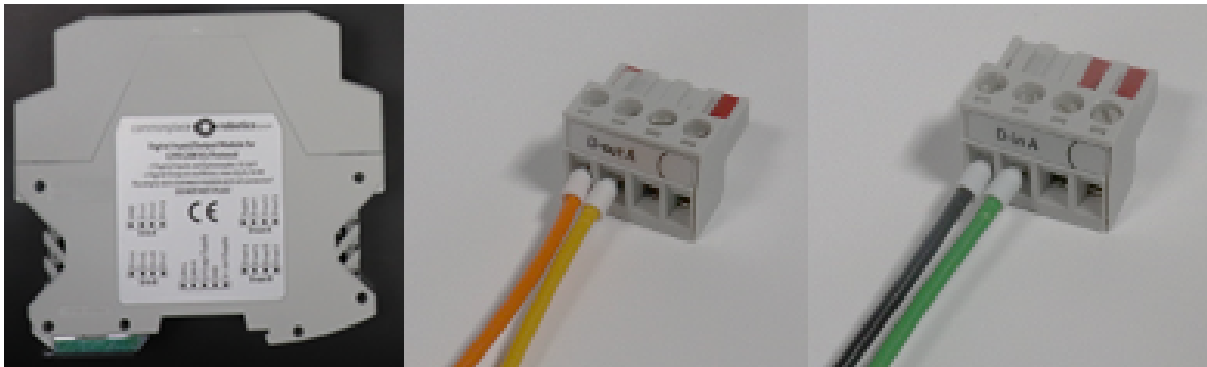


Abbildung 8: DIO Modul mit Steckverbindern

**Digital Out-Stecker A (D-out A):** Die Ausgangsrelais verbinden den Pin der Stromversorgung mit den entsprechenden Ausgangspins.

- Pin 1 (links): Eingangsspannung (für alle 7 Kanäle)
- Pin 2: D-Out-Kanal 1 (in der Software: Dout21)
- Pin 3: D-Out-Kanal 2 (in der Software: Dout22)
- Pin 4: D-Out-Kanal 3 (in der Software: Dout23)

**Digital Out-Stecker B (D-Out B):** Die D-out B-Pins sind (von links nach rechts) die Digital Out-Kanäle 4-7 (Bild nicht dargestellt).

**Digital In-Stecker A (D-In A):** Pin 1 von D-In A ist der entsprechende GND-Pin für alle Eingangspins.

- Pin 1 (links): Signal GND (für alle 7 Kanäle)
- Pin 2: D-In Kanal 1 (in der Software Din21)
- Pin 3: D-In-Kanal 2 (in der Software Din22)
- Pin 4: D-In-Kanal 3 (in der Software Din23)

**Digital In-Stecker B (D-in B):** Die D-in-B-Pins sind (von links nach rechts) die Digital-In-Kanäle 4-7 (Bild nicht dargestellt).

Im Abschnitt 12.1 finden Sie weitere Informationen zur Erweiterung der DIO und dem Anschluss von Aktoren und Sensoren.

## 7.4 Motorbremse

Um ein Absacken einer Achse zu verhindern, können elektromagnetische Bremsen verwendet werden. Die dazu notwendige Konfiguration wird in Abschnitt 12.3 vorgestellt.



Abbildung 9: Steuerrechner

## 7.5 Steuerrechner

Ein integrierter Computer ist mit der modularen Robotersteuerung kombiniert, so dass ein externer Computer nur für die Programmierung, nicht aber für den täglichen Betrieb erforderlich ist. Die Windows Software (iRC) kommuniziert grundsätzlich per Ethernet mit der Steuersoftware auf dem integrierten Computer (Linux). Die Steuersoftware kommuniziert per CAN mit den Modulen der Robotersteuerung. Abbildungen 9, 10 und 11 zeigen das Steuerungsmodul und dessen Anschlüsse.

### 7.5.1 Spannungsversorgung

Am Steckverbinder "IN Supply" des Steuerrechners (ganz rechts in Abbildung 10) wird angeschlossen (von links nach rechts):

- Pin1: +24V
- Pin2: GND

### 7.5.2 Busverbindung

Die CAN Verbindung wird über den mittleren, 5 poligen Steckverbinder "OUT-BUS" hergestellt (von links nach rechts):

- Pin1: CAN-L
- Pin2: CAN-H
- Pin3: +5V Versorgungsspannung für die Motormodule
- Pin4: Nicht verbinden



Abbildung 10: Bus- und Stromversorgungsanschluss. Der "+24V Out" Steckverbinder bleibt ungenutzt.

- Pin5: Nicht verbinden

Die Bus Verbindung wird an die Hutschienenmodule (Digital IO- und Motormodule) der Steuerung angeschlossen. Ein Supportmodul ist mit dieser Anschlussart nicht kompatibel, da es auch +5V auf den gleichen Bus speisen würde.

### 7.5.3 LAN-Anschluss



Abbildung 11: LAN-, USB- und Display-Anschluss.

Eine Verbindung zwischen Steuerrechner und Windows-PC oder SPS kann per Ethernet-Kabel hergestellt werden. Die IP des Steuerrechners ist 192.168.3.11, das Subnetz ist 255.255.255.0.

### 7.5.4 USB-Anschluss

Die USB-Buchsen können zum Anschluss von Kameras zur Arbeitsraumbeobachtung via CRI oder Cloud, sowie zur Kommunikation mit einer unterbrechungsfreien Stromversorgung (USV) verwendet werden.

### 7.5.5 Verbindung Handsteuergerät

Das Handsteuergerät (Touch-Display mit Joystick) wird am 9-poligen D-Sub Verbinder angeschlossen. Achtung! Die Verbindung ist proprietär. Ein Nullmodemkabel o.ä. kann hier nicht angeschlossen werden. Die Pin Belegung ist wie folgt (die Pins sind am Steckverbinder nummeriert):

- Pin 1: GND
- Pin 2: CAN-L
- Pin 3: RS232 TX
- Pin 4: nicht verbunden
- Pin 5: +24V Ausgang
- Pin 6: GND
- Pin 7: CAN-H
- Pin 8: RS232 RX
- Pin 9: GND

## 7.6 Handbediengerät



Abbildung 12: Handbediengerät (Touch Display und 3-Achsen Joystick)

Das Handbediengerät wird zur Steuerung des Roboters über den integrierten Computer verwendet. Der Anschluss des D-Sub-9-Steckers ist mit der D-Sub-9-Buchse des Schaltschranks zu verbinden. Die Pin Belegung ist des Handbediengerätes ist wie folgt:

- Pin 1: nicht verbunden
- Pin 2: nicht verbunden
- Pin 3: RS232 RX
- Pin 4: nicht verbunden
- Pin 5: +24V Eingang
- Pin 6: nicht verbunden
- Pin 7: RS232 TX
- Pin 8: nicht verbunden
- Pin 9: GND

## 8 Bewegen eines Roboters mit iRC

Die iRC - igus Robot Control ist eine Steuerungs- und Programmierumgebung für Roboter. Die 3D-Benutzeroberfläche hilft dabei, den Roboter schnell einsatzfähig zu machen. Durch den modularen Aufbau können verschiedene Kinematiken und Motortreiber gesteuert werden.

### 8.1 Die grafische Benutzeroberfläche

Dieser Abschnitt erklärt die iRC Software. Auch ohne einen angeschlossenen Roboter können alle Schritte simuliert werden. In Abschnitt 8.2 wird dann der reale Roboter angeschlossen und bewegt. Die Programmierumgebung iRC ermöglicht die Steuerung und Programmierung des Roboters. Sie können sowohl online als auch offline arbeiten, d.h. mit dem angeschlossenen Roboter oder in der Simulation.

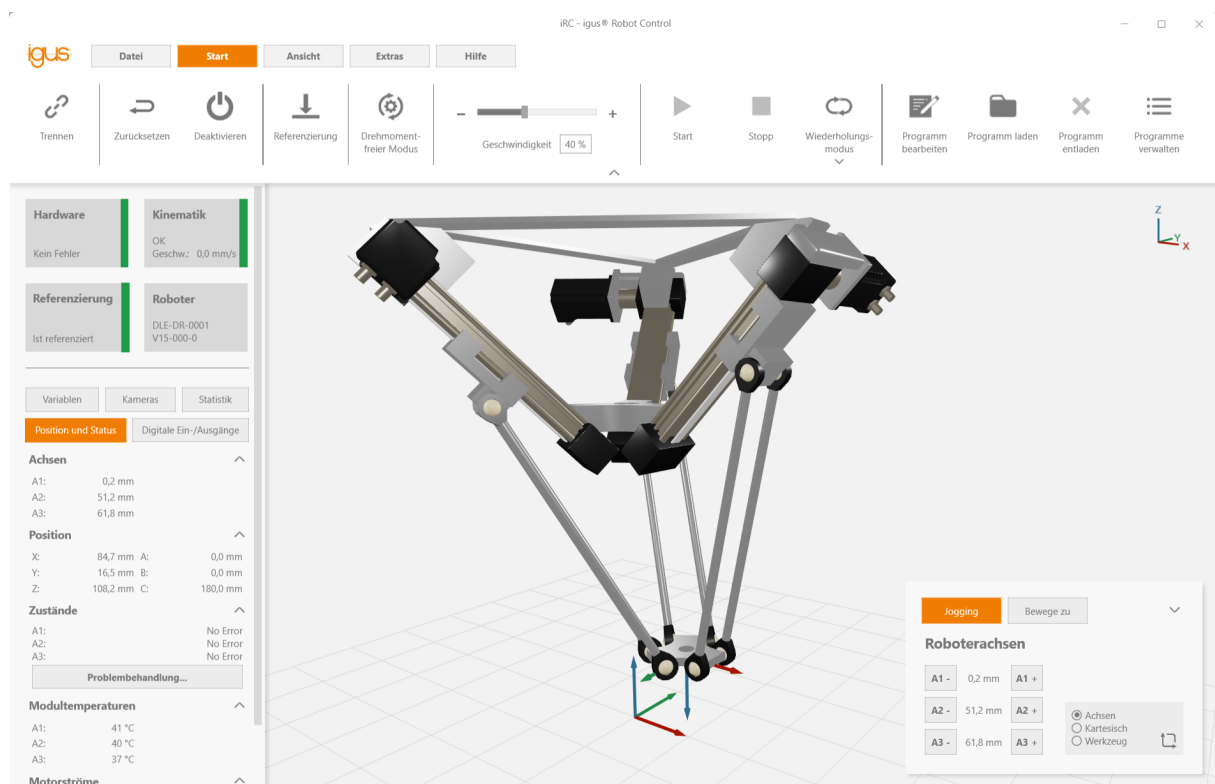


Abbildung 13: Benutzeroberfläche der iRC - igus Robot Control

In der linken oberen Ecke bieten die Registerkarten "Datei", "Start", "Ansicht", "Extras" und "Hilfe" Zugang zu den Hauptfunktionen. Auf der linken Seite werden Informationen über den aktuellen Zustand des physischen Roboters angezeigt. Ein Klick auf die "Hardware"-Kachel öffnet ein Fenster mit zusätzlichen Informationen zu dem Roboterstatus. Einstellungen zur Konfiguration des Projekts finden Sie im Register "Datei" (s. Abb. 13).

In der unteren rechten Ecke finden Sie den "Jogging"-Bereich, mit Tasten zur Bewegung des Roboters. Hier lässt sich auch der Bewegungsmodus ("Achsen" oder "Kartesisch") für direkte Bewegung anpassen. Zudem lässt sich über den "Bewege zu"-Bereich eine konkrete Position oder Pose anfahren.

Auf der linken Seite finden Sie zudem folgende Tabs:

- "Variablen": Zeigt die aktuellen Werte der Programmvariablen an.
- "Kameras": Bilder der angeschlossenen Kameras und erkannte Objektpositionen.
- "Statistik": Statistik über das System und das laufende Roboterprogramm.
- "Position & Status": Zeigt die Achswerte, die kartesische Position und weitere Informationen an.
- "Digitale Ein-/Ausgänge": Anzeigen und Einstellen der DIO-Schnittstellen der Robotersteuerung.

### 8.1.1 Simulieren eines Roboters

Um einen Roboter zu simulieren klicken Sie oben links auf "Datei" → "Roboter simulieren". Es erscheint eine Liste von Roboterarten und verschiedener Roboter in diesen Kategorien. Unter der Liste können Sie auswählen ob die 24V oder 48V-Konfiguration eines Roboters simuliert werden soll, dies legt die Geschwindigkeit fest. Klicken Sie einen Roboter an um diesen zu simulieren.

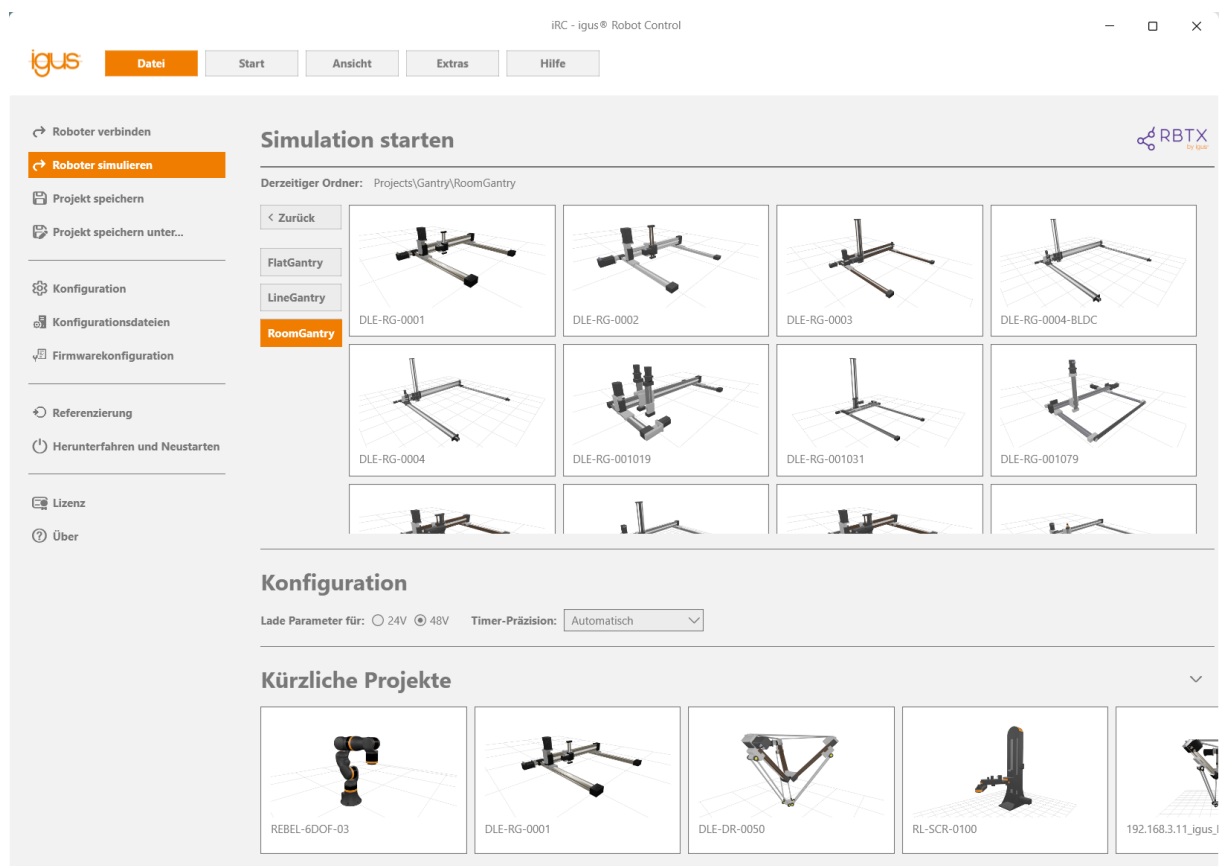


Abbildung 14: Auswahl des zu simulierenden Robotertyps über den Menüpunkt "Datei" → "Roboter simulieren"

Die Einträge der Liste werden Projekte genannt. Per Klick auf "Projekt speichern unter..." können Sie eine Kopie eines Roboters mit Ihren eigenen Einstellungen abspeichern.



Die Kategorie "MyRobots" enthält Kopien der Projekte Ihrer realen Roboter um diese zu simulieren. Die Einträge dort werden bei jedem Verbinden neu von dem Roboter heruntergeladen.

### 8.1.2 Verbinden eines realen Roboters

Um die iRC mit einem realen Roboter zu verbinden klicken Sie auf "Datei" → "Roboter verbinden". Dort finden Sie die Schaltfläche "Neue Verbindung" und eine Liste der bekannten Roboter. Klicken Sie auf "Neue Verbindung". Daraufhin erscheint die Schaltfläche "Automatisch verbinden", welche versucht zu einem Roboter an einer Standard-IP-Adresse zu verbinden. Falls die IP-Adresse Ihres Roboters geändert wurde tragen Sie diese rechts ein und klicken Sie auf "Verbinden". Nach einem kurzen Moment zeigt die iRC den Roboter in der 3D-Sicht an. Beim nächsten Verbinden können Sie ihn aus der Liste der bekannten Roboter wählen ohne die Adresse neu eingeben zu müssen.

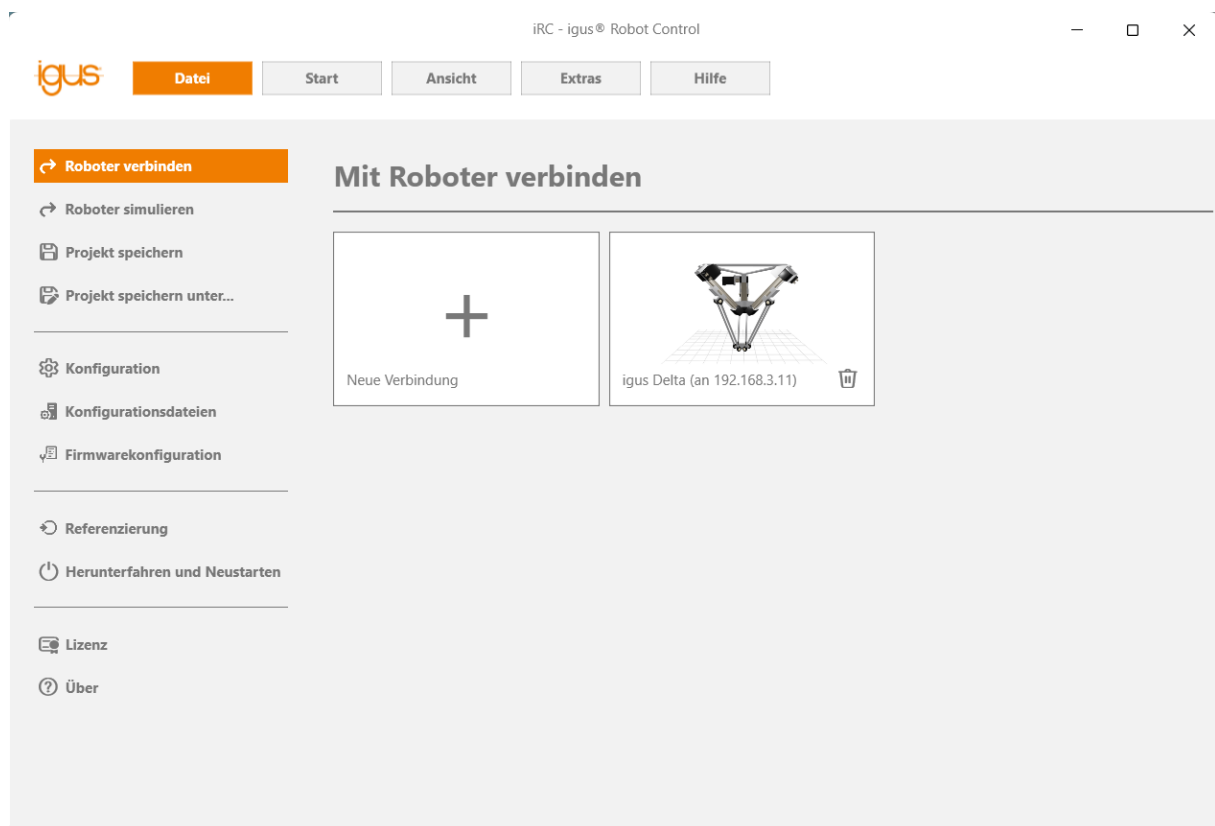


Abbildung 15: Hinzufügen oder Wählen einer Verbindung zu einem realen Roboter

Um die Verbindung zu trennen klicken Sie im "Start"-Tab links in der Leiste über der 3D-Ansicht auf "Trennen". Daraufhin wird eine Simulation des Roboters gestartet und die Schaltfläche wechselt zu "Verbinden". Ein weiterer Klick verbindet wieder zu dem vorher verbundenen realen Roboter.

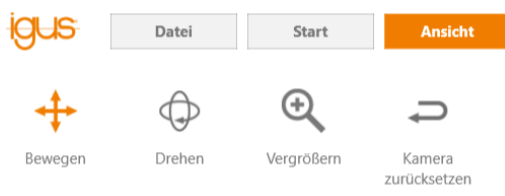
### 8.1.3 Navigation und Bewegen des Roboters in der 3D-Ansicht

Zur Navigation in der iRC - igus Robot Control 3D-Umgebung wird eine 3-Tasten-Maus empfohlen:



Abbildung 16: Bewegungsschweif-Einstellungen im Band über der 3D-Ansicht

- Linke Taste:
  - Auswahl von Symbolen und Funktionen im Menü.
  - Bewegen einer Roboterachse: Platzieren Sie den Cursor über ein Gelenk (es wird hervorgehoben), klicken Sie dann und bewegen Sie den Cursor auf und ab, während Sie die linke Maustaste gedrückt halten.
- Mittlere Taste/Mausrad:
  - Navigation in der Szene, um den Roboter zu drehen: Bewegen Sie den Cursor, während Sie die mittlere Maustaste gedrückt halten.
  - Mausrad drehen: Vergrößerung/Verkleinerung auf die aktuelle Cursorposition.
- Rechte Taste: Verschieben des Bildausschnitts



Die Funktion der linken Maustaste kann in der Registerkarte "Ansicht" geändert werden. Zur Auswahl stehen die Bewegungsoptionen "Bewegen", "Drehen" und "Vergrößern". "Kamera zurücksetzen" bringt Sie zurück in die Startansicht.

#### 8.1.4 Szenenelemente

Mit dem Menüpunkt "Szenenelemente" können verschiedene Elemente der 3D-Szene angepasst werden. Dazu gehören:

- "Bodenplatte": Definiert die Größe und Höhe des Rasters.
- "Lichter": Definiert die Lichtpunkte, die zur Ausleuchtung der Szene verwendet werden.
- "Roboter": Definiert die Position und Ausrichtung des Roboters in der Szene.

#### 8.1.5 Bewegungsschweif

Der Bewegungsschweif zeigt den Weg des Werkzeugs durch den Raum an. Durch die Farbe des Schweifes können zusätzliche Informationen wie z.B. die Orientierung oder Zustand eines digitalen Ausgangs dargestellt werden. Dies kann im Reiter "Ansicht" (s. Abb. 18) angepasst werden.

Mit der Schaltfläche "Bewegungsschweif" kann ausgewählt werden, ob der Bewegungsschweif am TCP des Roboters angezeigt werden soll oder nicht. "Schweif zurücksetzen" löscht den dargestellten Schweif.

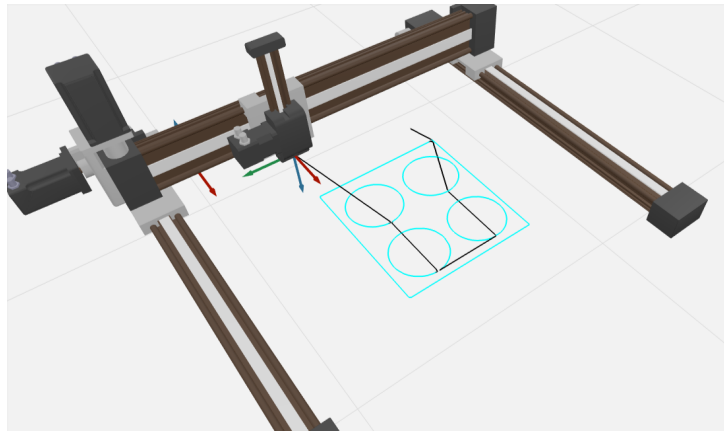


Abbildung 17: Darstellung eines digitalen Ausgangs im Bewegungsschweiß

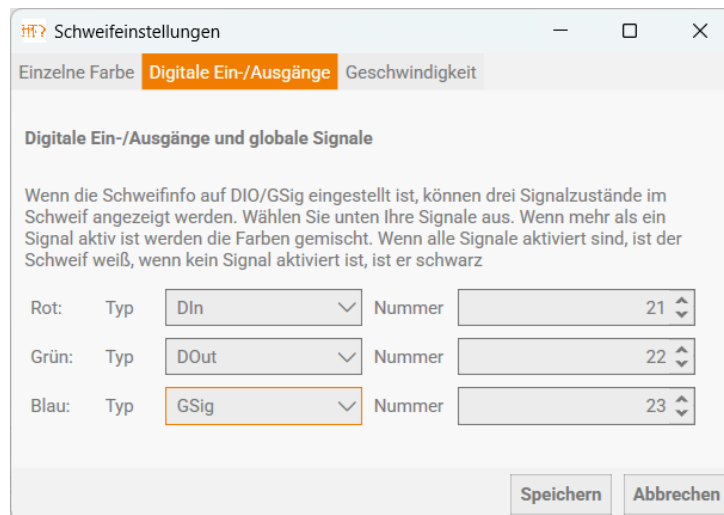


Abbildung 18: Erweiterte Konfiguration des Bewegungsschweißers

Daneben befinden sich Einstellungen zur Farbe des Schweißes. Im Auswahlménú "Schweifinfo" kann die anzuzeigende Information gewählt werden. Soll keine zusätzliche Information angezeigt werden kann eine Farbe unter "Schweiffarbe" festgelegt werden.

### DIOs/GSigs

Wenn dieser Punkt ausgewählt ist, wird mit der Schweißfarbe dargestellt, ob ein bestimmter Eingang, Ausgang oder Globales Signal gesetzt ist. Welche Signale angezeigt werden sollen kann in den Erweiterten Einstellungen festgelegt werden.

Es können dabei drei verschiedene Signale ausgewählt werden, die farblich dargestellt werden sollen. Ist kein Signal aktiv, ist der Trail weiß. Je nachdem welche Signale aktiv sind, ergibt sich die Trailfarbe. Ist beispielsweise DOut 21 und 22 aktiv, werden die Farben Rot und Gelb gemischt, sodass der Trail in orange dargestellt wird.

In der folgenden Tabelle sind alle möglichen Farben, sowie dem Zustand der Signale in diesem Beispiel dargestellt.

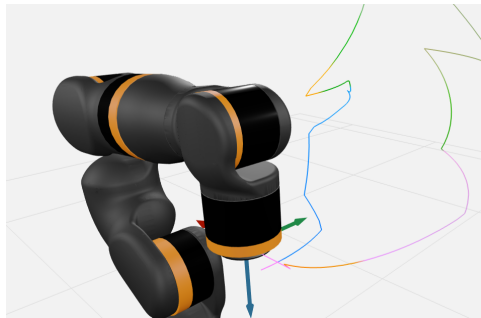


Abbildung 19: Darstellung der Werkzeugorientierung im Bewegungsschweif

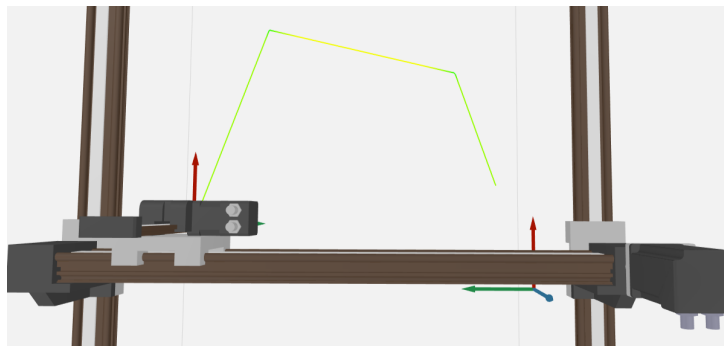


Abbildung 20: Darstellung der Werkzeuggeschwindigkeit im Bewegungsschweif

Farbe	IO 1	IO 2	IO 3
weiß			
rot	X		
gelb		X	
blau			X
orange	X	X	
violett	X		X
grün		X	X
schwarz	X	X	X

Tabelle 6: Farben des Bewegungsschweif abhängig vom Zustand der Ein-/Ausgänge

Da der Trail nur zyklisch in Form von Punkten dargestellt wird, die dann miteinander verbunden werden, gibt es zwischen zwei Punkten, wo beim ersten Punkt der Eingang aktiv ist und beim zweiten Punkt der Eingang inaktiv ist einen Übergangsbereich. Dieser Übergang wird fließend dargestellt.

#### **Orientierung**

Mit der Einstellung Orientierung kann die Orientierung des TCP dargestellt werden.

#### **Geschwindigkeit**

Mit der Einstellung Geschwindigkeit kann die Geschwindigkeit am TCP visualisiert werden. Dabei kann in den Einstellungen festgelegt werden, wie die Farbskala festgelegt werden soll.

Mit den Schaltflächen rechts kann die Länge des Bewegungsschweifs eingestellt werden.

## 8.2 Anschließen des Roboters

### 8.2.1 Verbindung zur Hardware

Der reale Roboter kann wie der Simulierte gesteuert werden, nur die Hardware muss zuerst durch Klicken auf die Symbole "Verbinden" und "Aktivieren" in der Schaltflächengruppe "Physikalischer Roboter" im Register "Bewegung" verbunden und aktiviert werden (siehe Abb. 21). Abhängig vom Robotertyp müssen die Achsen danach referenziert werden.



Abbildung 21: Schaltflächen für die Verbindung mit der Hardware, das Zurücksetzen von Fehlern und die Aktivierung der Motoren, die Referenzierung und "Status"-Anzeige.

1. "Verbinden": Stellen Sie die Verbindung zur Hardware her.
  - Hierüber wird eine Verbindung zum Roboter aufgebaut (meist per Ethernet, in einigen Fällen per USB-CAN-Adapter).
  - Die Status-Anzeige bei "Hardware" auf der linken Seite wechselt von grau zu rot oder grün.
  - Eventuell werden Fehlermeldungen in der "Hardware"-Kachel angezeigt.
2. "Zurücksetzen": Deaktiviert die Motoren und setzt die Fehler zurück.
  - Diese Taste wird zum Zurücksetzen der Fehlerspeicher der elektronischen Module der Robotersteuerung verwendet.
  - Die Achspositionen werden vom realen Roboter in die Simulationsumgebung übertragen. Die 3D-Visualisierung des Roboters sollte nun der aktuellen Position des realen Roboters entsprechen.



Dies muss bei jedem Fehler-Reset überprüft werden! Stimmen die Werte nicht überein, muss eine Referenzierung durchgeführt werden, wie in Abschnitt 8.3 beschrieben.

- Die "Status"-Anzeige wird rot. Die Fehlermeldungen werden gelöscht, nur "Motoren deaktiviert" bleibt bestehen. Wenn andere Fehlermeldungen angezeigt werden, versuchen Sie es erneut und folgen Sie den Anweisungen in der Roboterdokumentation.

### 3. "Aktivieren": Aktivierung der Motoren.

- Zuerst werden die Achsfehler zurückgesetzt, es muss daher nicht unbedingt vorher "Zurücksetzen" geklickt werden.
- Danach werden die Motoren aktiviert.
- Wenn keine Fehler aufgetreten sind ist die "Status"-Anzeige jetzt grün.

## 8.2.2 Den Roboter bewegen

Es ist jetzt möglich, den Roboter über die Jog-Tasten (im Bereich "Jogging" unten), mit einer Maus in der Benutzeroberfläche oder einem Gamepad zu bewegen, siehe Abschnitt 8.4.

## 8.3 Referenzieren des Roboters



- Nach dem Start muss der Roboter referenziert werden. Vor der Referenzierung können die Achsen des Roboters nur Bewegungen im Joint-Modus ausführen. Dadurch sollen Kollisionen während des unreferenzierten Roboterbetriebs vermieden werden.
- Erst nach der Referenzierung sind kartesische Bewegungen oder der Start eines Programms möglich.
- Der Status wird auf der linken Seite der iRC - igus Robot Control angezeigt.

Die Motormodule speichern die Position in einem EEPROM. Aufgrund der Schwerkraft oder anderer Kräfte können sich die Achsen jedoch bei ausgeschalteter Motorleistung bewegen. In diesem Fall melden die Motormodule nicht mehr die korrekte Position an die Software. Um die Position zwischen Software, Schrittmotormodul und Roboterachse zu synchronisieren, muss eine Referenzierung durchgeführt werden.

### 8.3.1 Schrittweise Anleitung der Referenzierung

1. Starten Sie die Robotersteuerung und der iRC - igus Robot Control.
2. Drücken Sie die Schaltflächen "Verbinden", "Zurücksetzen" und "Aktivieren" (Abb. 21).
3. Klicken Sie auf die Schaltfläche "Referenzierung" (s. Abb. 21) im Register "Start" (oder "Referenzierung" nach Klick auf "Datei"), um den Referenzierungsbereich zu öffnen.
4. Klicken Sie auf die Schaltflächen "Referenziere Achse", um mit der Referenzierung einer Achse zu beginnen, siehe Abb. 22. Mehrere Gelenke können die Referenzierung parallel durchführen.
5. Sie können auch auf "Alle Referenzieren" klicken, dann beginnen die Achsen in einer in der Projektdatei definierten Reihenfolge zu referenzieren.
6. Sobald alle Bewegungen ausgeführt sind und der Roboter wieder still steht, klicken Sie auf "Zurücksetzen" und "Aktivieren". Jetzt ist der Roboter voll funktionsfähig.

Mehr Informationen zur Reihenfolge bei "Alle Referenzieren" finden Sie auf unserem Wiki:

[https://wiki.cpr-robots.com/index.php/Referencing\\_sequence](https://wiki.cpr-robots.com/index.php/Referencing_sequence)

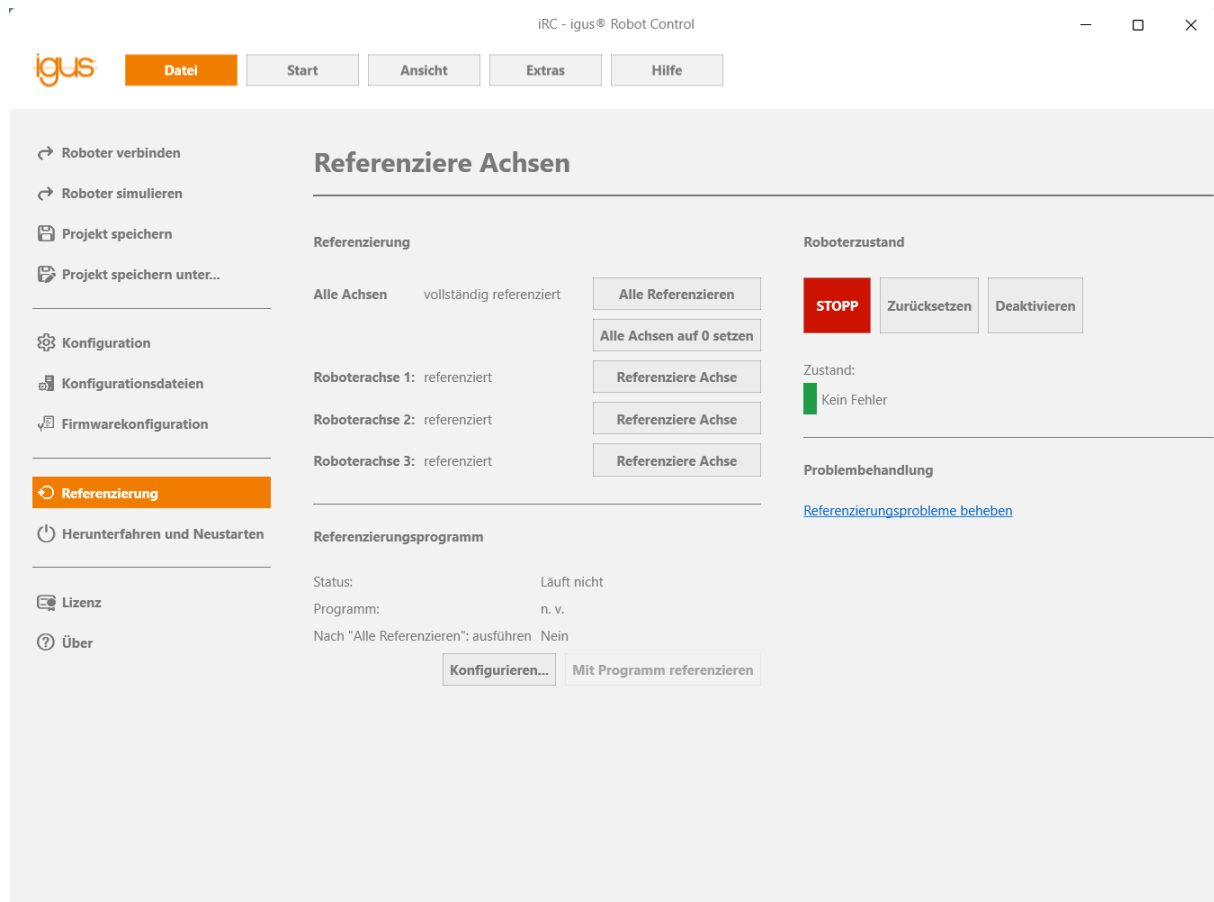


Abbildung 22: Referenzierung der Achsen.

### 8.3.2 Referenzierungsprogramm

Ein Referenzierungsprogramm kann definiert werden um die Präzision eines Roboters der über Absolutencoder referenziert zu verbessert. Für Roboter die Referenzschalter verwenden ist dies nicht relevant. Dabei werden zunächst alle Achsen normal referenziert, dann das Programm ausgeführt um an eine definierte Position zu fahren an der danach erneut referenziert wird.



Da die Achspositionen vor der Referenzierung nicht verlässlich sind ist es nicht möglich den Referenzierungsablauf zu programmieren um z.B. Hindernisse zu umfahren. Falls nötig kann die Reihenfolge der Achsen und die Verzögerungen zwischen dem Start der Referenzierung einzelner Achsen in der Roboterkonfigurationsdatei angepasst werden damit bestimmte Achsen zuerst aus dem Kollisionsbereich fahren können.

Mehr zu diesem Thema finden Sie im Abschnitt 13.9 sowie auf unserem Wiki:

<https://wiki.cpr-robots.com/index.php/Categorie:Referencing>



## 8.4 Bewegen des Roboters

Der Roboter kann manuell bewegt (oder "gejoggt") werden wenn kein Programm läuft. Dazu stehen folgende Möglichkeiten bereit:

- Softwaretasten
- Ziehen der Achsen im 3D-Bereich
- Gamepad

Im Register "Start" (s. Abb. 23) finden Sie die globale Geschwindigkeitseinstellung. Diese wirkt sich sowohl auf die jog-Geschwindigkeit als auch auf die Geschwindigkeit von Bewegungsanweisungen in Programmen aus. Es ist ratsam, die Geschwindigkeit zu reduzieren, wenn Sie den Roboter zum ersten Mal bewegen, um eine sichere Umgebung zu gewährleisten.

Im "Jogging" Bereich können Sie die Achsen direkt bewegen. Je nach ausgewähltem Bewegungsmodus bewegen die Schaltflächen entweder die Achsen oder den TCP des Roboters. Weitere Informationen zu den Bewegungsmodi finden Sie im Abschnitt 8.4.2.

Ein Gamepad können Sie im Register "Extras" verbinden. Sobald es verbunden ist, können Sie den Roboter mit den Tasten bewegen. Die Tastenbelegung kann im Konfigurationsbereich ("Extras" → "Konfiguration" → "Gamepad") angepasst werden.



Abbildung 23: Bedienfelder zum Bewegen des Roboters (blau markiert).

### 8.4.1 Gamepad

Ein Gamepad ist möglicherweise die intuitivste Art den Roboter zu bewegen. Die Abb. 24 zeigt die standardmäßige Tastenbelegung. Durch Drücken von "Gamepad verbinden" verbindet sich iRC mit einem Gamepad oder Joystick. Bei erfolgreicher Verbindung wird ein OK-Zeichen unter dem Symbol angezeigt.

Standardmäßig ist die folgende Tastenbelegung vorgegeben. Die Richtung der Achsen hängt vom Robotertyp ab.



Abbildung 24: Tastenbelegung des Gamepads

1.
  - oben: Bewegungsmodus ändern
  - unten: Geschwindigkeit senken
2.
  - oben: Tastenbelegung ändern: Umschalten zwischen X, Y, Z und A, B, C im kartesischen Bewegungsmodus oder A1, 2, 3 und A4, 5, 6 im Achs-Bewegungsmodus.
  - unten: Geschwindigkeit erhöhen
3.
  - oben: vorherige Anweisung im Programmeditor auswählen
  - unten: nächste Anweisung im Programmeditor auswählen
4.
  - oben: Nachbessern der aktuellen Anweisung im Programmeditor
  - rechts: Programmanweisung Achsbewegung hinzufügen
  - unten: Programmanweisung Linearbewegung hinzufügen
  - links: Ausgewählte Anweisung im Programmeditor entfernen



Die Belegung des Gamepads kann über die Projektkonfigurationsdatei geändert werden, siehe dazu:

[https://wiki.cpr-robots.com/index.php/Joystick\\_and\\_Gamepad](https://wiki.cpr-robots.com/index.php/Joystick_and_Gamepad)



#### 8.4.2 Softwaretasten

Software-Schaltflächen ermöglichen die Auswahl des Bewegungsmodus. Es stehen drei Modi zur Verfügung bei denen jeweils die Bewegungsgeschwindigkeit zwischen 0 und 100% verändert werden kann (Abb. 23):

- "Achse": Ein Klick auf A1 bis A6 bewegt die entsprechende Roboterachse (falls vorhanden). E1 - E3 bewegt die externen Gelenke. Dies können lineare oder Rotationsachsen sein (Abb. 25).
- "Basis": (kartesischer Modus) bewegt den Roboter in geraden Linien entlang der X-, Y- und Z-Achse des gerade gewählten Koordinatensystems (siehe Abschnitt 11.3).
- "Werkzeug": (kartesischer Modus) bewegt den Roboter in X, Y und Z des aktuellen Werkzeugkoordinatensystems.



Abbildung 25: Die Softwareschaltflächen für "Achse"-Bewegungen. In beiden kartesischen Modi wechseln die Schaltflächen auf X, Y, Z, A, B und C.

### 8.5 Nicht erreichbare Positionen und Singularitäten

Während im Achsmodus jede Achse in ihrem vollen Bewegungsbereich fahren kann (vorausgesetzt sie kollidiert nicht) sind im kartesischen Modus abhängig von der Kinematik nicht alle Positionen erreichbar. In einigen Fällen liegt dies an mathematischen Besonderheiten, beispielsweise können kleine Bewegungen bei ausgestrecktem Roboterarm zu schnellen Achsbewegungen oder einer nicht eindeutigen Ausrichtung führen - dies wird Singularität genannt.

Es sind unter anderem folgende Positionen nicht erreichbar:

- Außer Reichweite: Die Vorgabeposition ist zu weit entfernt oder kann von der Kinematik nicht erreicht werden. Bei Roboterarmen stoppt die Bewegung schon bevor der Arm vollständig ausgestreckt ist.
- Zentralachsensingularität: Tritt bei Roboterarmen auf wenn sich der Arm nahe der Zentralachse (Rotationsachse A1) befindet.
- Handgelenksingularität: Tritt bei 6-Achs-Roboterarmen auf wenn das Handgelenk (letzte 3 Achsen) zu weit ausgestreckt ist.

Wenn der Roboter im kartesischen Modus einer solchen Position nahe kommt wird die Bewegung gestoppt und ein eventuell laufendes Programm unterbrochen. Der Statusbereich links in der iRC zeigt den Kinematikfehler an.



Um Singularitäten zu vermeiden kann anstatt der kartesischen Bewegung (z.B. Liniaranweisung) die Achsbewegung verwendet werden. Dies ist besonders bei dynamischen Zielpositionen, z.B. von einer Kamera, sinnvoll.

Der Programmbefehl "Achsbewegung zu kart. Pos." ermöglicht es, die Zielposition in kartesischen Koordinaten anzugeben, die Bewegung wird aber im Achsmodus ausgeführt.

### 8.6 Starten von Roboterprogrammen

Ein Roboterprogramm kann wie folgt geladen und gestartet werden:

1. Zum Laden des Programms klicken Sie auf das Ordnersymbol "Programm laden" in der Registerkarte "Start" und wählen Sie ein Programm, z.B. "testReBeL.xml".

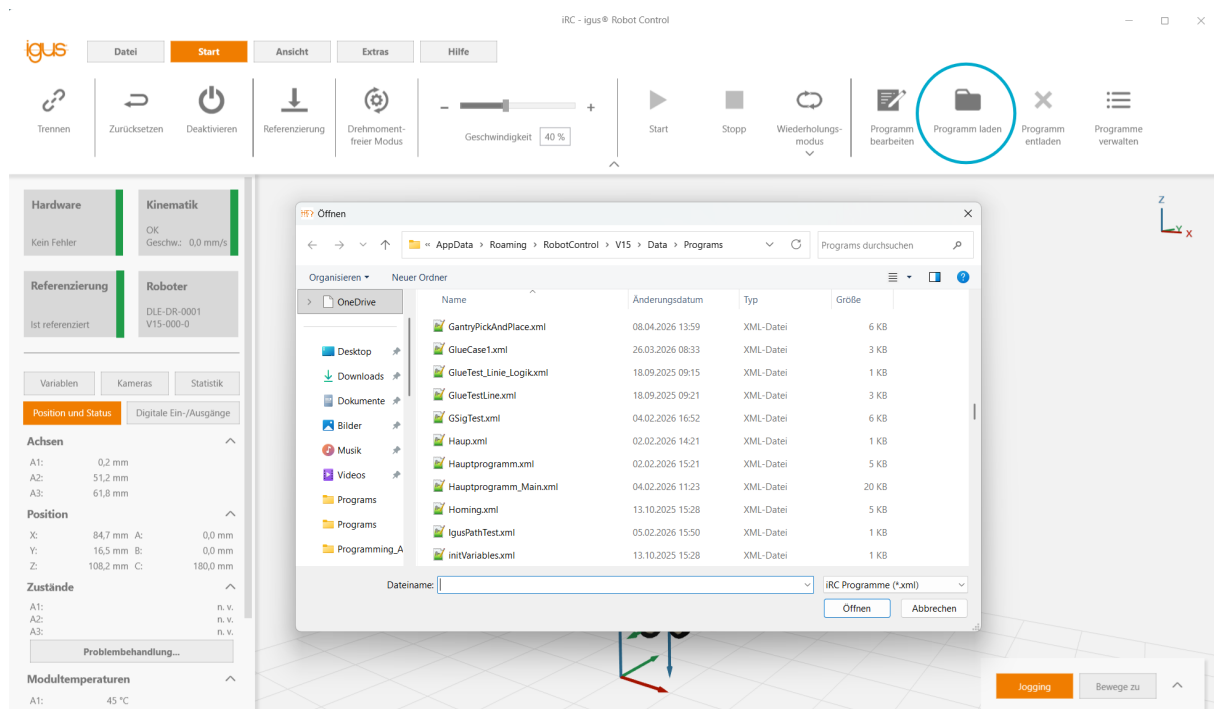
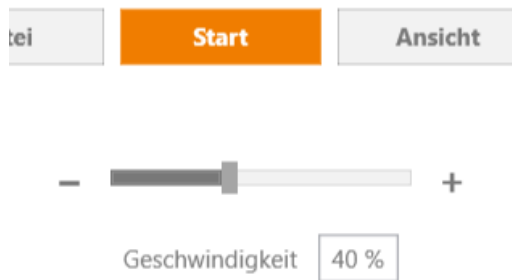


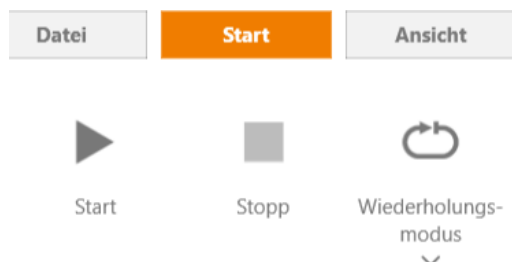
Abbildung 26: Laden eines Programms (blau markiert).

## 2. Stellen Sie die Grundgeschwindigkeit ein:



- Bevor Sie ein ungetestetes Programm starten, stellen Sie die Geschwindigkeit auf z.B. 20% ein.
- Seien Sie während des ersten vollständigen Programmablaufs besonders aufmerksam und halten Sie die Not-Aus-Taste bereit.

## 3. Starten Sie das Programm:



- Klicken Sie auf das Symbol "Start".

## 4. Das Programm anhalten oder unterbrechen:

- Nachdem Sie das Symbol "Pause" gedrückt haben, kann der Roboter mit dem Programm fortfahren, indem Sie erneut auf das Symbol "Start" klicken.
- Nach dem Drücken des "Stopp"-Symbols startet das Programm von vorne, wenn das "Start"-Symbol erneut angeklickt wird.
- Der "Wiederholungsmodus" kann auf drei verschiedene Werte eingestellt werden:
  - Einmal (das Programm stoppt nach einem einzigen Zyklus).
  - Wiederholen (das Programm stoppt nur durch "Pause" oder "Stopp").
  - Schritt (Dies ist nützlich für die Fehlersuche in einem Programm).

### 8.7 Digitale Ein- und Ausgänge

Der Zustand der Ein- und Ausgänge kann unter "Digitale Ein-/Ausgänge" überwacht werden. Sowohl Eingänge als auch Ausgänge können manuell aktiviert oder deaktiviert werden:

- Die Ausgänge können manuell eingestellt werden, wenn kein Programm läuft.
- Eingänge können nur in der Simulation gesetzt werden, wenn kein Roboter angeschlossen ist. Somit kann man die Reaktion von Programmen auf verschiedene Eingänge auch ohne die entsprechende Hardware testen.

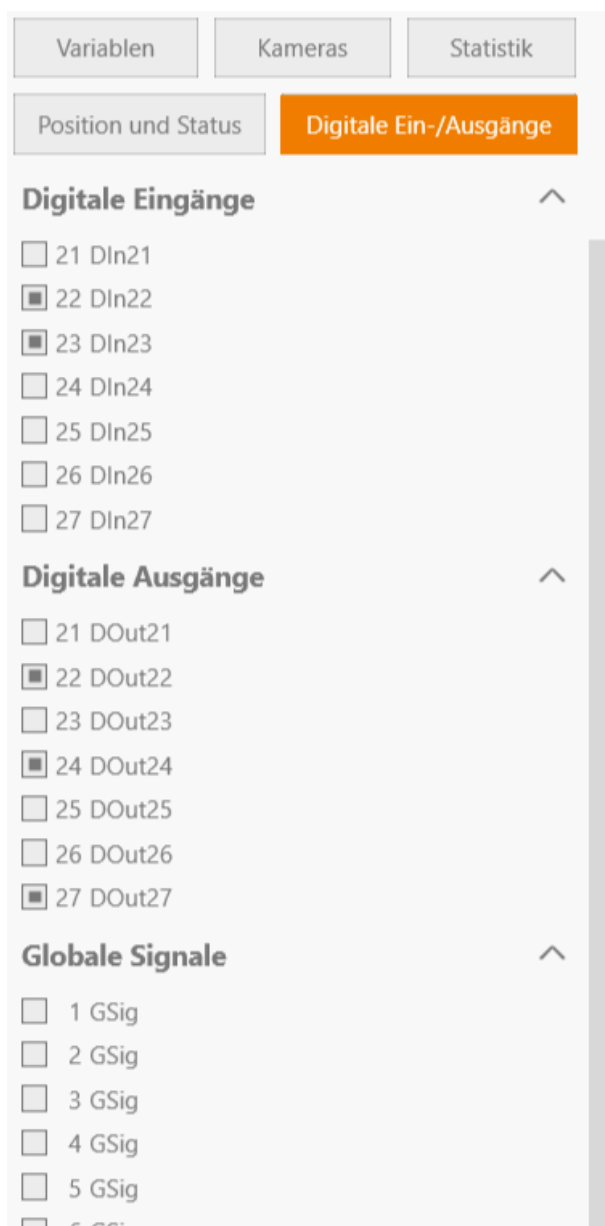


Abbildung 27: Digitale Ein-/Ausgänge der iRC - igus Robot Control.

Die Konfiguration der Ein- und Ausgänge ist in Abschnitt 12.1 beschrieben.

## 8.8 Software-Schnittstellen

Die Robotersteuerung stellt verschiedene Schnittstellen zur Verfügung:

- SPS-Schnittstelle zur Steuerung über die digitalen Ein- und Ausgänge. Insbesondere zum einfachen Starten und Stoppen von Programmen über eine SPS oder Taster.
- Modbus TCP-Schnittstelle zur Steuerung über eine SPS oder einen PC.
- CRI-Ethernet-Schnittstelle zur Steuerung und Konfiguration über eine SPS oder einen PC. Diese Schnittstelle bietet den größten Funktionsumfang, erfordert jedoch eine individuelle Imple-

mentation.

- ROS-Schnittstelle\* zum Betrieb des Roboters über das Robot Operating System ([www.ros.org](http://www.ros.org)).
- Schnittstelle für Objekterkennungskameras
- Cloud-Schnittstelle zur Überwachung des Roboterzustands
- App-Schnittstelle zur Erweiterung des Funktionsumfangs

Siehe Abschnitt 13.13 für die Konfiguration dieser Schnittstellen.



\*Die ROS-Schnittstelle befindet sich derzeit in der Entwicklung bei unserem Partner TruPhysics. Weitere Informationen finden Sie auf unserem Wiki:

[https://wiki.cpr-robots.com/index.php/ROS\\_packages](https://wiki.cpr-robots.com/index.php/ROS_packages)



### 8.8.1 App-Schnittstelle

Die App-Schnittstelle ermöglicht die Erweiterung der Robotersteuerung um neue Funktionen. Das Installieren von Apps wird in Abschnitt 13.13.6 beschrieben. Apps können die grafische Benutzerschnittstelle erweitern, in dem Fall finden Sie über der 3D-Ansicht zusätzliche Reiter. App-Funktionen können in den Ablauf von Roboterprogrammen eingebunden werden um z.B. auf Peripherie zuzugreifen oder komplexere Aktionen durchzuführen. Dies ist in Abschnitt 10.10 beschrieben. Apps und Informationen um eigene Apps zu erstellen finden Sie unter dem folgenden Link:

[https://wiki.cpr-robots.com/index.php/Apps\\_for\\_the\\_Robot\\_Control](https://wiki.cpr-robots.com/index.php/Apps_for_the_Robot_Control)



Da Apps beliebigen Code auf der Robotersteuerung ausführen können sollten Sie Apps sicherheitskritisch betrachten und nur Apps aus vertrauenswürdigen Quellen verwenden. Fehler können zu Schäden führen, böswillige Apps können bspw. Daten an Server Dritter senden oder das Netzwerk angreifen. Es ist empfehlenswert Roboter mit Apps nicht mit dem Firmennetzwerk oder dem Internet zu verbinden.

## 8.9 Aktualisieren der Software

Updates der iRC-Software finden Sie unter folgender Adresse: <https://wiki.cpr-robots.com/index.php/IgusRobotControl-DE>.



Erstellen Sie ein Backup da bei der Aktualisierung Dateien überschrieben werden können!  
Benennen Sie Ihren alten iRC-Ordner (z.B. C:\iRC-igusRobotControl) vor Beginn der Installation um. Auf diese Weise können Sie wieder zur alten Version zurückkehren.

Folgendes muss gegebenenfalls von der vorherigen Installation übernommen werden:

- Die erstellten Roboterprogramme
- Änderungen im Projekt oder in den Roboterkonfigurationen

## 9 Bewegen des Roboters per Handbediengerät

Um den Roboter ohne angeschlossenen Windows-PC zu betreiben, ist der integrierte Computer erforderlich. Er führt die RobotControl-Software als Robotersteuerungssoftware aus. Der integrierte Computer ermöglicht das Bewegen des Roboterarms (über das Handbediengerät) und die Wiedergabe von Roboterprogrammen. Um neue Programme einzurichten, ist er über Ethernet mit einem Windows-Computer verbunden, auf dem iRC läuft.

Dieser Abschnitt zeigt, wie der Roboter mit integriertem Computer und Handbediengerät betrieben wird.

Nachdem alle elektrischen Verbindungen zum Roboter hergestellt wurden, der Roboter eingeschaltet und der Not-Aus-Schalter freigegeben wurde, müssen die Fehler zuerst "zurückgesetzt" und dann der Roboter "aktiviert" werden.

Wenn Sie den Roboter bewegen, halten Sie immer eine Hand auf dem Not-Aus-Schalter, um zu verhindern, dass er unerwartet auf ein Objekt trifft, z.B. wenn er mit dem Tisch kollidiert.

### 9.1 Roboter Zurücksetzen und aktivieren



Abbildung 28: Drücken Sie den "Enable"-Knopf im oberen Menü (oben rechts) auf dem Touchscreen-Display, um zur Enable-Seite zu gelangen.

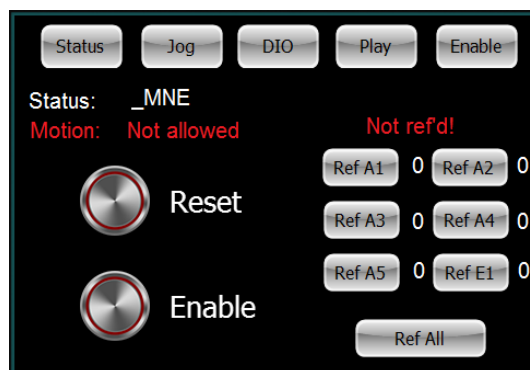


Abbildung 29: Drücken Sie jetzt "Reset": Der Status wechselt zu "MNE" (Motor Not Enabled)

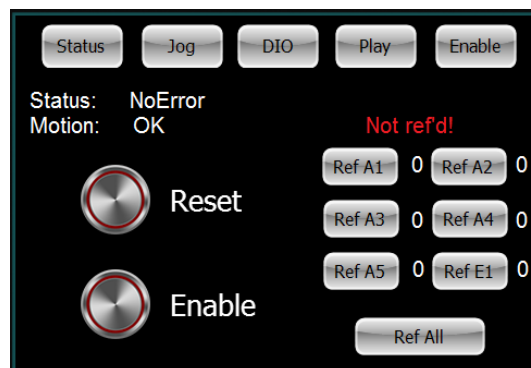


Abbildung 30: Drücken Sie "Enable": Der Status wechselt zu "NoError". "Not ref'd!" (in rot) bedeutet "nicht referenziert".

## 9.2 Bewegen des Roboters mit dem 3-Achsen-Joystick

Sobald der Roboter aktiviert ist, können die Achsen des Roboters bewegt werden.

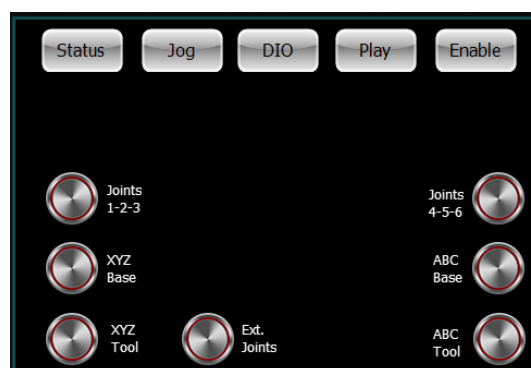


Abbildung 31: 1. Drücken Sie dazu die Taste "Jog" oben auf dem Display. 2. Drücken Sie "Joints 1-2-3". Dann bewegen und drehen Sie den Joystick. Sie können nun die Achsen 1, 2 und 3 bewegen. 3. Drücken Sie "Joints 4-5-6". Bewegen Sie nun die Achsen 4 und 5 durch Bewegen und Drehen des Joysticks.

## 9.3 Programmauswahl

Auf der Registerkarte Play können Sie ein Programm für den Roboter auswählen. In dieser Registerkarte können Sie auch den Override für die Robotergeschwindigkeit einstellen. Die Robotergeschwindigkeit ändert sich hierbei für Jog und Programmausführung.



Abbildung 32: Programmauswahl

## 9.4 Referenzierung

Um einen automatischen Programmablauf und Linearbewegungen ("XYZ Tool" / "XYZ Base") zu ermöglichen, muss der Roboter referenziert werden. Die Art der Referenzbewegung hängt vom Roboter- oder Encodertyp ab. Im Zweifel sollte jede Achse des Roboters wie oben beschrieben zunächst (Abb. 31) in die Nähe des Referenzsensors der Achse gefahren werden.



Abbildung 33: Navigieren Sie in per Button oben Rechts "Enable" in das Enable-Menü.

- Drücken Sie den runden "Reset"-Knopf gefolgt vom runden "Enable"-Knopf: Der Status wechselt zu "NoError". "Not ref'd!" (in rot) bedeutet "nicht referenziert".
- Der Roboter muss sich im Status "NoError" befinden.
- Drücken Sie nun die Schaltfläche "Ref All", um alle Achsen zu referenzieren. Der Roboter führt nun Suchbewegungen für jede Achse aus. Eine erfolgreich referenzierte Achse wird angezeigt, indem die Zahl "0" hinter dem Button "RefXXX" auf "1" wechselt.
- Nach der Referenzierung befinden sich die Achsen im Fehlerzustand. Dies ist notwendig, weil sich die Ist-Position während des Referenzierungsvorgangs geändert hat.



Abbildung 34: Drücken Sie nun nochmals den runden Knopf "Reset", gefolgt von dem runden Knopf "Enable". Der Roboter ist nun referenziert. Linearbewegungen und Programmieren können nun abgefahren werden.

## 9.5 Starten und Stoppen eines Programms



Abbildung 35: Drücken Sie dazu die Taste "Play" am oberen Rand des Displays und wählen Sie mit den Tasten "prev" und "next" ein Programm auszuwählen, das für Ihren Roboter geschrieben wurde.

- Laden Sie das Programm mit der Schaltfläche "Load".
- Lassen Sie das Programm mit "Single Play" einmal abspielen.
- "Cont. Play" spielt das geladene Programm kontinuierlich ab.
- "Stop" hält die Bewegung an.
- Der Schieberegler "Override" kann nach rechts bewegt werden, um die Geschwindigkeit der Bewegung zu erhöhen, oder nach links, um sie zu verringern.

## 9.6 Setzen der digitalen Ausgänge und Ablesen der digitalen Eingänge

Die digitalen Ausgänge können durch Drücken der abgebildeten Schalter gesetzt werden (s. Abb. 36). Die Felder darunter zeigen die Zustände der digitalen Eingänge. Mit den drei Schaltflächen auf der linken Seite können Sie zwischen mehreren DIO-Modulen wechseln.

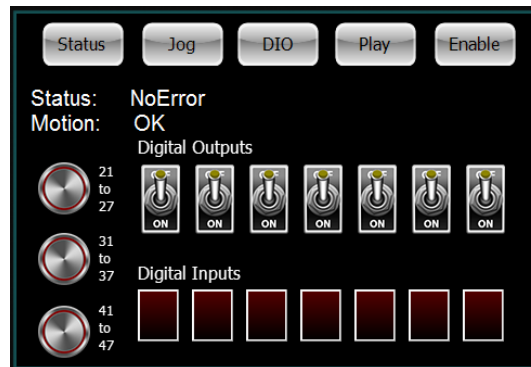


Abbildung 36: Wenn z.B. ein Greifer an das DIO-Modul angeschlossen ist, kann dieser durch Umschalten der abgebildeten Schalter aktiviert oder deaktiviert werden.

### 9.7 Anzeige von Statusinformationen

Statusinformationen können durch Drücken der Taste "Status" in der linken oberen Ecke des Displays angezeigt werden.

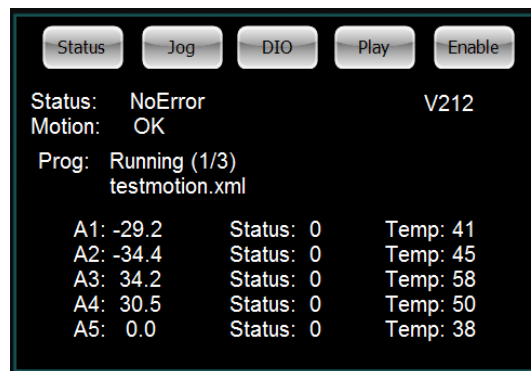


Abbildung 37: Der Name eines Programms wird angezeigt, z.B. während ein Programm läuft. Die Achspositionen der Achsen 1-5 werden angezeigt (A1-A5). Die Temperatur der Achsmodule wird ebenfalls angezeigt. Aufgrund der unterschiedlichen Belastungen der einzelnen Achsen werden unterschiedliche Halteströme angelegt, die zu wechselnden Temperaturen der Schrittmodule führen.

## 10 Programmierung eines Roboters mit iRC

Die iRC - igus Robot Control ermöglicht die Erstellung von Roboterprogrammen. Die Art der Programmierung wird als "Teach-In-Programmierung" bezeichnet, welche wie folgt funktioniert:

1. Bewegen Sie den Roboter manuell an die Position, die Sie aufzeichnen möchten
2. Zeichnen Sie die Position auf und legen Sie fest, wie diese Position erreicht werden soll (Linear-/Achsbewegung)
3. Wiederholen Sie diese Schritte und fügen Sie bei Bedarf zwischendurch digitale Ausgabebefehle oder Programmflussbefehle hinzu.

Zum Erstellen und Bearbeiten dieser Programme steht der integrierte Editor zur Verfügung.

### 10.1 Programmeditor

Im Roboterprogramm besteht jeder Befehl aus einer Zeile, z.B. "Joint" oder "Wait". Nur Befehle, die den Programmfluss steuern, werden in mehrere Zeilen aufgeteilt, z.B. "Loop" und "EndLoop". Der Programmeditor wird mit der Schaltfläche "Programm bearbeiten" im "Start"-Tab der iRC geöffnet.

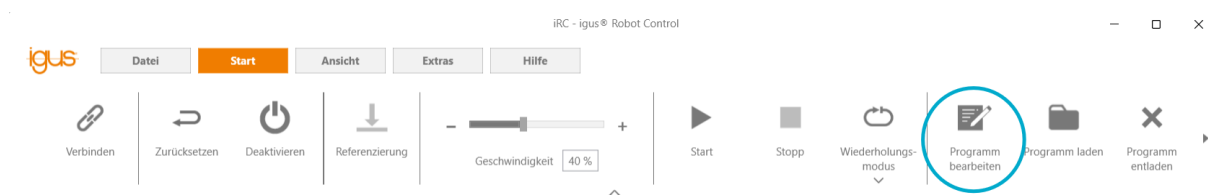


Abbildung 38: Öffnen des Programmeditors mit der Schaltfläche "Bearbeiten".

Es öffnet sich das folgende Fenster, hier mit einem kurzen Programm:



Abbildung 39: Programmeditor mit einem kurzen Programm

Die folgenden Abschnitte zeigen, wie man Programme mit dem Editor erstellt.

### 10.1.1 Ändern der Befehlssequenz

Um einen Befehl zu verschieben verwenden Sie die Pfeile auf der rechten Seite der Befehlszeile. Alternativ können Sie im Kontextmenü der Befehlszeile auf "Nach unten" oder "Nach oben" klicken (s. Abb. 40).

Der Programmeditor verhindert fehlerhafte Befehle, die die Struktur des Programms beschädigen würden. Wenn das Verschieben eines Befehls nach oben oder unten nicht möglich ist, werden die entsprechenden Schaltflächen und Menüpunkte ausgegraut.

### 10.1.2 Position nachbessern

Bestimmte Befehle erfordern Positionswerte als Parameter. Häufig ist es wünschenswert, die aktuelle Position des Roboters im gerade ausgewählten Bezugssystem zu verwenden. Die Eingabe von Hand kann einige Zeit dauern und ist fehleranfällig. Für solche Fälle können Sie den Befehl "nachzubessern":

- Markieren Sie den Befehl und klicken Sie im Menü "Bearbeiten" auf "Nachbessern!"
- Markieren Sie den Befehl aus und drücken Sie dann Strg+T
- Öffnen Sie das Kontextmenü durch einen Rechtsklick auf die Befehlszeile und klicken Sie auf "Nachbessern!" (s. Abb. 40).

Der Programmeditor ersetzt dann die Positionswerte im Befehl durch die aktuelle Position des Roboters.

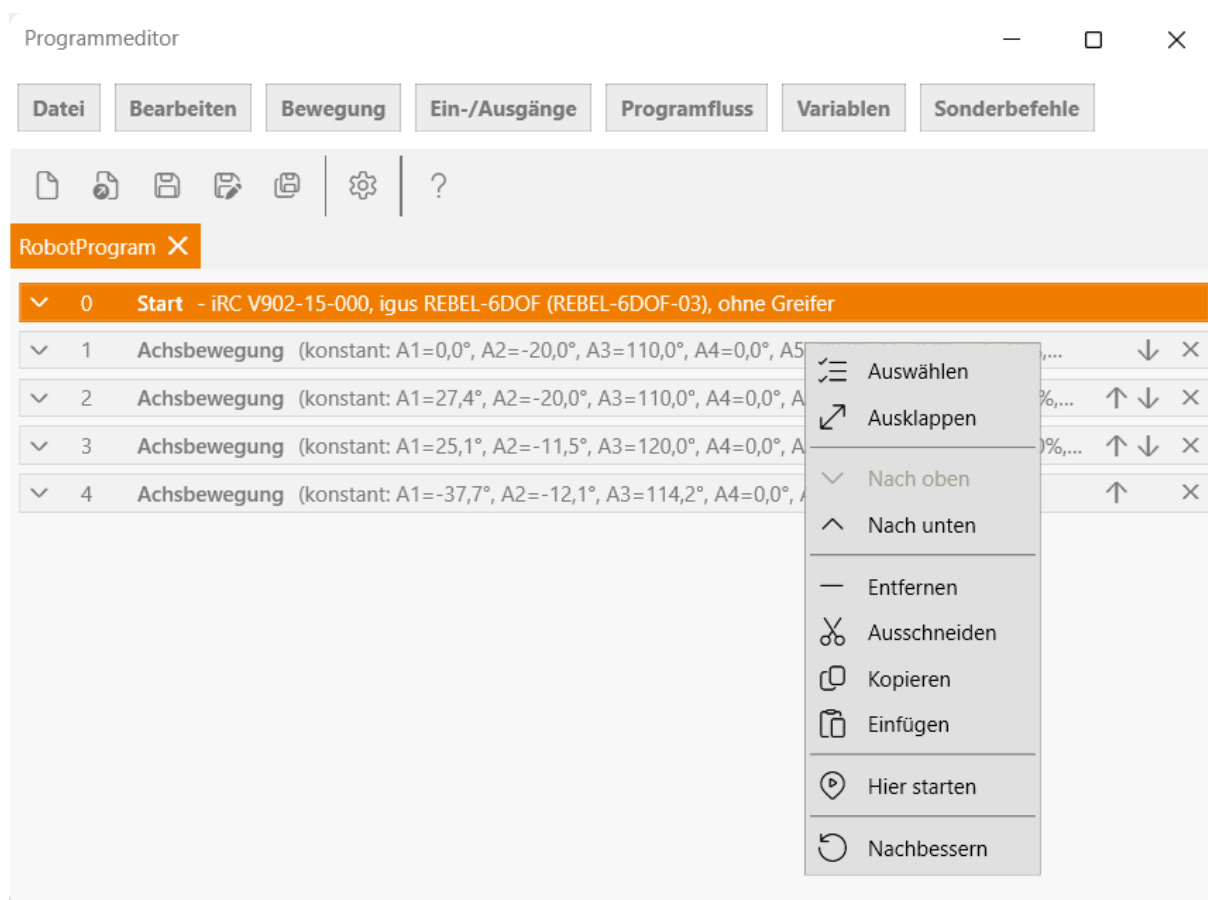


Abbildung 40: Das Kontextmenü einer Befehlszeile

### 10.1.3 Startbefehl festlegen

Es ist möglich, Programme schrittweise auszuführen oder zu Testzwecken einen bestimmten Befehl als Startpunkt eines Programms auszuwählen. Der Befehl, der beim nächsten Start des Programms zuerst ausgeführt wird, oder - falls das Programm gerade läuft - der aktuell ausgeführte Befehl wird im Programmeditor durch einen Punkt markiert. Das Unterprogramm, das diesen Befehl enthält, ist ebenfalls durch einen Punkt vor dem Programmnamen gekennzeichnet.

Um einen bestimmten Befehl als Startpunkt für die Ausführung zu wählen, klicken Sie im Kontextmenü auf "Hier starten" (s. Abb. 40).

## 10.2 Roboter- und Logikprogramme

Innerhalb eines Roboterprogramms werden Anweisungen immer nacheinander ausgeführt, es ist also beispielsweise nicht möglich während einer Bewegung einen digitalen Ausgang zu schalten oder Berechnungen durchzuführen. Falls dies notwendig ist kann ein Logikprogramm eingesetzt werden. Ein Logikprogramm wird wie ein Roboterprogramm über den Programmeditor erstellt, darf allerdings keine Bewegungsanweisungen enthalten. Nachdem es zugewiesen wurde wird es dauerhaft wiederholt, auch wenn das eigentliche Roboterprogramm nicht läuft. Um ein Programm als Logikprogramm zu laden muss es zunächst auf die Robotersteuerung übertragen werden indem es einmal wie ein nor-

males Programm geladen wird. Danach kann es im Konfigurationsbereich "Datei" → "Konfiguration" → "Programm" → "Logikprogramm" zugewiesen werden.



Programme haben eine minimale Ausführungsdauer von 500ms. Kürzere Programme, beispielsweise schnell laufende Logikprogramme, werden beim automatischen Neustart ggf. verzögert. Dies ist besonders zu beachten wenn das Logikprogramm dazu verwendet wird digitale Ausgänge zu Schalten oder die Zustände der Eingänge auszuwerten. Falls eine schnelle Wiederholung des Programms nötig ist kann eine Dauerschleife (Bedingung "False") angelegt werden die alle anderen Anweisungen des Programms enthält.



**Anwendungsbeispiel:** Während einer Bewegung soll ein Ventil geöffnet werden um Klebstoff aufzutragen.

1. Definieren Sie die Bewegung im Roboterprogramm.
2. Bevor die Bewegung startet setzen Sie im Roboterprogramm ein globales Signal (GSig, Anweisung "Digitaler Ausgang") um dem Logikprogramm zu signalisieren dass es bald den digitalen Ausgang für das Ventil schalten soll.
3. Erstellen Sie ein Logikprogramm mit einer Dauerschleife (Schleife mit Bedingung "False", siehe Info-Box oben).
4. In der Dauerschleife des Logikprogramms erstellen Sie eine Bedingungsanweisung ("IF"). Definieren Sie dessen Bedingung so, dass das globale Signal aktiv und die Sollposition erreicht sein muss. Beispielsweise "GSig1 and #position.x > 150" wenn der Ausgang ab X=150mm aktiviert werden soll.
5. Innerhalb der Bedingungsanweisung setzen Sie den digitalen Ausgang um das Ventil zu aktivieren
6. Setzen Sie dort ebenfalls das globale Signal zurück sodass die Bedingung im nächsten Durchlauf nicht wieder aufgerufen wird. In komplexeren Anwendungsfällen könnte das Logikprogramm auch ein zweites globales Signal setzen um nach Erreichen der Zielposition das Ventil wieder zu schließen.

## 10.3 Kommentare und Informationen im Programm

### 10.3.1 Informationen zum Programm

Der Programmreditor fügt zu Beginn jedes Programms den Pseudobefehl "Start" ein. Er stellt keinen echten Befehl dar, sondern zeigt Informationen über die aktuelle Hardware, Software und Kinematik an. Es ist nicht möglich, ihn zu verschieben oder zu entfernen.

Beim Laden eines Programms werden diese Informationen abgeglichen um zu vermeiden, dass ein Roboter ein inkompatibles Programm ausführt.

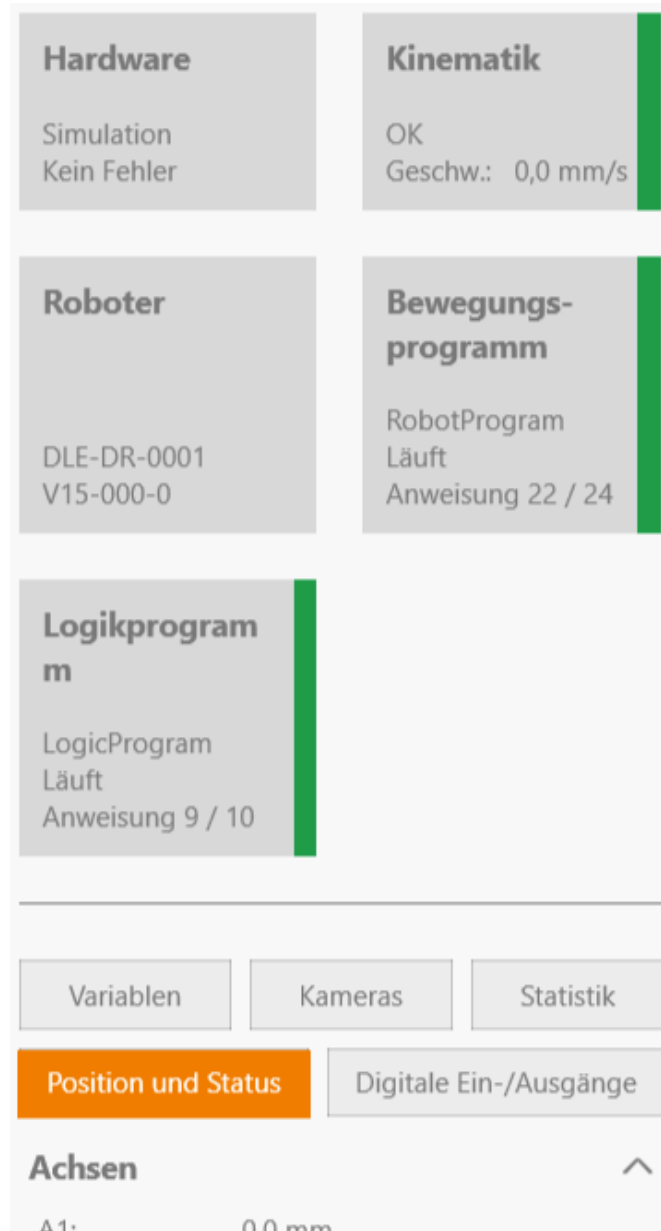


Abbildung 41: Die Kachel links in der iRC zeigt den Zustand von Roboter- und Logikprogramm, sofern ein Programm geladen ist.

Variablen	Kameras	Statistik
Position und Status	Digitale Ein-/Ausgänge	
<b>Aktionen</b>	▼	
<b>Filter</b>	▼	
<b>Zahlenvariablen</b>	+ ^	
#logicprogramrunning	1.0	
#parts-bad	0.0	
#parts-good	25.0	
#path_index	-1.0	
#programrunning	1.0	

Abbildung 42: Über die Variable #logicprogramrunning kann das Roboterprogramm prüfen ob das Logikprogramm läuft

Start - iRC V902-15-000, igus REBEL-6DOF (REBEL-6DOF-03), ohne Greifer		
Informationen		
Robotername:	igus REBEL-6DOF	Kinematik: Arm6Dof
Robotertyp:	REBEL-6DOF-03	Software: iRC V902-15-000
Greifertyp:	keiner	
Geschw.-Einstellung:	24V	

Abbildung 43: Die Start-Zeile enthält Informationen über die aktuelle Hardware.

### 10.3.2 Beschreibungen

Jeder Befehl eines Programms enthält eine Beschreibung. Sie sollte dazu genutzt werden anderen Benutzern zu beschreiben wozu der Befehl dient.

### 10.3.3 Kommentare

Der Befehl Kommentar kann verwendet werden, um reine Beschreibungen in Programme einzufügen. Er hat keine Auswirkungen auf den Roboter während der Ausführung.

Er ist im Programmmeditor im Menüeintrag "Sonderbefehle" → "Kommentar" zu finden.

## 10.4 Bewegung

### 10.4.1 Abbruchbedingungen

Jeder Bewegungsbefehl kann mit einer Abbruchbedingung versehen werden. Es handelt sich um einen bedingten Ausdruck, der in Abschnitt 10.7.1 beschriebenen Syntax folgt. Während der Ausführung des Bewegungsbefehls wird die Anweisung kontinuierlich ausgewertet, und in dem Moment, in dem sie als "wahr" bewertet wird, hält der Roboter die Bewegung an. Sie kann jeweils unter "Abbruchbedingung" für jeden Bewegungsbefehl angegeben werden.

### 10.4.2 Beschleunigung und Glättung

Für jeden Bewegungsbefehl kann eine Achsbeschleunigung (prozentualer Anteil der maximalen Beschleunigung) und ein Glättungsfaktor angegeben werden um abrupte Bewegungen zu verhindern. Bei einem Glättungsfaktor von 1-100% wird die Bewegungsanweisung mit der darauf folgenden Anweisung überschleift, sodass beispielsweise mehrere Linearbewegungen flüssige Kurven bilden anstatt bei jedem Zielpunkt abzubremsen und neu zu anzufahren.

Das Überschleifen ist nur bei direkt auf einander folgenden Bewegungsanweisungen gleicher Art möglich. Z.B. können Linear und Kreisbewegungen mit einander überschleift werden und Achsbewegungen mit sich selbst. Wird eine Bewegungssequenz von einer Bewegung anderer Art oder einer Logikanweisung unterbrochen wird auch das Überschleifen unterbrochen und der Roboter stoppt kurz. Die maximale Anzahl der auf einander folgenden überschleifbaren Anweisungen ist je nach Roboter auf 5-20 begrenzt um Rechenpausen im Programmablauf zu verhindern. Bei langen Bahnen zeigt sich dies durch ein regelmäßiges Abbremsen und neu Anfahren.



Falls Ihr Anwendungsfall lange ununterbrochene Bahnen erfordert können Sie die Begrenzung in Konfigurationsbereich "Roboterkonfiguration" -> "Pfadgenerierung" erhöhen. Beachten Sie dass dies bei sehr langen Pfaden zu einer kurzen Rechenpause vor dem Anfahren führen kann. Mehr Informationen finden Sie auf unserem Wiki unter dem Stichwort "LookAhead".

[https://wiki.cpr-robots.com/index.php/Motion\\_Smoothing](https://wiki.cpr-robots.com/index.php/Motion_Smoothing)



### 10.4.3 Achsbewegung

Der Befehl "Achsbewegung" bewegt den Roboter zu einer absoluten Zielposition, die in Achskoordinaten (z.B. Achswinkel oder Position einer Linearachse) angegeben ist. Die daraus resultierende Bewegung des TCP ist in der Regel eine Kurve und keine gerade Linie. Die Zielposition kann auf folgende Weise angegeben werden (wählen Sie die entsprechende "Quelle"):

- "Konstante": Die Zielposition ist ein konstanter Wert für jede Achse.

1 Achsbewegung (konstant: A1=0,0°, A2=-20,0°, A3=110,0°, A4=0,0°, A5=90,0°, A6=0,0°, vel=60%,...
✕

Quelle  
Konstante

Parameter  
Geschwindigkeit: 60,0 % | Beschleunigung: 40 % | Glättung: 0 %

Achse 1: 0,00 °

Achse 4: 0,00 °

Ext. Achse 1: n. v.

Achse 2: -20,00 °

Achse 5: 90,00 °

Ext. Achse 2: n. v.

Achse 3: 110,00 °

Achse 6: 0,00 °

Ext. Achse 3: n. v.

Abbruchbedingung  
False

Beschreibung

- "Variable": Die Zielposition wird aus der Positionsvariablen übernommen, die unter "Variable" angegeben wird.

1 Achsbewegung (variabel: var=myPosVar, vel=60%, acc=40%, smooth=0%)
✕

Quelle  
Variable

Parameter  
Geschwindigkeit: 60,0 % | Beschleunigung: 40 % | Glättung: 0 %

Variable  
myPosVar

Abbruchbedingung  
False

Beschreibung

Die Bewegungsgeschwindigkeit wird durch "Geschwindigkeit" angegeben. Sie wird in Prozent der maximal erlaubten Bewegungsgeschwindigkeit für die jeweiligen Roboterachsen gemessen. Der Achsbewegung-Befehl kann im Programmreditor unter den Menüeinträgen "Bewegung" → "Achsbewegung" aufgerufen werden.

#### 10.4.4 Lineare Bewegung

Der Befehl "Linear" bewegt den Roboter zu einer absoluten Zielposition, die in kartesischen Koordinaten angegeben ist. Die resultierende Bewegung des TCP folgt einer geraden Linie. Die Zielposition kann wie folgt angegeben werden (wählen Sie die entsprechende "Quelle"):

- "Konstante": Die Zielposition ist eine Konstante, die durch kartesische Koordinaten X, Y, Z, die Eulerwinkel A, B, C sowie die Positionen der externen Achsen gegeben ist, wenn diese von der Roboterkinematik unterstützt werden.

1 Linear (konstant: X=200 mm, Y=0 mm, Z=350 mm, A=180,0°, B=0,0°, C=180,0°, vel=100 mm/s, acc=40%, smooth=0%)

Quelle: Konstante

Parameter: Geschwindigkeit: 100,0 mm/s | Beschleunigung: 40 % | Glättung: 0 % | Koordinatensystem: #base

Konstante

X: 200,0 mm	A: 180,00 °	Ext. Achse 1: n. v.
Y: 0,0 mm	B: 0,00 °	Ext. Achse 2: n. v.
Z: 350,0 mm	C: 180,00 °	Ext. Achse 3: n. v.

Abbruchbedingung: False

Beschreibung:

- "Variable": Die Zielposition wird aus der Positionsvariablen übernommen, die unter "Variable" angegeben wird.

1 Linear (variabel: var=myPosVar, vel=100 mm/s, acc=40%, smooth=0%)

Quelle: Variable

Parameter: Geschwindigkeit: 100,0 mm/s | Beschleunigung: 40 % | Glättung: 0 % | Koordinatensystem: #base

Variable: myPosVar

Abbruchbedingung: False

Beschreibung:

Die Bewegungsgeschwindigkeit wird durch "Geschwindigkeit" in mm/s angegeben. Wenn sie die maximal zulässige Bewegungsgeschwindigkeit des Roboters überschreitet, führt dies zu einem kinematischen Fehler während der Ausführung. Der Linearbewegung-Befehl kann im Programmreditor unter "Bewegung" → "Linearbewegung" aufgerufen werden.

Bitte beachten Sie, dass die Zielposition einer Linearbewegung sich immer auf ein bestimmtes Koordinatensystem bezieht, in Abschnitt 11 finden Sie weitere Informationen zum Thema benutzerdefinierte Koordinatensysteme.



Der Roboter wird versuchen die Bewegung 100% genau auszuführen, auch wenn es durch die Kinematik nicht möglich ist. Achten Sie darauf, dass die Zielpositionen so gewählt werden, dass sie von der Kinematik des Roboters erreicht werden können. Andernfalls kann es zu unerwarteten Bewegungen oder Fehlern kommen.

#### 10.4.5 Achsbewegung zu kartesischer Position

Nicht immer ist eine lineare Bewegung zu einer Position im kartesischen Raum (XYZ) sinnvoll oder möglich. Beispielsweise wenn die kartesische Zielposition schnellstmöglich angefahren werden soll, der Weg dorthin aber nicht linear sein muss oder wenn auf dem Weg dorthin eine Singularität durchfahren werden muss. Hierzu kann die Anweisung "Achsbewegung zu kartesischer Position" verwendet werden. Anders als der normalen Achsbewegung wird dieser eine kartesische Position gegeben.

Zu Beginn der Bewegung wird diese in Ziel-Achswinkel umgerechnet und wie mit der normalen Achsbewegung angefahren. Der Roboter fährt so schnell wie die langsamste Achse erlaubt. Der Befehl kann im Programmreditor unter "Bewegung" → "Achsbewegung zu kart. Pos." aufgerufen werden.

#### 10.4.6 Relative Bewegung

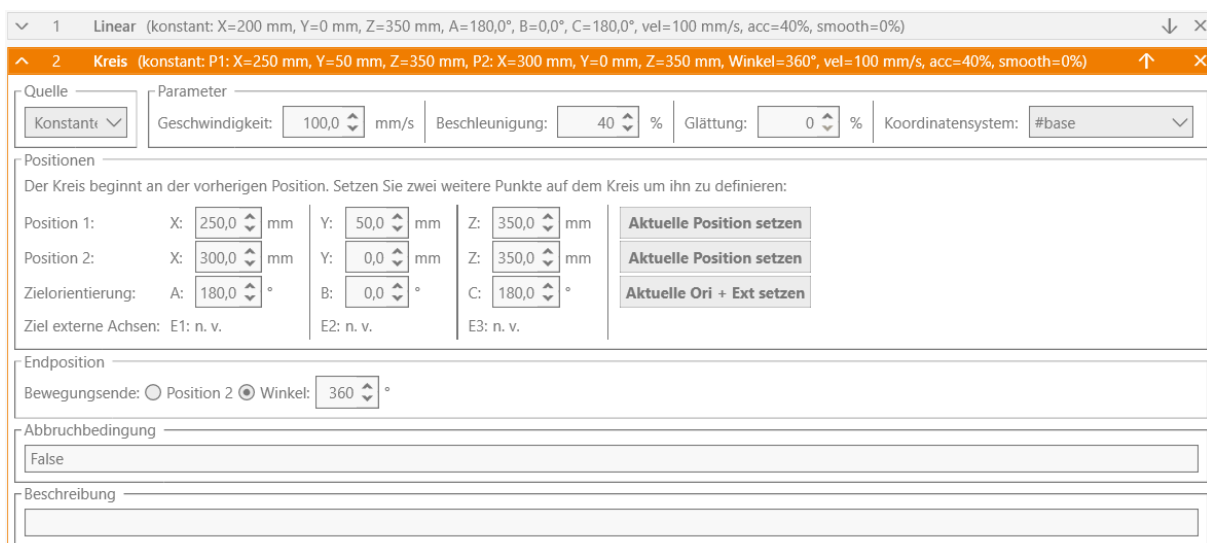
Der Befehl "Relativ" erlaubt es, den Roboter relativ zu seiner aktuellen Position zu bewegen. Er kann über die Menüpunkte unter "Bewegung" → "Relative Bewegung" aufgerufen werden.

Unter "Typ" können die folgenden Modi der Relativbewegung gewählt werden:

- "Achsbewegung": Der relative Versatz wird in Achskoordinaten angegeben. Die Bewegungsgeschwindigkeit wird durch "Geschwindigkeit" in Prozent der maximal erlaubten Bewegungsgeschwindigkeit für die jeweiligen Roboterachsen angegeben.
- "Linear - Basis": Es wird eine lineare Bewegung mit einem in kartesischen Koordinaten angegebenen Versatz durchgeführt. Für den Versatz wird das angegebene Koordinatensystem verwendet (in Abschnitt 11 finden Sie weitere Informationen zum Thema benutzerdefinierte Koordinatensysteme). Die Geschwindigkeit wird durch "Geschwindigkeit" angegeben. Sie wird in mm/s gemessen, wenn sie die maximal zulässige Bewegungsgeschwindigkeit des Roboters überschreitet, führt dies zu einem kinematischen Fehler während der Ausführung.
- "Linear - Werkzeug": Es wird eine lineare Bewegung mit einem in kartesischen Koordinaten angegebenen Versatz durchgeführt. Das für den Versatz verwendete Koordinatensystem sind Werkzeugkoordinaten. Die Bewegungsgeschwindigkeit wird durch "Geschwindigkeit" angegeben. Sie wird in mm/s gemessen. Wenn sie die maximal zulässige Bewegungsgeschwindigkeit des Roboters überschreitet, führt dies zu einem kinematischen Fehler während der Ausführung.

#### 10.4.7 Kreisbewegung

Die Anweisung "Kreisbewegung" ermöglicht Bewegungen entlang einer Voll- oder Teilkreisbahn. Sie ist kompatibel zu Linearbewegungen, sodass der Übergang von und zu Linearbewegungen ruckfrei überschritten werden kann.



Die Kreisbahn wird durch drei auf dem Kreis liegende Punkte definiert, wie in Abbildung 44 gezeigt. Der Startpunkt wird durch die Zielposition der vorherigen Bewegungsanweisung festgelegt. Position 1 und Position 2 sind beliebige Punkte auf dem Kreis. Der Parameter "Bewegungsende" legt fest, ob der Kreis an Punkt 2 oder nach einem bestimmten Winkel verlassen wird. Der Winkel kann dabei vor oder nach Punkt 2 liegen, auch ein mehrfaches Kreisen (Winkel größer  $360^\circ$ ) oder fahren in umgekehrter Richtung (negativer Winkel) ist möglich.

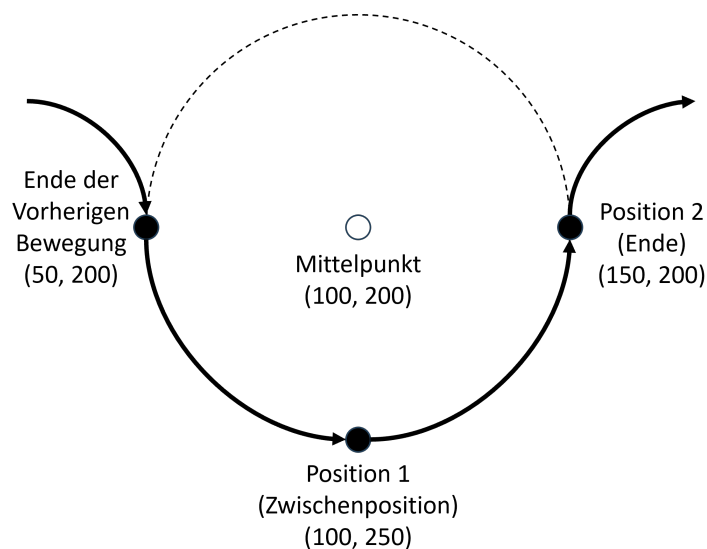


Abbildung 44: Beispiel einer Kreisdefinition. In der Ebene lassen sich die drei Kreispositionen einfach berechnen indem der Radius (hier 50mm) auf den Mittelpunkt addiert oder subtrahiert wird.



Die Kreisbewegung darf nicht die erste Bewegungsanweisung im Roboterprogramm sein. Falls das Programm an einer unerwarteten Position startet kann der Kreis größer oder kleiner werden als erwartet.



#### Programmierung einer Kreisbewegung durch Berechnung der Kreispunkte

Für einfache Kreisbewegungen in einer Ebene empfiehlt es sich wie in Abbildung 44 gezeigt zur Berechnung der Start-, Hilfs- und Zielposition die Koordinaten des Kreismittelpunkts zu berechnen und den Radius entlang der kartesischen Achsen zu addieren bzw. zu subtrahieren. Machen Sie ggf. eine Skizze.

Auch wenn nicht  $180^\circ$  gefahren werden soll empfiehlt es sich häufig Positionen 1 und 2 bei  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  oder  $270^\circ$  anzulegen und den tatsächlich zu fahrenden Winkel über den Parameter Bewegungsende anzugeben. Dadurch können die Positionen wie oben beschrieben ohne Trigonometrie berechnet werden.



### Programmierung einer Kurvenbahn per Teach-In

1. Fahren Sie den Roboter an die Startposition der Kurven und fügen Sie eine Linearanweisung hinzu.
2. Fahren Sie den Roboter an eine beliebige Position auf der Kurven, z.B. die Hälfte des Wegs. Fügen Sie die Kreisanweisung hinzu.
3. Fahren Sie den Roboter an das Ende der Kurven und klicken Sie die Schaltfläche "Aktuelle Position setzen" in der Zeile "Position 2"
4. Optional: Klicken Sie die Schaltfläche "Aktuelle Ori + Ext setzen" wenn sich die Werkzeugorientierung oder die Zusatzachsen während der Kurvenbahn ändern.

Wenn das Werkzeug während der Kreisbewegung mitgedreht werden soll kann die Zielorientierung angegeben werden. Die Winkel von der Startorientierung bis zur Zielorientierung linear interpoliert. Falls angegeben werden die Positionen der externen Achsen ebenso berechnet.

Die Angabe der Positionen ist konstant oder über zwei Positionsvariablen möglich und bezieht sich immer auf das mit angegebene Koordinatensystem (in Abschnitt 11 finden Sie weitere Informationen zum Thema benutzerdefinierte Koordinatensysteme). Dabei sind die Zielorientierung und die externen Achsen in der Variable für Position 2 (Zielposition) anzugeben.

#### 10.4.8 Dauerbewegung

Die Anweisung "Dauerbewegung" weist eine externe Achse dazu an sich mit konstanter Geschwindigkeit zu bewegen, beispielsweise für ein Förderband. Die Bewegung stoppt erst wenn eine Bewegung von 0 Einheiten/s zugewiesen oder das Programm gestoppt wird. Diese Funktion wirkt sich nur auf externe Achsen aus, die für den Geschwindigkeitsmodus konfiguriert sind.

#### 10.4.9 Pfad

Die Pfadanweisung dient dazu, entlang eines Pfades einer .csv oder .dxf Datei zu fahren.

Achtung: Es dürfen nur folgende Elemente in der .dxf Datei verwendet werden: Linie, Point, Polyline, und Circle.

Für die Punkte in der .csv Datei gilt folgende Vorschrift:

```

1 X 2.91; Y 0.34; Z 16.79; A 124.15; B -70.0; C 0.0
2 X 2.39; Y 21.03; Z 172.47; A 113.77; B -70.0; C 0.0

```

Als Trennzeichen zwischen den Werten werden ";" verwendet, Nachkommastellen werden durch einen "." getrennt.

Über "Datei wählen..." kann die entsprechende Datei ausgewählt werden.

Über den Steuerungsmodus kann eingestellt werden, ob die Punkte positionsgenau oder mit konstanter Geschwindigkeit angefahren werden sollen. Es ist nicht möglich, dass die Punkte exakt angefahren werden und gleichzeitig die Geschwindigkeit konstant ist, da dies unendlich große Beschleunigungen voraussetzen würden. Mit der Aktivierung der automatischen Reihenfolge werden die einzelnen Elemente in der DXF-Datei zusammengefasst, wenn Start und Endpunkt gleich sind. Außerdem wird die Richtung und Reihenfolge der Elemente verändert, damit der Fahrweg zwischen den Elementen möglichst gering wird.

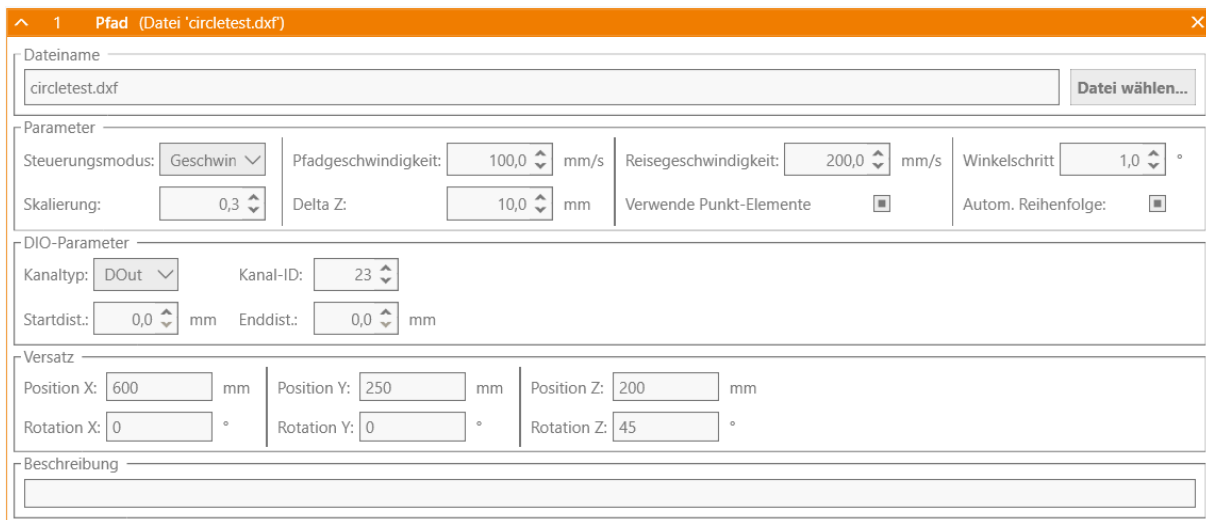


Abbildung 45: Pfad-Anweisung

Mit dem Parameter Skalierung kann das DXF-Objekt vergrößert oder verkleinert werden. Dabei wird das Objekt vom Koordinatenursprung aus skaliert, wodurch, je nach Lage der Punkte, eine Verschiebung des Objekts entstehen kann. Mit dem Parameter Delta Z kann eine Z-Höhe eingestellt werden, um die das Werkzeug zwischen zwei geschlossenen Objekten nach oben fahren soll.

Mit dem DIO-Parameter kann eingestellt werden, welcher digitale Ausgang oder welches globale Signal beim Abfahren der Bahn gesetzt werden soll. Dabei kann eine Start Distanz angegeben werden, die angibt, ab welcher Entfernung vor dem ersten Punkt der Ausgang bereits gesetzt werden soll. Mit der Enddistanz kann eingestellt werden, ab welcher Entfernung vor dem letzten Punkt der Ausgang wieder deaktiviert werden soll.

Wenn die Position des abgefahrenen Objekts verschoben werden soll, kann dies über die Offsets in X-, Y- und Z-Richtung erfolgen.

#### 10.4.10 Punktfolge

Mit der Punktfolge-Anweisung können DXF- oder CSV-Dateien abgefahren werden. An jedem Punkte wird die Z-Achse verfahren und dann auf ein GSig, ein DIn oder eine bestimmte Zeit gewartet, bevor der nächste Punkt angefahren wird.

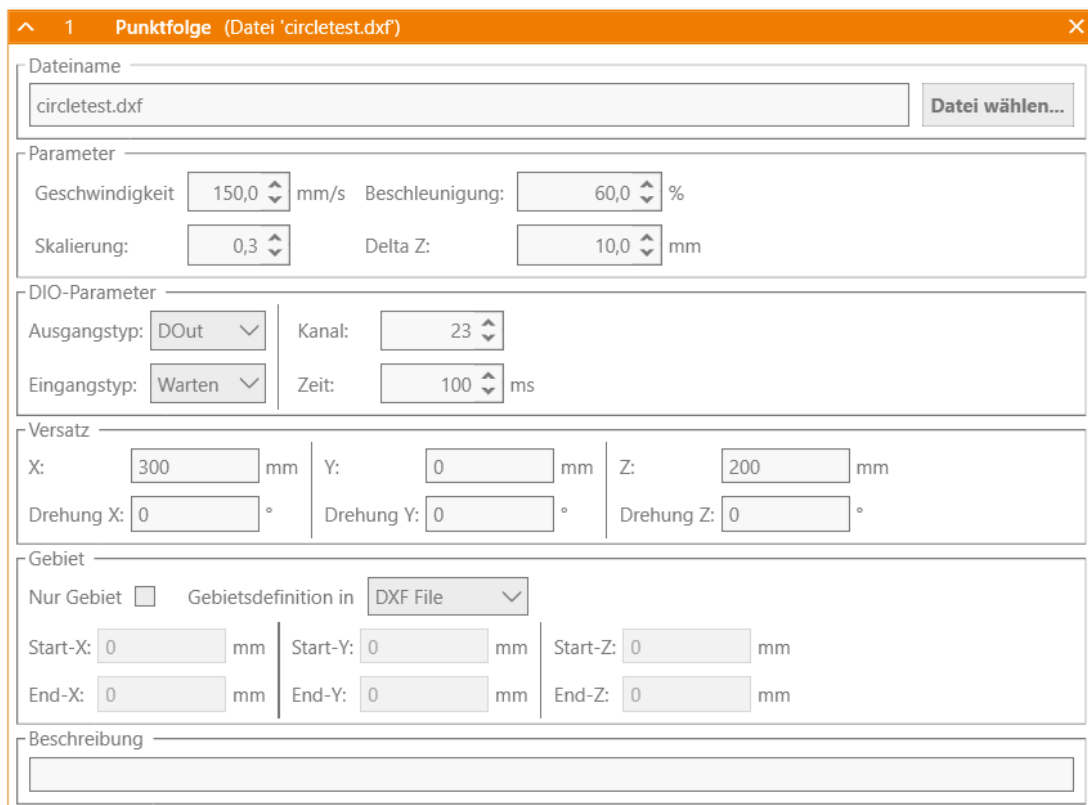
In der DXF-Datei werden nur Punkte oder Mittelpunkte von Kreisen angefahren. Alle anderen Elemente werden ignoriert. Die Punkte in der CSV-Datei müssen folgendermaßen aufgebaut sein:

```
1 X 2.91; Y 0.34; Z 16.79; A 124.15; B -70.0; C 0.0 X 2.39; Y 21.03; Z 172.47; A 113.77; B
-70.0; C 0.0
```

Ein ";" wird als Trennzeichen zwischen den Werten genutzt, als Dezimaltrennzeichen wird ein "." verwendet.

Die Anweisung ist folgendermaßen ist wie in Abbildung 46 gezeigt aufgebaut. Unter Dateiname wird die Datei ausgewählt.

Im Bereich "Parameter" können Einstellungen wie die Geschwindigkeit, ein Skalierungsfaktor und das Delta Z hinzugefügt werden. Das Delta Z gibt an, wie viel höher der Roboter in Z-Richtung zwischen zwei Punkten fahren soll.



1 Punktfolge (Datei 'circlestest.dxf')

Dateiname: circlestest.dxf Datei wählen...

Parameter

Geschwindigkeit: 150,0 mm/s Beschleunigung: 60,0 %

Skalierung: 0,3 Delta Z: 10,0 mm

DIO-Parameter

Ausgangstyp: DOut Kanal: 23

Eingangstyp: Warten Zeit: 100 ms

Versatz

X: 300 mm Y: 0 mm Z: 200 mm

Drehung X: 0 ° Drehung Y: 0 ° Drehung Z: 0 °

Gebiet

Nur Gebiet  Gebietsdefinition in: DXF File

Start-X: 0 mm Start-Y: 0 mm Start-Z: 0 mm

End-X: 0 mm End-Y: 0 mm End-Z: 0 mm

Beschreibung

Abbildung 46: Punktfolge-Anweisung

Im Bereich DIO-Parameter wird der Ausgang ausgewählt, der gesetzt wird, wenn der Roboter sich auf dem Punkt befindet. Mit dem Eingangstyp kann bestimmt werden, ob auf einen Eingang, ein globales Signal oder eine Zeit gewartet werden soll.

Wenn die Position des Objekts verschoben werden muss, kann dies mit Hilfe der Offsets in X-, Y- und Z-Richtung geschehen. Wenn es notwendig ist das Objekt zu drehen, können die Rotationsfelder verwendet werden. Rotation X dreht die DXF-Datei um die X-Achse, Rotation Y dreht die Y-Achse und Rotation Z dreht um die Z-Achse. Soll die Datei beispielsweise in der X/Y-Ebene um 90° gedreht werden, muss um die Z-Achse gedreht werden.

Im Bereich Gebiet können die angefahrenen Punkte auf einen Teilbereich beschränkt werden. Dabei wird durch die Startwerte in XYZ und die Endwerte in XYZ ein Quader festgelegt, in welchem sich die Punkte befinden dürfen. Es kann gewählt werden, ob die Bereichsdefinition im Koordinatensystem der Datei oder im Basiskoordinatensystem des Roboters gewählt werden soll.

## 10.5 Koordinatensysteme

Wie im Kapitel 11 erklärt können benutzerdefinierte Koordinatensysteme (BKS) eingerichtet werden um beispielsweise Bewegungsabläufe an mehreren unterschiedlich ausgerichteten Objekten wieder verwenden zu können. Benutzerkoordinatensysteme verschieben quasi den Nullpunkt und drehen die XYZ-Achsen an einem beliebigen Punkt.

Die Bewegungsanweisungen mit kartesischen Zielpositionen enthalten dafür den Parameter "Koordinatensystem", über den das zu verwendende Koordinatensystem angegeben werden kann. Neben den BKS sind die Koordinatensysteme "#base" und "#tool" vorgegeben. "#base" ist das Basiskoordinatensystem des Roboters mit Nullpunkt üblicherweise an dessen Basis. "#tool" ist relativ zum Greifpunkt des Werkzeugs.

### 10.5.1 Benutzerkoordinatensystem kopieren

Es ist häufig nützlich im Programmablauf zwischen Koordinatensystemen umzuschalten. So kann beispielsweise ein Bewegungsablauf mit einem Platzhalterkoordinatensystem in einem Unterprogramm definiert werden. Vor Aufruf des Unterprogramms wählen Sie welches Koordinatensystem verwendet werden soll. Hierzu ist die Anweisung "Benutzerkoordinatensystem kopieren" zuständig die ein existierendes Koordinatensystem in den Platzhalter kopiert.

Die Anweisung kann über "Bewegung" → "Koordinatensystem" → "Benutzerkoordinatensystem kopieren" hinzugefügt werden. Sie enthält die Parameter "Quelle" und "Ziel". Das unter Quelle angegebene Koordinatensystem wird an das unter Ziel angegebene Koordinatensystem kopiert. Existiert dieses noch nicht wird es erstellt. Achtung! Dies kann bereits bestehende Koordinatensysteme überschreiben!

## 10.6 Greifer und digitale Ein-/Ausgänge

### 10.6.1 Digitale Eingänge

Die Zustände der digitalen Eingänge können in Bedingungen verwendet werden (siehe Abschnitt 10.7.1). Der erste digitale Eingang des ersten digitalen I/O-Moduls hat die Nummer 21 und kann in Bedingungen über das Schlüsselwort DIn21 verwendet werden.

### 10.6.2 Digitale Ausgänge

Mit dem Befehl Digital Output werden digitale Ausgängen und globale Signale gesetzt.

Unter "Kanaltyp" wird angegeben, ob ein digitaler Ausgang oder ein globales Signal gesetzt werden soll. Unter "Kanal ID" wird der Kanal des digitalen Ausgangs oder des globalen Signals, unter "Zustand" der gewünschte Zustand nach Ausführung des Befehls angegeben. Der Befehl ist im Programmmeditor unter "Ein-/Ausgänge" → "Digitaler Ausgang" zugänglich.

### 10.6.3 Globale Signale

Globale Signale sind interne Merker die im Roboterprogramm wie digitale Ausgänge gesetzt und wie digitale Eingänge ausgewertet werden können. Sie können beispielsweise verwendet werden um einfache Zustandsinformationen zu merken oder um zwischen Roboterprogramm und Logikprogramm zu kommunizieren. Da sie auch über die CRI- und Modbus-Schnittstellen gelesen und gesetzt werden können können Roboterprogramme über sie auch mit externen Anwendungen oder einer SPS kommunizieren. Das Konzept der globalen Signale ist vergleichbar mit den Coils in Modbus oder einem Boolean in der Hochsprachenprogrammierung.

In Bedingungen im Roboterprogramm kann über das Schlüsselwort GSig der Zustand des globalen Signals abgefragt werden, beispielsweise GSig1 für das erste Signal. Es stehen 100 globale Signale zur Verfügung.

### 10.6.4 Öffnen/Schließen des Greifers

Der Befehl "Greifer" ermöglicht die Steuerung des Greifers des Roboters. Er ist im Programmmeditor über den Menüeintrag "Ein-/Ausgänge" → "Greifer" zugänglich.

Unter "Öffnung" können Sie die gewünschte Öffnung, gemessen in Prozent, einstellen. Ein Wert von 0% steht für einen vollständig geschlossenen, 100% für einen vollständig geöffneten Greifer. Bei Greifern, die nur entweder vollständig geöffnet oder vollständig geschlossen werden können, liegt die Schwelle zwischen diesen Zuständen bei 50%.



Damit der Befehl funktioniert, muss der Greifer zunächst korrekt konfiguriert werden. (siehe Abschnitt 13.10).

## 10.7 Programmfluss

### 10.7.1 Bedingungen

Bedingungen können in if-then-else-Befehlen, Schleifen und als Abbruch-Bedingungen in Bewegungsbefehlen verwendet werden. Die Bedingungen können Kombinationen aus digitalen Eingängen, globalen Signalen, booleschen Operationen und Vergleichen sein. Groß- und Kleinschreibung sowie Leerzeichen zwischen Symbolen wird dabei nicht beachtet.

Im einfachsten Fall kann eine Bedingung beispielsweise prüfen ob am digitalen Eingang 21 ein Signal anliegt:

DIn21

Komplexere Bedingungen lassen sich durch die Schlüsselworte AND und OR sowie durch Klammerung aufbauen. Die folgende Bedingung ist erfüllt wenn entweder an Eingang 21 oder an Eingängen 22 und 23 ein Signal anliegt:

```
DIn21 OR (DIn22 AND DIn23)
```

Um einen Ausdruck zu negieren setzen Sie ein Ausrufezeichen (!) davor. Das ist auch vor geklammerten Ausdrücken möglich. Die folgende Bedingung ist erfüllt wenn entweder an Eingang 21 kein Signal anliegt oder wenn an Eingängen 22 und 23 nicht gemeinsam ein Signal anliegt:

```
!DIn21 OR !(DIn22 AND DIn23)
```

Ebenso kann der Zustand der globalen Signale (siehe Abschnitt 10.6.3) abgefragt werden. Globale Signale sind interne Merker die auch zur Kommunikation zwischen Roboter- und Logikprogramm sowie externen Anwendungen oder SPS verwendet werden können:

```
GSig1 AND !DIn21
```



Die Zustände der digitalen Ausgänge können in Bedingungen nicht abgefragt werden. Wenn nötig kann nach dem Setzen des Ausganges ein globales Signal gesetzt werden welches den Ausgang repräsentiert.

Daneben können auch die Werte von Zahlen- und Positionsvariablen geprüft werden. Zahlenvariablen repräsentieren eine einzelne Zahl während Positionsvariablen mehrere Zahlenkomponenten enthalten. Bei Positionsvariablen muss daher immer angegeben werden welche Komponente verglichen werden soll.

```
meinezahlenvariable = 5
meinezahlenvariable < 10
meinezahlenvariable >= 42
meinepositionsvariable.X = 123
meinepositionsvariable.B >= 90
meinepositionsvariable.A3 > 300
meinepositionsvariable.E1 < 500
```

Zahlenwerte können auch mit einander verglichen werden:

```
meinepositionsvariable.X > meinezahlenvariable
meinepositionsvariable.A1 <= anderepositionsvariable.A1
```

Zum Vergleich von Positionen können die folgenden Positionskomponenten verwendet werden:

- kartesisch
  - X, Y, Z - Position in Millimeter
  - A, B, C - Orientierung in Grad
- Achspositionen
  - A1 bis A6 - Roboterachsen in Grad oder Millimeter
  - E1 bis E3 - Zusatzachsen in Grad, Millimeter oder selbst definierter Einheit

Zusammengefasst lässt sich die Bedingungssyntax durch die folgende EBNF-Definition beschreiben:

```
Ausdruck      := ["!"] <Boolean> <Boolscher Operator> <Boolean> ...
```

Boolean	:= <Boolesche Konstante>   <Ausdruck>   "(" <Ausdruck> ")"   VerglAusdruck   "(" <VerglAusdruck> ")"   <Digitaleingänge>   "(" <Digitaleingänge> ")"
BoolescherOperator	:= "And"   "Or"
BoolescheKonstante	:= "True"   "False"
Digitaleingänge	:= <KanalTyp> <KanalId>
KanalTyp	:= "Din"   "GSig"
Kanalld	:= ein ganzzahliger Wert
VerglAusdruck	:= <VerglWert> <VerglOperator> <VerglWert>
VerglWert	:= <Variable>   <Zahl>
Variable	:= <Zahlenvariable>   <Positionskomponente>
Zahlenvariable	:= Name einer Zahlenvariablen
Positionskomponente	:= <Positionsvariable> "." <Komponente>
Positionsvariable	:= Name einer Positionsvariablen
Komponente	:= "x"   "y"   "z"   "A"   "B"   "C"   "A1"   "A2"   "A3"   "A4"   "A5"   "A6"   "E1"   "E2"   "E3"
Zahl	:= Ganzzahl oder Gleitkommazahl
VerglOperator	:= "="   ">"   "<"   ">="   "<="

### 10.7.2 Stop

Der Befehl "Stop" stoppt die Programmausführung.

Er ist über den Menüeintrag "Programmfluss" → "Stoppen" verfügbar.

### 10.7.3 Pause

Der Befehl "Pause" unterbricht die Ausführung des Programms. Die Ausführung kann später vom Benutzer wieder aufgenommen werden.



"Stop" und "Pause" beeinflussen nur die Programmausführung des Bewegungsprogramms. Ein ggf. definiertes Logikprogramm läuft weiter.

### 10.7.4 Warten

Der Befehl "Warten" weist den Roboter an zu warten, bis eine vorgegebene Zeitspanne vergangen oder eine Bedingung erfüllt ist. Er ist über die Menüeinträge unter "Programmfluss" → "Warten" im Programmmeditor der iRC zugänglich.

Die verschiedenen Modi können unter "Typ" ausgewählt werden:

- "Zeitspanne": Die unter "Zeitspanne" angegebene Zeit wird abgewartet.
- "Bedingung": Es wird gewartet, bis die unter "Ausdruck" angegebene Bedingung als "wahr" bewertet wird.

### 10.7.5 Wenn-dann-sonst

Der "Wenn"-Befehl verzweigt die Ausführung des Programms in Abhängigkeit vom Wert eines bedingten Ausdrucks. Er ist über den Menüeintrag "Programmfluss" → "Wenn..., dann..." im Programmmeditor der iRC zugänglich.

Die angegebene Bedingung muss der in Abschnitt 10.7.1 beschriebenen Syntax entsprechen. Die Anweisungen zwischen "Wenn" und "Sonst" werden ausgeführt, wenn die Bedingung als wahr ausgewertet wird. Andernfalls werden die Anweisungen zwischen "Sonst" und "Wenn-Ende" ausgeführt.



Abbildung 47: Die Wenn-Anweisung verzweigt den Programmfluss.

### 10.7.6 Schleifen

Der Befehl "Schleife" ermöglicht die Definition von Ausführungsschleifen. Unter "Typ" kann zwischen den folgenden Schleifentypen gewählt werden:

- "Bedingung": Die Schleife wird so lange wiederholt, bis die angegebene Bedingung als "wahr" ausgewertet wird. Sie muss der in Abschnitt 10.7.1 beschriebenen Syntax entsprechen.
- "Zähler": Die Schleife wird die unter "Wiederholungen" angegebene Anzahl von Malen wiederholen.

Der Schleifenbefehl ist über die Menüpunkte "Programmfluss" → "Schleife" zugänglich.

### 10.7.7 SwitchCase

Der Befehl "SwitchCase" ermöglicht die Verzweigung des Programmflusses in mehrere Pfade abhängig von einem Zahlenwert der unter "Variable" angegeben ist. Er ist über den Menüeintrag "Programmfluss" → "Fallauswahl" zugänglich. Es können beliebig viele Fälle definiert werden. Jeder Fall ist mit einem Wertebereich verknüpft, definiert durch eine Unter- und eine Obergrenze. Wenn der Wert der angegebenen Variable innerhalb des Wertebereichs eines Falles liegt, wird der Anweisungsblock dieses Falles ausgeführt. Es ist möglich, dass mehrere Fälle den selben Wertebereich abdecken. In diesem Fall wird der Anweisungsblock des ersten passenden Falles ausgeführt. Wenn kein passender Fall gefunden wird, wird der Anweisungsblock unter "DefaultCase" ausgeführt.

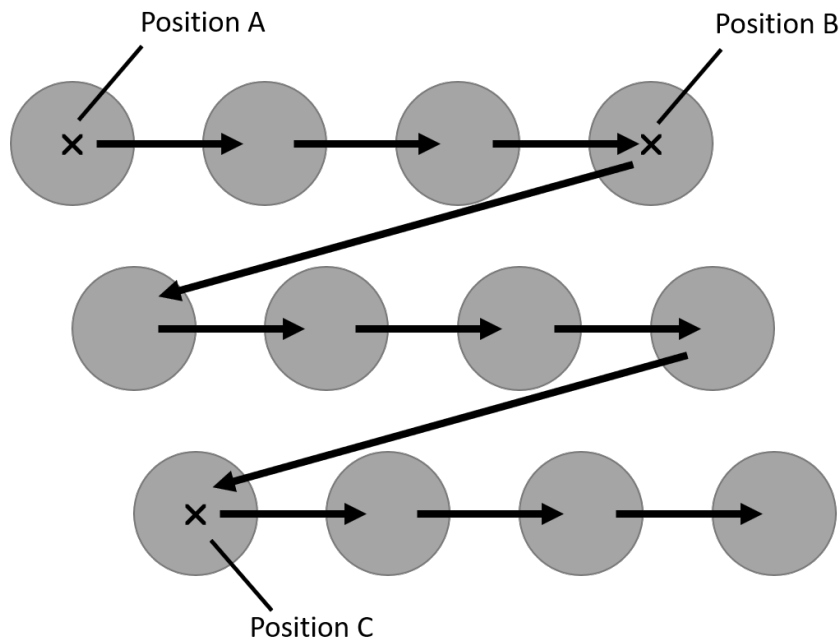


Abbildung 48: Die Matrixbewegung erfolgt von Punkt A nach B, dann in Richtung C versetzt.

### 10.7.8 Rasterbewegungen / Palettierung

Die Rasteranweisungen berechnen an einem Raster ausgerichtete Positionen, z.B. als Greif- oder Ablageposition für Palettierungsaufgaben. Abbildung 48 zeigt ein Bewegungsmuster, das durch die Verwendung der Rasteranweisungen ausgeführt werden kann.

iRC bietet zwei Ansätze um dies zu realisieren:

Anweisungsart	Beschreibung
Rasterschleife	Führt einen Anweisungsblock für jede Rasterposition aus.
Rasterdefinition und Rasterabfrage	Ermöglicht die Berechnung beliebiger Rasterpositionen anhand einer Indexvariablen.

Die Rasterschleife eignet sich für einfache Anwendungsfälle bei denen die Positionen strikt in ihrer Reihenfolge abgearbeitet werden. Die Schleife beginnt mit der ersten Position und wird erst nach der letzten Position verlassen. Die Rasterdefinition und -abfrage ist flexibler, es können mehrere Raster gleichzeitig verwendet werden und die Positionen in beliebiger Reihenfolge abgerufen werden, allerdings muss die Indexvariable durch zusätzliche Logik gezählt werden. Dies ermöglicht bspw. mit einer teilgefüllten Palette zu beginnen oder Positionen zu überspringen.

Die Rasterschleife kann über den Menüpunkt "Programmfluss" → "Rasterschleife" → "Rasterschleife" hinzugefügt werden. Rasterdefinition und Rasterabfrage befinden sich ebenfalls im Menü "Programmfluss" → "Rasterschleife".

Die als "Punkt A", "Punkt B" und "Punkt C" angegebenen Positionsvariablen definieren die Ecken des Bereichs, der von der Rasterschleife abgedeckt wird (siehe Abbildung 48). Die Anzahl der durchzuführenden Schritte wird durch "Zähler X" (von A nach B) und "Zähler Y" (von A nach C) angegeben. In der obigen Abbildung ist beispielsweise X=4 und Y=3.

---

Abbildung 49 zeigt eine Matrixschleife. Der Block zwischen "Matrix" und "Matrix End" wird für jeden Matrix-Schritt ausgeführt. Die Positionsvariable "Zielposition" enthält die Position des aktuellen Zielpunktes für den jeweiligen Schritt. Zeile und Spalte des aktuellen Schrittes werden in den Zahlenvariablen gespeichert, die an "Zähler X" bzw. "Zähler Y" gegeben werden.

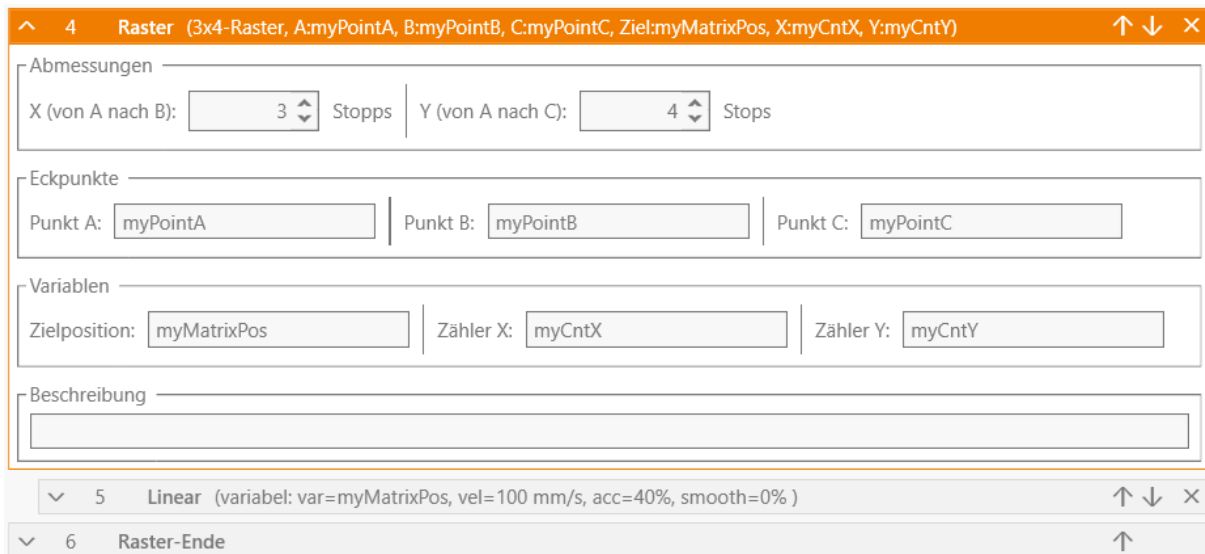


Abbildung 49: Definition einer Rasterschleife.

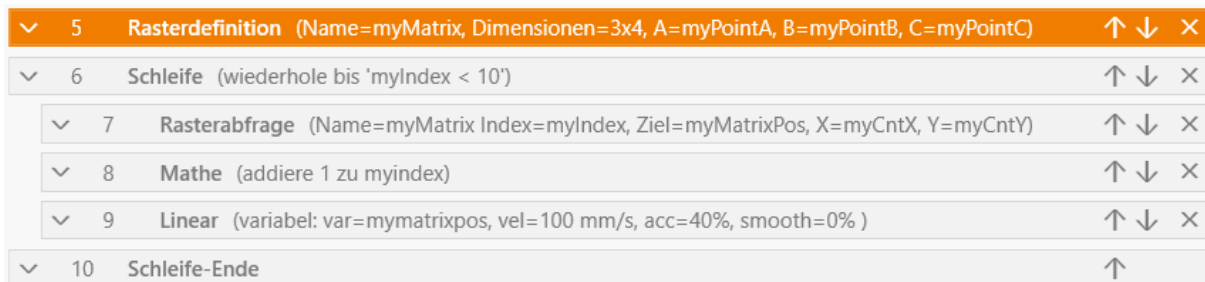


Abbildung 50: Verwendung der Rasterdefinition- und Rasterabfrageanweisung. Die Definition der Variablen wurde ausgelassen.

Die Anweisungen Rasterdefinition und -abfrage (s. Abb. 50) verwenden die selben Parameter: Die Abmessungen und Eckpunkte sind in der Rasterdefinition festgelegt, die Index- und Ausgabevariablen in der Rasterabfrage. Durch einen Namen werden unterschiedliche Raster identifiziert.

### 10.7.9 Unterprogramme

Mit dem Sub-Befehl können Unterprogramme aufgerufen werden.

Der Pfad zur Datei des Unterprogramms ist unter "Dateiname" angegeben. Er ist relativ zum Unterverzeichnis "Programs" des iRC-Ordners "Data". Der Befehl kann über den Menüpunkt "Programmfluss" → "Unterprogramm" aufgerufen werden.

## 10.8 Variablen und Variablenzugriff

In Roboterprogrammen werden zwei Arten von Variablen unterstützt:

- **Zahlenvariablen:** Diese können zur Speicherung von Ganzzahl- oder Fließkommazahlen verwendet werden.

- **Positionsvariablen:** Diese können zur Speicherung von kartesischen Positionen und Gelenkpositionen verwendet werden. Ob eine solche Variable als kartesische Position oder Gelenkposition interpretiert wird, hängt vom Kontext ab.  
Die Einheiten der kartesischen Position sind mm für die Komponenten x, y, z und Grad für die Eulerwinkel A, B, C. Die Achswerte werden je nach Art des Gelenks in mm oder Grad gemessen.

### 10.8.1 Benutzervariablen

Es ist möglich, Benutzer-Variablen mit dem Befehl "Setze" zu definieren, der im Programmeditor über die Menüeinträge unter "Variablen" → "Variablendefinition" zugänglich ist.

Drei Arten von Speicheroperationen können gewählt werden:

- "Derzeitige Position":  
Eine Positionsvariable wird mit der kartesischen und Achsposition initialisiert, die der Roboter bei der Ausführung des Befehls hat.
- "Zahlenkonstante":  
Eine Zahlenvariable wird mit der unter "Wert" angegebenen Konstante initialisiert (s. Abb. 51).
- "Positionskonstante":  
Eine Positionsvariable wird mit den unter "Kartesische Position", "Achsposition" und "Externe Achsen" angegebenen Konstanten initialisiert (s. Abb. 52). Je nach dem kinematischen Modell des aktuellen Roboters kann es sein, dass bestimmte Achsen nicht verfügbar sind.



Abbildung 51: Definition einer Zahlenvariable.

Der Name der Variable kann unter "Variable" eingestellt werden. Wenn eine Variable mit dem selben Namen bereits definiert wurde, werden ihr Wert und ihr Typ überschrieben. Alle Variablen sind global, d.h. sie sind auch von Unterprogrammen aus zugänglich.

### 10.8.2 Systemvariablen

Die folgenden vordefinierten Variablen stehen zur Verfügung, ohne dass sie definiert werden müssen:

- #position: Die aktuelle Position des Roboters.
- #programmrunning: 1 wenn das Roboterprogramm läuft, sonst 0
- #logicprogramrunning: 1 wenn das Logikprogramm läuft, sonst 0

Beachten Sie, dass das System die Werte von vordefinierten Variablen kontrolliert - sie können nicht vom Programm geändert werden. Die Namen von vordefinierten Variablen beginnen immer mit "#"

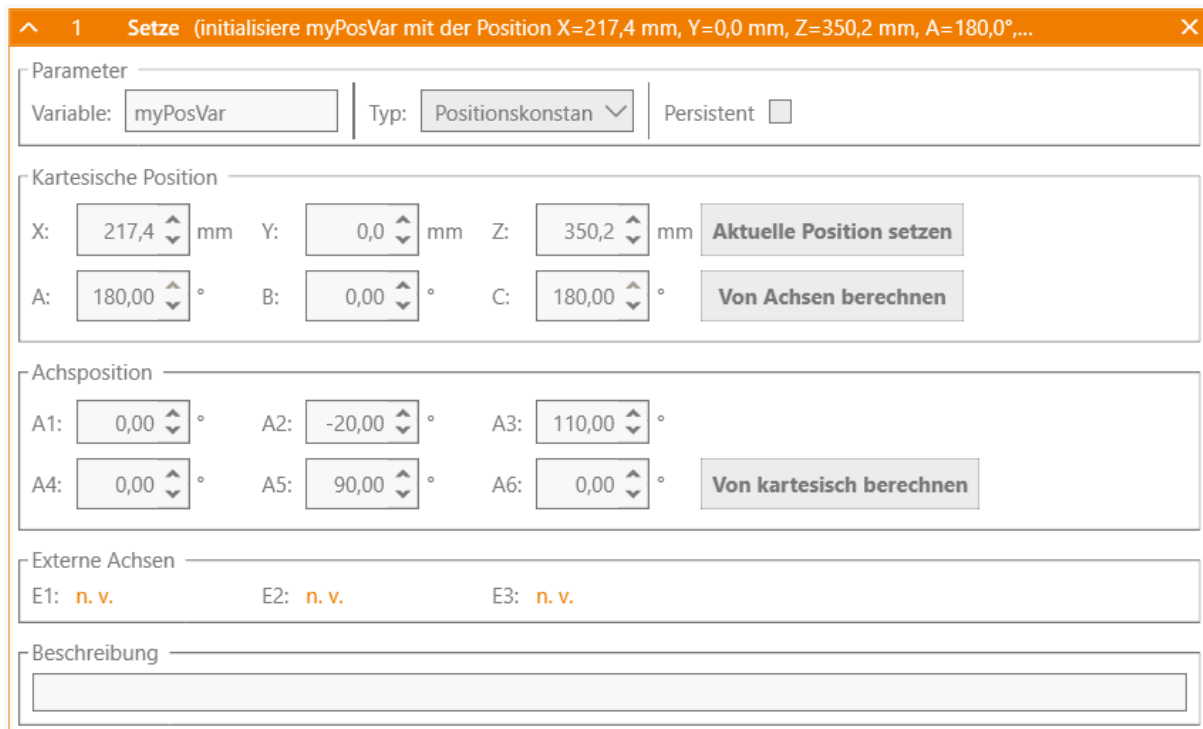


Abbildung 52: Definition einer Positionsvariable.

### 10.8.3 Persistente Variablen

Mit der Checkbox "Persistent" können Sie festlegen, dass der Wert einer Variablen auch nach Beendigung des Programms erhalten bleibt. Das ist beispielsweise nützlich, um Informationen zwischen verschiedenen Programmläufen zu speichern. Persistente Variablen können über die Benutzeroberfläche oder den Befehl "Zuweisung" zurückgesetzt werden, um sie wieder auf ihren Standardwert zu setzen. Der Befehl "Setze" hat keinen Einfluss mehr auf persistente Variablen, sobald diese schon einmal definiert wurden.



Persistente Variablen können auch in der Oberfläche im "Variablen"-Bereich auf der linken Seite durch Klicken auf das "+"-Symbol definiert werden.

### 10.8.4 Zugriff auf Elemente

Positionsvariablen enthalten folgende Elemente:

- Position: x, y, z
- Drehung: a, b, c
- Achswerte: a1, a2, a3, a4, a5, a6, e1, e2, e3

Der Zugriff auf die Elemente erfolgt durch Anhängen mit einem Punkt, z.B. "myvariable.x" oder "myvariable.a3".

### 10.8.5 Berechnungen mit Variablen

Berechnungen mit Variablen können mit dem Befehl "Mathe" ausgeführt werden, der im Programmmeditor über die Menüeinträge "Variablen" → "Mathematik" zugänglich ist.

"Erster Operand" definiert den ersten Operanden der Operation, die ausgeführt werden soll. Er wird auch dazu verwendet, das Ergebnis zu speichern.

"Zweiter Operand" definiert den zweiten Operanden der Operation. Er kann numerische Konstanten, Namen von Zahlenvariablen oder Komponenten von Positionsvariablen enthalten.

Die folgenden Operationen werden unterstützt und können unter "Operation" ausgewählt werden:

- Zuweisung: Der erste Operand wird auf den Wert des zweiten Operanden gesetzt.
- Addition: Der erste Operand um den Wert des zweiten Operanden erhöht.
- Subtraktion: Der erste Operand um den Wert des zweiten Operanden verringert.
- Multiplikation: Der erste Operand mit dem Wert des zweiten Operanden multipliziert.
- Division: Der erste Operand durch den Wert des zweiten Operanden geteilt.
- Modulo: Der Rest der Division des ersten Operanden durch den zweiten Operanden wird in den ersten Operanden gespeichert.

Folgende Kombinationen von Operanden und Operator sind erlaubt (mit Zahl sind hier ebenfalls Positionskomponenten gemeint):

	Zuweisung	Plus	Minus	Multiplikation	Div.	Modulo
Beide sind Zahlen	x	x	x	x	x	x
Beide sind Positionen	x	x	x			
Op 1 ist Position, Op 2 ist Zahl				x	x	x
Op 1 ist Zahl, Op 2 ist Position						

### 10.8.6 Variablen beobachten

Sie können die aktuellen Werte aller definierten Variablen der iRC links im Bereich "Variablen" beobachten.

## 10.9 Kamera

Die Kameraanweisung ermöglicht das Abrufen von Objektinformationen von einer Objekterkennungskamera. Zu den Informationen gehören Greifposition und Orientierung, sowie Objekttyp und Erkennungszustand.

Um eine Kamera zu verwenden muss diese im Konfigurationsbereich definiert und kalibriert sein (s. Abschnitt 13.13.7). Die Programmanweisung kann über den Menüeintrag "Sonderbefehle" → "Kamera" hinzugefügt werden.



Abbildung 54: Kameraanweisung im Programmmeditor.

Variablen		Kameras	Statistik
Position und Status		Digitale Ein-/Ausgänge	
<b>Aktionen</b>	▼		
<b>Filter</b>	▼		
<b>Zahlenvariablen</b>	+ ^		
#logicprogramrunning	0.0		
#parts-bad	0.0		
#parts-good	25.0		
#path_index	-1.0		
#programrunning	0.0		
<b>Positionsvariablen</b>	+ ^		
#position			
X:	0,0 mm	Y: 0,0 mm	Z: 139,0 mm
A:	0,0°	B: 0,0°	C: 180,0°
A1:	0,0 mm	A2: 0,0 mm	A3: 0,0 mm
#position-userframe			
X:	0,0 mm	Y: 0,0 mm	Z: 0,0 mm
A:	0,0°	B: 0,0°	C: 0,0°
A1:	0,0 mm	A2: 0,0 mm	A3: 0,0 mm

Abbildung 53: Die Werte der Variablen werden im Infobereich angezeigt.

Unter Typ muss der Typ der Kamera gewählt, unter Name der in der Konfiguration festgelegte Name eingetragen werden. Die Ausgabevariablen für Zielposition und Modellklasse müssen zuvor per Store-Anweisung deklariert worden sein. Die Zielposition enthält die Position und Orientierung des Objekts im Koordinatensystem des Roboters, während die Modellklasse eine Identifikationsnummer für den erkannten Objekttyp enthält. Wenn kein Objekt erkannt wurde ist der Wert der Modellklasse "-1".

Der Befehl "Kameraauslöser" forciert die Kamera eine Erkennung durchzuführen, falls keine kontinuierliche Erkennung eingestellt ist. Dies kann beispielsweise in einem Schleifenblock erfolgen, um kontinuierlich nach einem Objekt zu suchen. Wenn die kontinuierliche Erkennung eingestellt ist, führt die Kamera zyklisch eine Erkennung durch, was gegebenenfalls zu Verzögerungen führen kann.



Der "Kamera"-Befehl ist nur mit Kameras nutzbar, die die Kamera-Schnittstelle verwenden. Andere Kamera-Systeme, die die App-Schnittstelle verwenden, nutzen stattdessen den Befehl "App-Funktion".



Die Kameraanweisung wartet nicht, wenn kein Objekt erkannt wurde. Prüfen Sie durch eine If-Bedingung oder eine Bedingungs Schleife, ob die Kamera tatsächlich ein Objekt erkannt hat!



Die Orientierung des Objekts kann zu einer langsamen Linearbewegung des Roboters führen!

Wenn die Drehung der Werkzeugachse länger dauert als die Bewegung des Werkzeugs zur Objektposition, dann wird die Bewegung entsprechend verlangsamt. Dies geschieht auch wenn keine Werkzeugachse verbaut ist. Falls die Objektorientierung nicht relevant oder keine Werkzeugachse verbaut ist kann dies wie folgt vermieden werden:

1. Ermitteln Sie die Orientierung des Roboters vor dem Anfahren der Objektposition, beispielsweise indem Sie dort eine neue Positionsvariable definieren und mit der aktuellen Position initialisieren. Falls keine Werkzeugachse verbaut ist können Sie die konstanten Orientierungswerte aus dem Informationsbereich von iRC verwenden.
2. Erstellen Sie drei Zuweisungs-Anweisungen (Math-Anweisung) und überschreiben Sie die A, B und C-Komponenten der Zielposition mit den ermittelten Werten.

## 10.10 Appfunktion

Die App-Anweisung ermöglicht die Integration von App-Funktionen in den Programmablauf. Wenn mindestens eine App aktiv ist die Funktionen definiert kann diese in der Anweisung ausgewählt werden. Falls die App Funktionsparameter definiert können diese daraufhin in der Anweisung angegeben werden.

Wenn der Roboter die Anweisung erreicht sendet er den Funktionsaufruf samt Parameter an die App und wartet bis diese den Abschluss der App bestätigt. Die App kann währenddessen beispielsweise auf Daten des Roboters oder angeschlossener Peripherie zugreifen, Berechnungen durchführen, Daten loggen oder an einen Server senden. Sie kann auch Variablen oder globale Signale setzen um mit den folgenden Programmanweisungen zu interagieren.

## 10.11 Audio

Die Audioanweisungen erlauben eine einfache Ausgabe von Tönen und synthetisierter Sprache. In der Simulation werden die Töne über den PC ausgegeben, an der Robotersteuerung standardmäßig über die Miniklinke- oder USB-Buchse an der Seite des Steuerungsmoduls (diese ist möglicherweise nicht bei allen Steuerungen zugänglich).

### 10.11.1 Ton abspielen

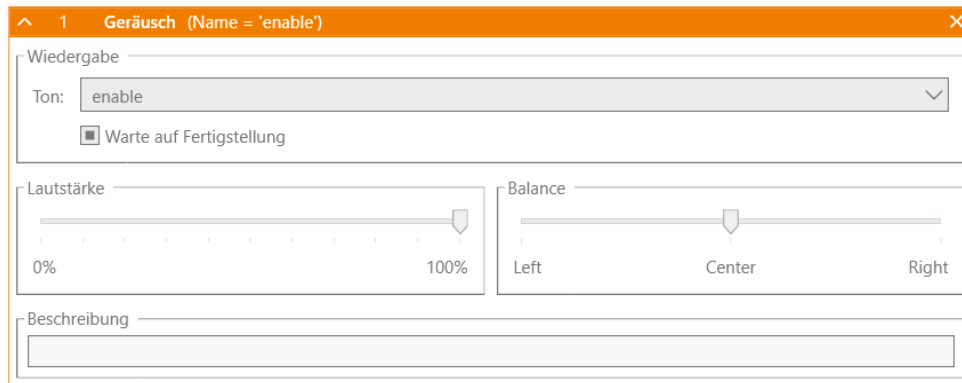


Abbildung 55: Anweisung "Ton abspielen"

Die Anweisung "Ton abspielen" bietet eine Liste vordefiniert Audiodateien, sowie einen Lautstärke- und Balance-Regler (links/rechts). Wird das Feld "Warte auf Fertigstellung" gesetzt wird die folgende Anweisung ausgeführt sobald die Ausgabe abgeschlossen ist, andernfalls spielt der Ton während die folgenden Anweisungen ausgeführt werden. Mehrere Tonausgaben können einander überlagern.

Der vordefinierte Ton "#none" gibt nichts aus.

Um Audiodateien hinzuzufügen müssen diese im WAV-Format im Verzeichnis Data/Audio der Simulation bzw. auf der integrierten Robotersteuerung (per SFTP) abgelegt werden. Zusätzlich müssen sie in der Datei audio.xml im Verzeichnis Data definiert sein. Das Format der Datei ist wie folgt:

```
<?xml version="1.0" standalone="yes"?>
<Audio Mute="0">
  <WaveFile Name="enable" Filename="enable.wav" />
  <WaveFile Name="gripper" Filename="gripper.wav" />
</Audio>
```

### 10.11.2 Sprachausgabe

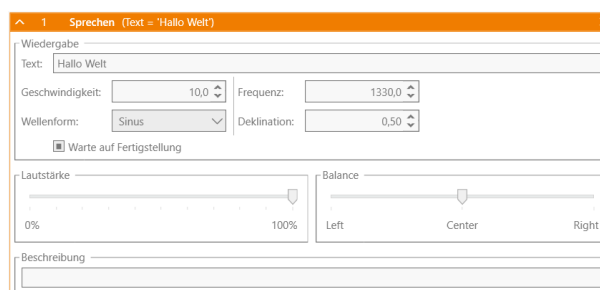


Abbildung 56: Anweisung "Sprachausgabe"

Die Anweisung "Sprechen" ermöglicht eine simple synthetisierte Sprachausgabe.

Geben Sie den zu sprechenden Text im Textfeld ein. Der Wert "Geschwindigkeit" beeinflusst wie schnell der Text gesprochen wird, höhere Werte sind dabei langsamer. Frequenz und Wellenform beeinflussen den Klang der Stimme. Deklination ändert die Betonung über den Verlauf der Ausgabe, beispielsweise klingt ein Wert von -10 fragend. Die Felder Lautstärke, Balance und "Warte auf Fertigstellung" entsprechen denen in Anweisung "Ton abspielen".

## 11 Koordinatensysteme

Wird der Roboter im kartesischen Modus verfahren (siehe 8.4.2), so beziehen sich die der Bewegung zu Grunde liegenden Koordinaten immer auf ein vorher bestimmtes Koordinatensystem. Die iRC bietet die Möglichkeit dieses Koordinatensystem zu ändern. Hierzu stehen zum einen zwei vordefinierte Koordinatensysteme zur Verfügung ("`#base`" und "`#tool`"), desweiteren können eigene, vom Benutzer definierte Koordinatensysteme (Benutzerkoordinatensysteme) gewählt werden. Alle Koordinatensysteme in iRC sind orthonormal.

### 11.1 Vordefinierte Koordinatensysteme

Die vordefinierten Koordinatensysteme stehen immer zur Verfügung und sind durch ein `#` Zeichen als ersten Buchstaben des Namens gekennzeichnet. Sie können weder verändert noch gelöscht werden.

#### 11.1.1 Das Basiskoordinatensystem

Das Basiskoordinatensystem hat den Namen "`#base`". Es handelt sich um das "natürliche" Koordinatensystem des Roboters. Die genaue Lage hängt vom Typ des Roboters ab. In den meisten Fällen hat es seinen Ursprung an der Basis des Roboters.

#### 11.1.2 Das Werkzeugkoordinatensystem

Das Werkzeugkoordinatensystem ("`#tool`") hat seinen Ursprung im Greifpunkt des Roboters. Dieses Koordinatensystem bewegt sich also immer mit dem Werkzeug des Roboters.

### 11.2 Benutzerkoordinatensysteme

Benutzerkoordinatensysteme (kurz "BKS") bieten eine Möglichkeit den Roboter relativ zu einen vom Anwender vorgegebenen Koordinatensystem zu bewegen. Ein BKS wird durch die Vorgabe von drei Punkten (A,B und C) im Raum eindeutig beschrieben:

- Punkt A bezeichnet dabei den Ursprung des Koordinatensystems.
- Punkt B liegt immer in positiver x-Richtung vom Ursprung und definiert somit die Richtung der x-Achse, welche immer von A aus zum Punkt B zeigt.
- Punkt C liegt immer in positiver y-Richtung vom Ursprung. Die y-Achse liegt immer in der von der x-Achse und dem Punkt C festgelegten Ebene und steht rechthwinklig zur x-Achse. Die Richtung der y-Achse ist durch diese Eigenschaften eindeutig bestimmt.
- Die Richtung z-Achse ergibt sich immer als die (eindeutig bestimmte) Richtung, welche sowohl zur x-Achse als auch zur y-Achse derart senkrecht steht, dass sich ein rechtshändiges Koordinatensystem ergibt.

#### 11.2.1 Erstellen eines neuen BKS

Der erste Schritt, um ein neues BKS zu definieren besteht darin den Assistenten zur Erstellung neuer BKS über den Projektkonfigurationsbereich zu starten. Siehe dazu 13.11. Dies geht auch über das "+" Symbol im "Jogging"-Bereich, wenn der Bewegungsmodus auf "Kartesisch" eingestellt ist. Nachdem der Assistent gestartet wurde, führen Sie die folgenden Schritte aus:

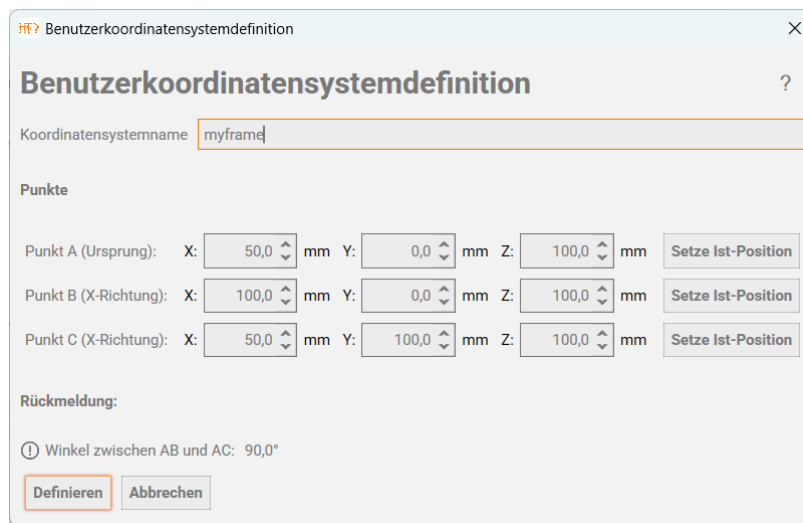


Abbildung 57: Der Assistent zum Erstellen neuer BKS.

1. Bewegen sie den Roboter in eine Pose, in welcher der Greifpunkt an der Stelle liegt an der Punkt A definiert werden soll und klicken Sie danach auf "Setze Aktuellen". Die Koordinaten des gewählten Punktes werden nun neben "Punkt A (Ursprung)" angezeigt.
2. Bewegen sie den Roboter in eine Pose, in welcher der Greifpunkt an der Stelle liegt an der Punkt B definiert werden soll und klicken Sie danach auf "Setze Aktuellen". Die Koordinaten des gewählten Punktes werden nun neben "Punkt B (Pos. x vom Ursprung)" angezeigt.
3. Bewegen sie den Roboter in eine Pose, in welcher der Greifpunkt an der Stelle liegt an der Punkt C definiert werden soll und klicken Sie danach auf "Setze Aktuellen". Die Koordinaten des gewählten Punktes werden nun neben "Punkt C (Pos. y vom Ursprung)" angezeigt.
4. Geben sie den Namen des neuen BKS im Texteingabefeld unter "Name" ein und klicken Sie anschließend auf die Schaltfläche "Definieren", um die Erstellung des neuen BKS abzuschließen.



Bitte beachten Sie, dass in iRC alle Koordinatensysteme rechtwinklig sind. Es gelingt normalerweise nicht, die Punkte A,B und C exakt so zu legen, dass sich daraus ein rechtwinkliges Koordinatensystem ergibt. Aus diesem Grunde wird jedes neu definierte BKS von iRC derart orthonormalisiert, dass ein so gut wie möglich zu den vorgegebenen Punkten A,B,C passendes, rechtwinkliges Koordinatensystem erzeugt wird.

### 11.3 Koordinatensysteme wechseln

Das momentan ausgewählte Koordinatensystem wird im Jogging-Bereich der iRC angezeigt (siehe Abbildung 58), wenn der Bewegungsmodus auf "Kartesisch" eingestellt ist und kann dort per Dropdown-Menü gewechselt werden. Bitte beachten Sie, dass die Auswahl von Koordinatensystemen nur verfügbar ist, wenn zuvor mindestens ein Benutzerkoordinatensystem definiert wurde.

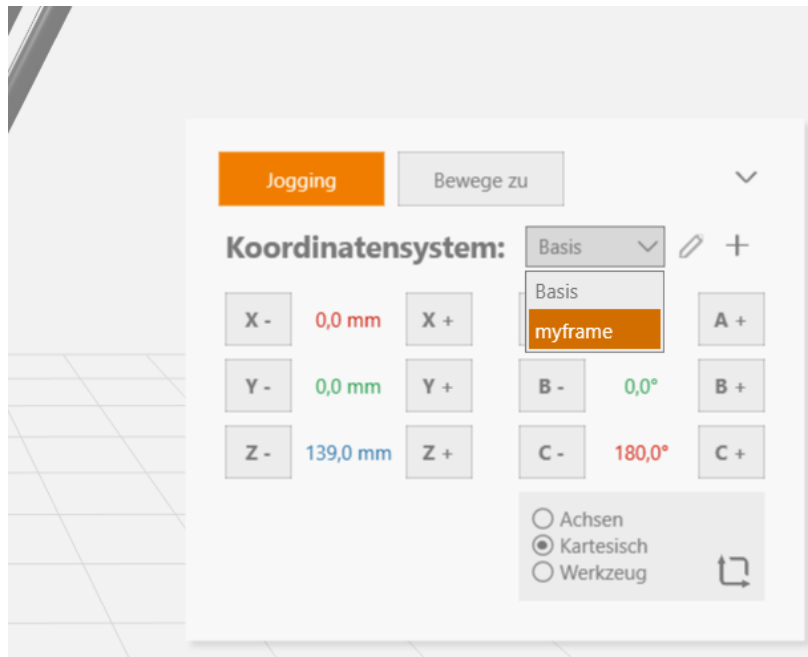


Abbildung 58: Das Dropdown-Menü zur Auswahl von BKS.

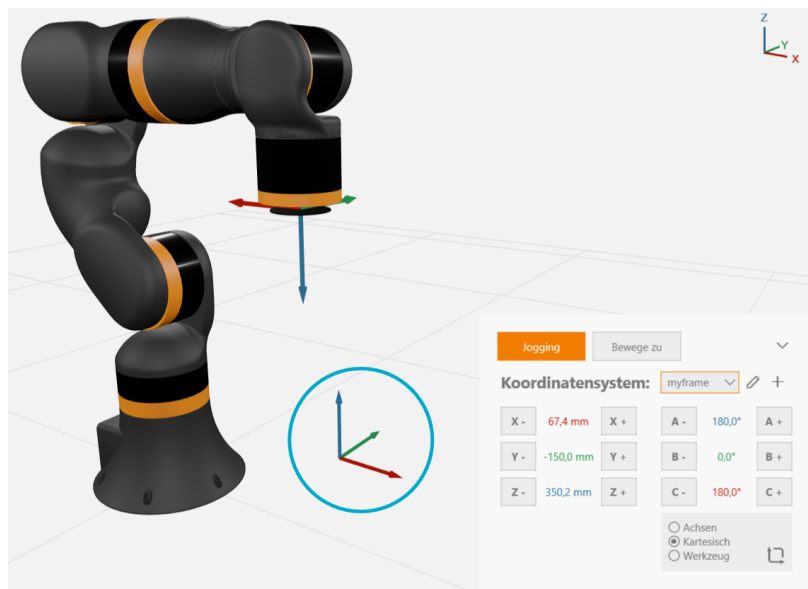


Abbildung 59: iRC zeigt das gerade gewählte Koordinatensystem durch ein farbiges Koordinatenkreuz an.

Im kartesischen Modus zeigt die iRC die Lage des gerade gewählten Koordinatensystems im Raum durch ein farbiges Koordinatenkreuz an (siehe Abbildung 59).

Im kartesischen Modus zeigt der Bereich mit den Softwareschaltflächen zur Bewegung des Roboters



Abbildung 60: Die Softwareschaltflächen zur Bewegung im kartesischen Modus.

das gerade gewählte Koordinatensystem an und auch die angezeigte Position bezieht darauf.

## 12 Hardwarekonfiguration

Um zusätzliche Hardware wie Schalter, SPS, Aktoren, Achsbremsen oder Zusatzachsen verwenden zu können müssen diese zunächst angeschlossen und konfiguriert werden. Dieses Kapitel erklärt die Konfiguration der die Achsen und Ein-/Ausgabe betreffenden Hardware. Weitere Einstellungen die das Verhalten des Roboters und die Softwareschnittstellen betreffen (darunter auch Kameras) finden Sie im Kapitel 13.

### 12.1 Ein-/Ausgänge

Die Steuerung unterstützt verschiedene Arten von Ein- und Ausgängen, die im Konfigurationsbereich unter "Ein-/Ausgänge" konfiguriert werden können:

- Basis-Ein-/Ausgänge: Diese Ein- und Ausgänge mit Nummern von 1-20 sind nur für Roboter der Mover-Serie relevant. Bei DIN-Rail-Steuerungen werden sie ignoriert.
- DIN-Rail Ein-/Ausgänge: Bis zu 4 Ein-/Ausgabemodule können ähnlich der Motormodule über die DIN-Schiene angeschlossen werden. (Ein fünftes Modul kann hinzugefügt werden, jedoch werden dann nur 4 der 7 Ein- und Ausgänge dieses Moduls unterstützt.)
- Globale Signale: Diese sind virtuelle Ein-/Ausgänge. Sie können über Roboterprogramme, die CRI-Schnittstelle oder die Modbus-Schnittstelle gesetzt und gelesen werden. Es können bis zu 100 globale Signale verwendet werden.

#### 12.1.1 Elektrische Integration

Der einfachste Weg, eine speicherprogrammierbare Steuerung (SPS) anzuschließen ist über digitale Ein- und Ausgänge. Jede Robotersteuerung wird mit einem DIO-Modul geliefert. Dieses bietet 7 Eingänge und 7 Ausgänge. Wenn zusätzliche Ein- und Ausgänge erforderlich sind, können bis zu vier zusätzliche DIO-Module integriert werden. Insgesamt können bis zu 5 DIO-Module angesteuert werden (das fünfte jedoch nur eingeschränkt).

Die Ausgänge werden über Halbleiterrelais gesteuert und können bis zu 500 mA schalten. Dieser Wert darf während des Schaltvorgangs nicht überschritten werden (z.B. durch Ladeströme von Kondensatoren), um eine Beschädigung der Relais zu verhindern.

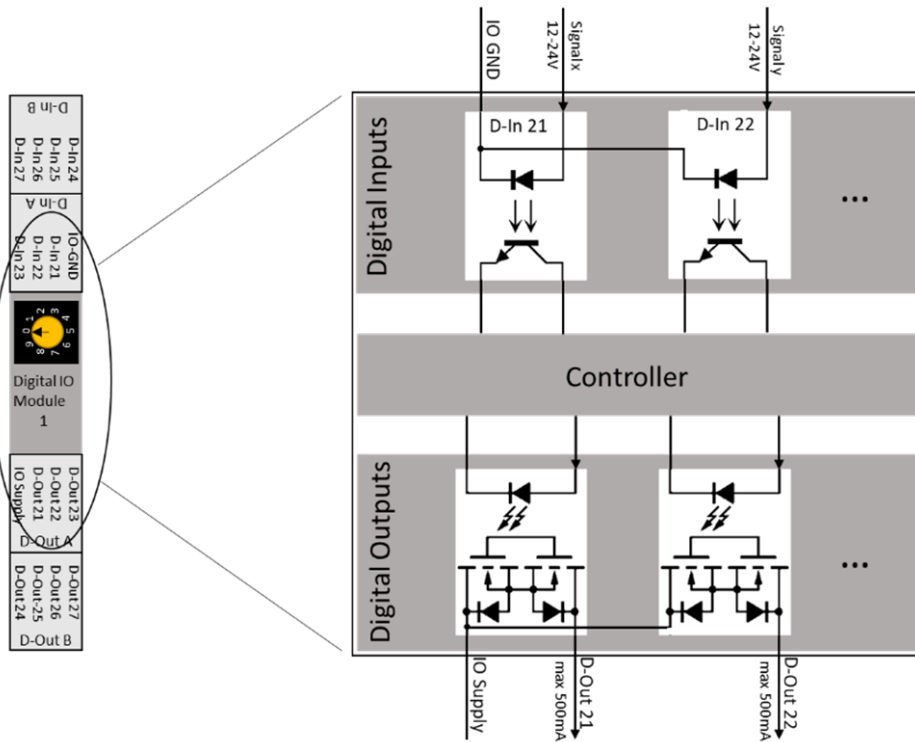


Abbildung 61: Interner Aufbau eines DIO-Moduls

Die Ein- und Ausgänge sind von der Robotersteuerung galvanisch getrennt. Eine Stromversorgung (im Bild oben mit "IO Supply" bezeichnet) muss angeschlossen werden. Die integrierte 24 V-Versorgung kann ebenfalls verwendet werden.

Um DIN-Rail-Module hinzuzufügen bauen Sie diese zunächst in den Schaltschrank ein und stellen Sie die CAN-IDs entsprechend der Tabellen 9 ein.

### 12.1.2 Software-Konfiguration

Schalten Sie dann die Steuerung ein und verbinden Sie die iRC. Geben Sie im Konfigurationsbereich die Anzahl der Ein-/Ausgabemodule ein. Speichern Sie die Konfiguration und Starten Sie nach 30 Sekunden die Steuerung samt iRC neu. Falls andere CAN-IDs gewünscht sind können diese manuell in der Projektkonfigurationsdatei eingetragen werden.

Schalterposition	Modulnummer	CAN-ID (dez.)	CAN-ID (hex.)
0	1	112	0x70
1		120	0x78
2	2	128	0x80
3		136	0x88
4	3	144	0x90
5		152	0x98

Schalterposition	Modulnummer	CAN-ID (dez.)	CAN-ID (hex.)
6		160	0xA0
7		168	0xA8
8		176	0xB0
9		184	0xB8

Tabelle 9: Standardmäßige CAN-ID-Schalterstellungen an CPR-DIO-Modulen



Wenn mehr Module definiert als verbaut sind wird die Steuerung einen Modulfehler melden und erst nach Beheben des Fehlers benutzbar sein.

Je Ein- und Ausgang können Sie einen beschreibenden Namen angeben. Dieser ist nur für die Anzeige, jedoch nicht als Name in Programmbedingungen relevant. Der Reset-Zustand gibt an in welchem Zustand ein Ausgang bei Reset wechselt, der Fehlerzustand wird angenommen, wenn ein Fehler auftritt.

### 12.1.3 Sensoren und Taster anschließen

- Pin 1 (GND) des "D-in A"-Anschlusses muss mit dem negativen Pol (-) der Stromversorgung des Sensors verbunden werden.
- Das Sensorsignal (positiv) muss an einen Eingangspin angeschlossen werden: "D-in A" Stecker Pins 2-4 oder "D-in B" Stecker Pins 1-4.
- Die positive Seite des Sensors (+) muss an die Stromversorgung des Sensors angeschlossen werden.
- Der Status der Eingänge kann im Register "Digitale Ein-/Ausgänge" links in der iRC überwacht werden.
- Ein Roboterprogramm kann Eingaben abfragen und darauf reagieren, z.B. mit einer if-then-else-Anweisung.

### 12.1.4 Aktoren anschließen

- Pin 1 (Versorgungsspannung) des Steckers "D-out A" muss mit einer Stromversorgung (z.B. 24V) verbunden werden.
- Der Aktor (Relais usw.) wird dann über einen freien Pin der "D-out"-Stecker (D-out A Pin 2-4 und D-out B Pin 1-4) mit Strom versorgt. Maximal darf die Stromstärke von 500 mA nicht überschritten werden.
- Sie können die Ausgaben manuell im Register "Digitale Ein-/Ausgänge" links in der iRC einstellen.
- Ein Roboterprogramm kann den Zustand der Ausgänge mit der "Digitaler Ausgang"-Anweisung einstellen.

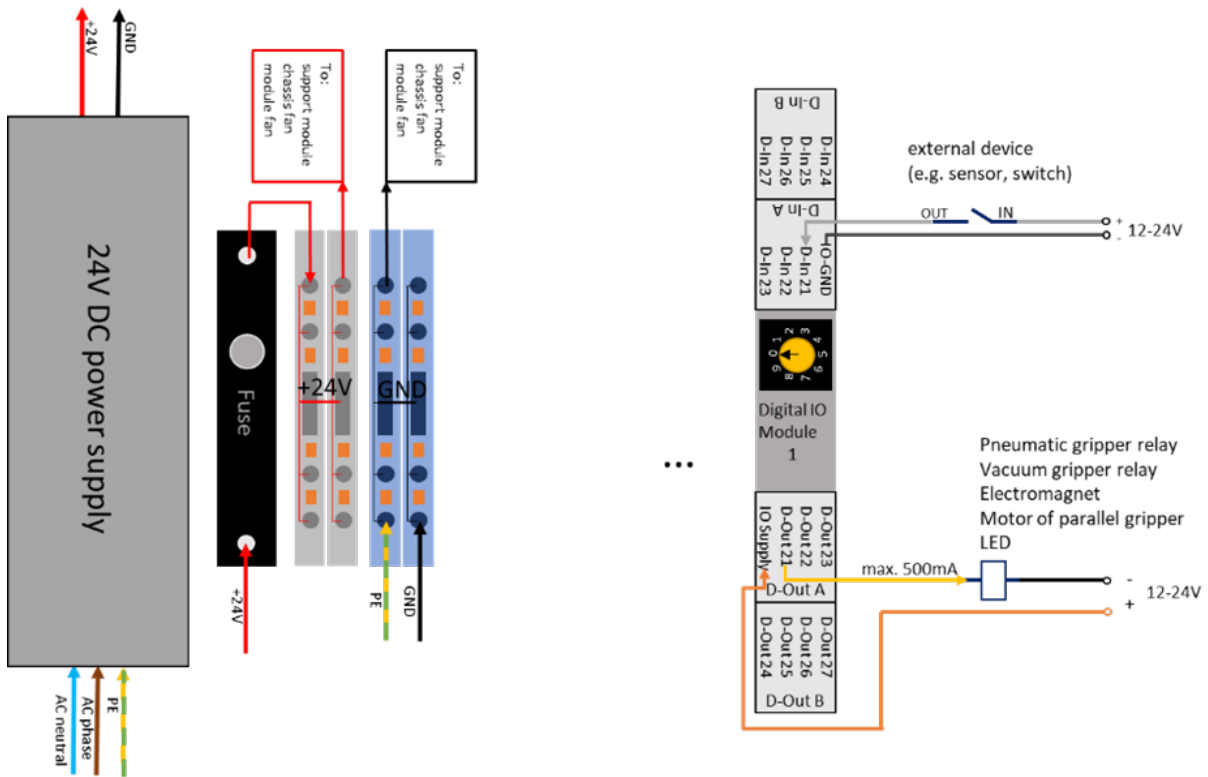


Abbildung 62: Beispiel: Anschluss eines Sensors und eines Aktors

### 12.2 Roboterkonfiguration - Achslängen

Die Roboterkonfiguration enthält robotertypspezifische Parameter. Änderungen beeinflussen daher alle Projekte des jeweiligen Typs.

Die Achslängen von Portalrobotern können im Konfigurationsbereich (Datei → Konfiguration → Portal) geändert werden. Abhängig von der Achszahl und -konfiguration stehen dafür verschiedene Parameter zur Verfügung.

Parameter	Bedeutung
Gesamte Achslänge	Länge der Achsen in der 3D-Simulation
Bewegungslimits	Nutzbare Länge der Achsen



Nach dem Ändern der Achslängen muss die Roboterkonfiguration gespeichert und neu geladen werden. Ein Roboter mit integrierter Steuerung muss nach mindestens 30s neu gestartet werden.

### 12.3 Motorbremse

Um ein Absacken der Achsen zu verhindern können elektromagnetische Bremsen verwendet werden, die durch Anlegen einer Spannung die Achse freigeben. Ohne Spannung fallen sie durch Federn in den Bremszustand.



Die hier beschriebene Integration bezieht sich nur auf Stepper-Motoren. BLDC-Motoren werden direkt vom Motorcontroller-Modul gesteuert ("GND"-Pin 5 am "Motor"-Eingang). Die Konfiguration erfolgt über die Firmwareparameter (siehe Abschnitt 12.5).

#### 12.3.1 Elektrische Integration



Die in den Motor integrierten Standardbremsen arbeiten unter 24V. Beachten Sie aber bitte die Dokumentation der Motoren.

Um die Bremsen zu lösen müssen zwei Voraussetzungen erfüllt sein:

- Der NotAus muss freigegeben sein
- Die Steuerung muss die Bremse freigeben

Die elektrische Umsetzung erfolgt über zwei in Serie geschaltete Relais. Zusätzlich kann der NotAus-Status als Digitaler Eingang eingelesen werden um auch ein sehr schnelles Auslösen/Freigeben des NotAus zu behandeln.

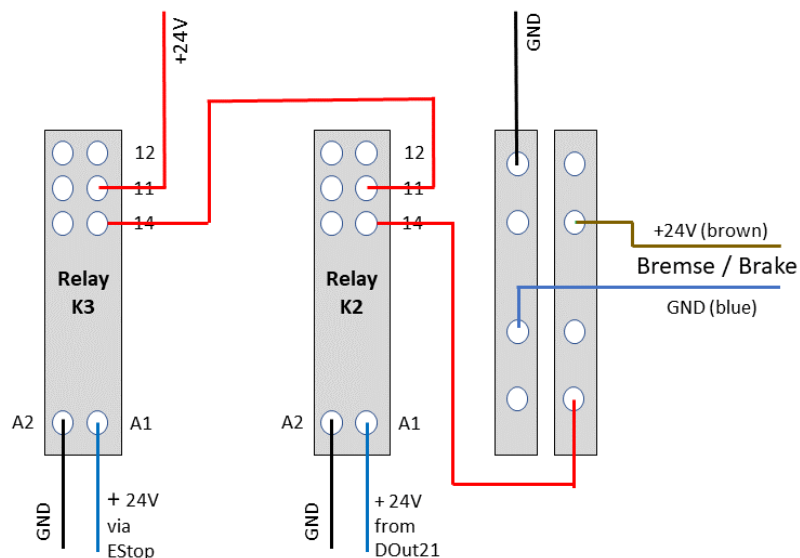


Abbildung 63: Anschluss Bremsrelais und Bremse

Für weitere Details ziehen Sie bitte die auf dem Wiki verfügbaren Schaltpläne zu Rate.

### 12.3.2 Software-Konfiguration

Unter "Datei" → "Konfiguration" → "Bremsen" der Software kann ein digitaler Ausgang für die Motorbremse festgelegt werden. Der Ausgang wird beim Freigeben der Motoren aktiv geschaltet, sodass er ein Relais ansteuert und darüber die Bremsen freigeben kann. Im Fehlerfall wird der Ausgang deaktiviert.

Der Parameter "Freigabeverzögerung" gibt an wie lange die Bremsen nach Aktivieren der Motoren noch im Bremszustand bleibt. Dies ermöglicht den Motoren erst ein Haltemoment aufzubauen bevor sie die Achslast tragen müssen. Der Standardwert ist 500ms.

Die Notausüberwachung ermöglicht die Kopplung eines digitalen Eingangs mit dem 24V-Notausrelais um einen Spannungsverlust schneller zu erkennen (die Achsmodule erkennen einen Spannungsverlust aufgrund der verbauten Kondensatoren teils erst verzögert - dies kann zu Problemen führen wenn der Notaus sehr schnell wieder gelöst wird). Die Bremsen werden schneller aktiviert und die digitalen Ausgänge in ihre Fehlerzustände zurückgesetzt.



Nach dem Ändern der Bremskonfiguration muss die Roboterkonfiguration gespeichert und neu geladen werden. Ein Roboter mit integrierter Steuerung muss nach mindestens 30s neu gestartet werden.

### 12.4 Zusatzachsen

Die Robotersteuerung unterstützt bis zu 3 Zusatzachsen (in der iRC auch "externe Achse" genannt). Diese können entsprechend der Roboterachsen eingebaut und über den Konfigurationsbereich von iRC konfiguriert werden.

Beim Einbau der Achsmodule muss die CAN-ID festgelegt werden. Bei Motormodulen von Common-place Robotics geschieht dies über den Drehschalter oben am Modul, Tabelle 11 beschreibt die standardmäßige Belegung. Noch nicht vergebene CAN-IDs können für Zusatzachsen verwendet werden.

Schalterposition	Roboterachse	CAN-ID (dez.)	CAN-ID (hex.)
0	1	16	0x10
1		24	0x18
2	2	32	0x20
3		40	0x28
4	3	48	0x30
5		56	0x38
6	4	64	0x40
7		72	0x48
8	5	80	0x50
9	6	88	0x58

Tabelle 11: Standardmäßige CAN-ID-Schalterstellungen an CPR-Motormodulen

Nach Einschalten der Steuerung und Verbinden der iRC kann im Konfigurationsbereich ("Datei" → "Konfiguration" → "Externe Achsen") die Anzahl der Zusatzachsen angegeben werden. Daraufhin

erscheint für jede Achse ein Parametersatz mit folgenden Einträgen:

Parameter	Bedeutung
Bewegungsmodus	Legt fest ob die Achse im Positions- oder Geschwindigkeitsmodus arbeitet. Im Geschwindigkeitsmodus kann eine konstante Geschwindigkeit im Roboterprogramm gesetzt werden, Positionsanweisungen werden jedoch ignoriert. Dies ist bspw. für Förderbänder nützlich.
CAN-ID	CAN-ID des Achsmoduls (dezimal, s.o.). Kann alternativ auch per "Nummer am Modul" eingestellt werden.
Encoderschritte je Einheit	Übersetzungsfaktor von Millimeter bzw. Grad zu Encoderschritte des Motors (Berechnung siehe unten)
Position min/max	Bewegungslimits (für Positionsmodus)
Max. Geschwindigkeit	Höchstgeschwindigkeit
Beschleunigung	Beschleunigung (für Positionsmodus)
Ruck	Steigerung der Beschleunigung (für Positionsmodus)
Steuerungstyp	Art des Motors. Wird nur zur korrekten Berechnung von Motorströmen und Temperatur benötigt.



Die Geschwindigkeit im Geschwindigkeitsmodus ist nicht präzise. Testen Sie die Konfiguration mit geringem Override und passen Sie den Geschwindigkeitsparameter gegebenenfalls an.



Nach dem Ändern der externen Achsen muss das Projekt gespeichert und neu geladen werden. Ein Roboter mit integrierter Steuerung muss nach mindestens 30s neu gestartet werden.

Die Rechenformel 1 zeigt die Berechnung der Encoderschritte je Einheit aus der Anzahl der Encoderschritte, der Getriebeübersetzung und der zurückgelegten Strecke einer vollen Abtriebsumdrehung in beliebiger Einheit. Beispiel 2 zeigt die Berechnung einer Rotationsachse, wie sie beispielsweise als Achse 4 eines RL-DP-5 verbaut ist. Beispiel 3 zeigt die Berechnung einer Linearachse ohne Getriebe.

$$\frac{4 \cdot \text{Encoderschritte} \cdot \text{"Übersetzung"}}{\text{Strecke}} = \text{Encoderschritte je Einheit} \quad (1)$$

$$\frac{4 \cdot 500 \cdot 38}{360^\circ} = 211,111 \quad (2)$$

$$\frac{4 \cdot 500 \cdot 1}{70\text{mm}} = 28,57 \quad (3)$$

Tabelle 13 enthält einige Beispiele für Linear- und Rotationsachsen.

Achstyp	Untersetzung	Encoderschritte je Einheit
igus ZLW-1040	70	28.471
igus GRW-0630	45	44.444
igus SAW-1040	2	1000.00
igus ZLW-20200	144	41.667
igus RL-D-50	48	266.67
igus RL-D-30	50	277.78
igus RL-D-20	38	211.11
igus RL-X0223	1	5.56

Tabelle 13: Beispielhafte Werte verschiedener Linear- und Rotationsachsen

## 12.5 Konfiguration der Motorsteuerungen

Für die Feineinstellung der Bewegung und der Referenzierung enthält jedes Achsmodul einen eigenen Konfigurationssatz. Dieser kann über die Schaltflächen "Datei" → "Firmwarekonfiguration" abgerufen und geändert werden.

Eine detaillierte Beschreibung der Parameter kann unter folgendem Link gefunden werden:

[https://wiki.cpr-robots.com/index.php/Firmware\\_Parameter\\_Configuration](https://wiki.cpr-robots.com/index.php/Firmware_Parameter_Configuration)



Ändern Sie die Firmwareparameter nur wenn Sie wissen was Sie tun! Testen Sie den Roboter mit langsamer Geschwindigkeit und beobachten Sie die Temperaturen der Elektronikmodule und Motoren.

## 13 Softwarekonfiguration

Das Verhalten des Roboters kann über die Konfiguration geändert werden. Die wichtigsten Parameter finden Sie im Konfigurationsbereich der iRC - igus Robot Control, der über "Datei" → "Konfiguration" geöffnet werden kann (siehe Abb. 64).

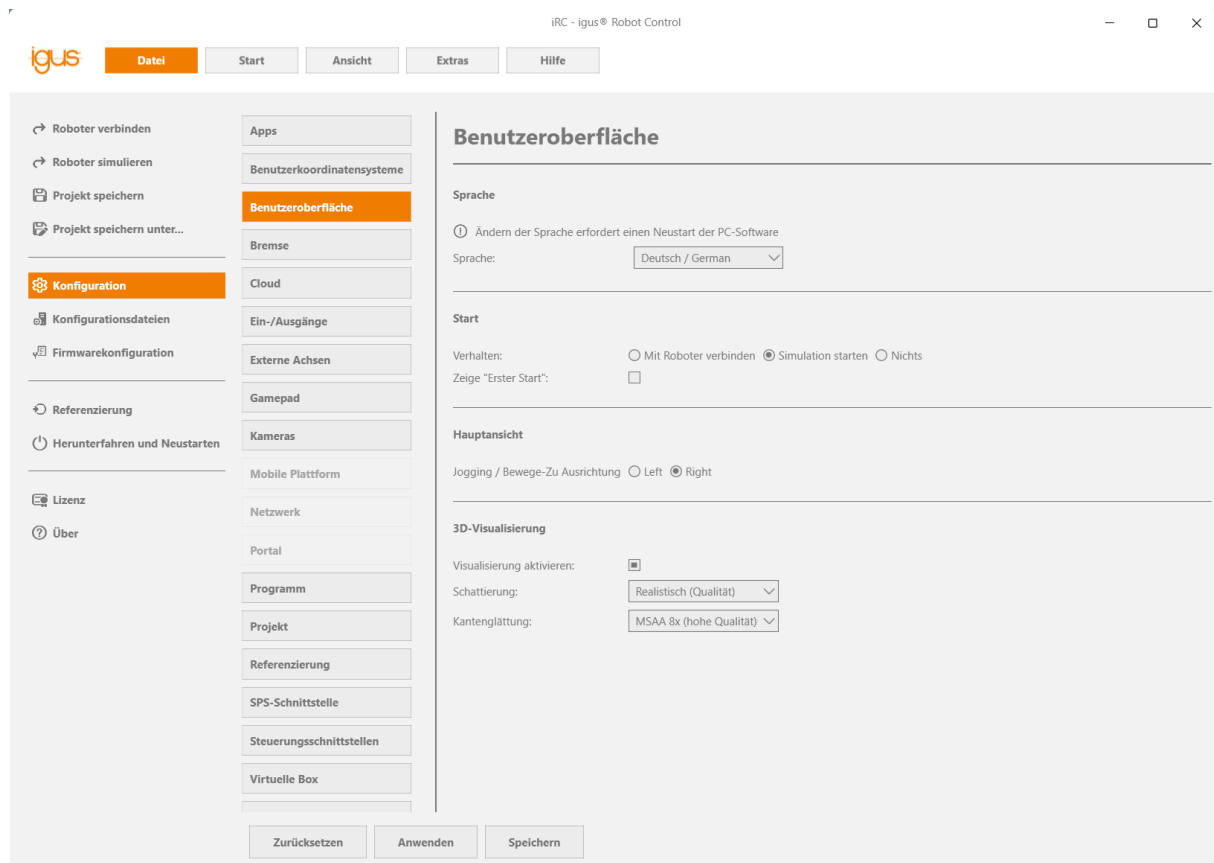


Abbildung 64: Der Projektkonfigurationsbereich.

Speziellere Einstellungen können über die Projekt-, Roboter-, und Werkzeugkonfigurationsdateien vorgenommen werden (s. Abschnitt 13.14). Die Einstellungen der Achsmodule lassen sich über "Firmwarekonfiguration" abrufen und ändern (s. Abs. 12.5).



Ändern Sie die Konfigurationsdateien nur wenn Sie wissen was Sie tun! Testen Sie den Roboter vorsichtig, da er sich unerwartet schnell bewegen oder kollidieren könnte! Änderungen der Firmwareparameter können zur Überhitzung der Motoren oder der Elektronik führen!



Einige Änderungen der integrierten Robotersteuerung werden erst nach einem Neustart übernommen. Warten Sie nach dem Übertragen mindestens 30 Sekunden und starten Sie den Roboter neu.

### **13.1 Benutzeroberfläche**

In den allgemeinen Einstellungen finden Sie roboterunabhängige Einstellungen der iRC wie beispielsweise die Sprache der Benutzeroberfläche, das Startverhalten der Software oder die Anzeigeoptionen.

### **13.2 Projekt**

Hier können Sie den Roboter benennen und den Autor der Konfiguration angeben. Der Robotername wird in der Liste der realen Roboter verwendet um Ihnen zu helfen Ihre Roboter zu unterscheiden. Den Konfigurations-Autor können Sie optional angeben um zu vermerken wer den Roboter konfiguriert hat oder wer für ihn zuständig ist.

### **13.3 Bremse**

Hier können Sie die Bremsen des Roboters per DOut-Ansteuerung konfigurieren und die Freigabeverzögerungszeit einstellen. Zudem können sie hier die Notausüberwachung konfigurieren.

### **13.4 Externe Achsen**

Hier können Sie die Anzahl der externen Achsen und deren Verhalten einstellen. Externe Achsen sind alle Achsen, die nicht zur Roboterkinematik gehören, wie z.B. Förderbänder oder Drehtische.

### **13.5 Ein-/Ausgänge**

Hier kann die Anzahl der Ein-/Ausgabemodule und das Verhalten der Ein-/Ausgänge eingestellt werden. Die globalen Signale sind interne Merker zur Kommunikation zwischen Roboterprogrammen und SPS.

Zu jedem digitalen Ausgang kann angegeben werden welcher Zustand bei Reset und im Fehlerfall angenommen wird.

### **13.6 Programm**

Hier können Sie das Roboter- und Logikprogramm, die Bewegungsgeschwindigkeit (in Prozent der Höchstgeschwindigkeit), den Abspielmodus und die Reaktion bei Programmfehlern festlegen. Zudem können Sie hier die Anzahl an Bewegungsbefehlen einstellen, die maximal in einem Roboterprogramm miteinander überschritten werden können.

### **13.7 Mobile Plattform**

Hier finden Sie Einstellungen für die mobile Plattform, wie z.B. die Aktivierung von Lidar Sensoren oder die Verwendung der SLAM Algorithmen.

### **13.8 Portal**

Über den Menüpunkt "Portal" können Sie die Größe Ihres Portals einstellen, damit die Bewegungen des Roboters entsprechend angepasst werden können. Die "Gesamte Achslänge"-Einstellung passt die 3D-Darstellung an. Die "Bewegungslimits" definieren die Grenzen des Portals, die der Roboter nicht verlassen darf. Bei Überschreiten dieser Grenzen stoppt die Bewegung.



Wenn der Roboter nicht Referenziert ist, werden die Bewegungslimits möglicherweise nicht zur tatsächlichen Position des Roboters passen. Joggen Sie den Roboter vorsichtig, um Kollisionen zu vermeiden.

### 13.9 Referenzierung

Das Referenzierungsprogramm ist nützlich um die Genauigkeit von Roboter die über einen Absolutencoder referenzieren zu verbessern. Roboter die über einen Referenzschalter referenzieren oder Achsen an der aktuellen Position auf 0 setzen haben eher wenig Nutzen hieran. Wenn ein Referenzierungsprogramm festgelegt ist werden zunächst alle Achsen referenziert, dann das Programm ausgeführt um an eine benutzerdefinierte Position zu fahren, an welcher alle Achsen erneut referenziert werden.



Damit das Referenzierungsprogramm gestartet werden kann muss der Roboter bereits referenziert sein. Das Referenzierungsprogramm kann nicht dazu verwendet werden einen Referenzierungsablauf zu definieren oder zwischen dem Referenzieren mehrerer Achsen einem Hindernis auszuweichen.



Wenn Sie ein Referenzierungsprogramm verwenden beobachten Sie die Referenzierung und halten Sie den Notaus bereit. Der Roboter wird sich bewegen. Bei Fehler in der ersten Referenzierung wird möglicherweise eine unerwartete Position angefahren.

Als Referenzierungsprogramm legen Sie einfach ein normales Roboterprogramm an das im einfachsten Fall nur eine Achs-Anweisung enthält. Um Getriebespiel auszugleichen sollte eine Position angefahren werden an der alle Achsen unter leichter Last stehen, d.h. der Roboter sollte nicht z.B. in Kerzenposition stehen.

Setzen Sie das Feld "Nach Referenziere alle ausführen" um das Referenzierungsprogramm bei der normalen Referenzierung (z.B. auch über die SPS-Schnittstelle, Modbus oder CRI) auszuführen. Wenn das Feld nicht gesetzt ist kann es nur über die Schaltfläche im Referenzierungsbereich oder über die entsprechende CRI-Funktion ausgeführt werden.

### 13.10 Werkzeug

Hier kann das montierte Werkzeug festgelegt werden. Das Ändern des Werkzeug erfordert ein Neuladen des Projekts beziehungsweise den Neustart der integrierten Steuerung.

Neue Werkzeuge können als Konfigurationsdatei im Verzeichnis "Data/Tools" definiert werden. Neue und geänderte Werkzeuge werden nicht automatisch mit der integrierten Steuerung synchronisiert. Um Änderungen zu Übertragen wählen Sie "Hochladen..." und wählen Sie die neue oder geänderte Werkzeugkonfigurationsdatei aus.

Mehr hierzu finden Sie im Wiki:

[https://wiki.cpr-robots.com/index.php/Tool\\_Configuration](https://wiki.cpr-robots.com/index.php/Tool_Configuration)



### 13.11 Benutzerkoordinatensysteme

Hier können sie die Liste bestehender Benutzerkoordinatensysteme einsehen und nach Bedarf erweitern. Um ein neues Benutzerkoordinatensystem zu definieren, klicken Sie auf die Schaltfläche "Erstellen", woraufhin der Assistent zum Erstellen eines neuen Benutzerkoordinatensystems startet. Ein bestehendes Benutzerkoordinatensystem kann entfernt werden, indem Sie es erst in der Liste markieren und danach auf die Schaltfläche "Löschen" klicken. Weiterführende Informationen zu benutzerdefinierten Koordinatensystemen finden sie in Abschnitt 11.

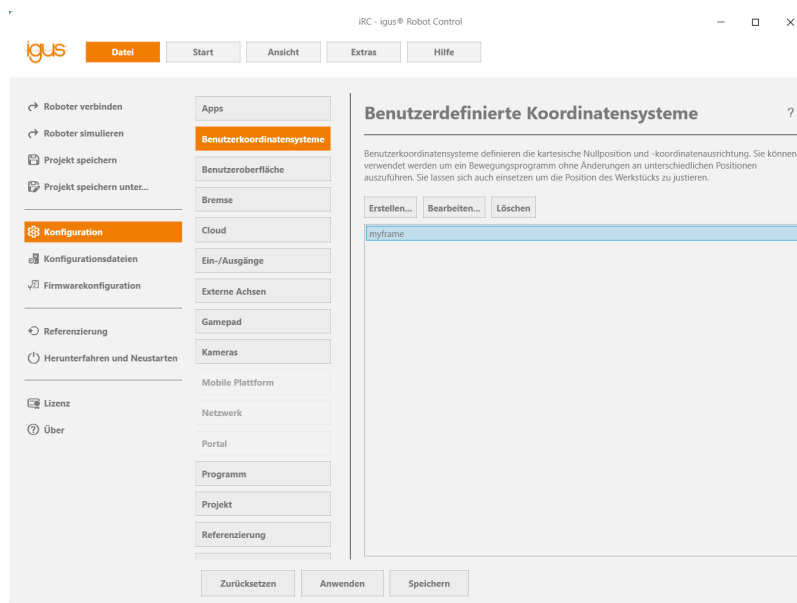


Abbildung 65: Konfiguration von BKS.

### 13.12 Virtuelle Box

Die virtuelle Box definiert einen Bereich, den der Werkzeugmittelpunkt des Roboters nicht verlassen darf. Bei Überschreiten der Grenzen stoppt die Bewegung.

### 13.13 Schnittstellen

Zur Kommunikation mit und Steuerung durch andere Software und Geräte bietet iRC eine SPS-Schnittstelle, die CRI-Ethernet-Schnittstelle und eine Schnittstelle zur Anbindung von ifm O2D-Kameras.

#### 13.13.1 SPS-Schnittstelle

Die SPS-Schnittstelle ermöglicht die Ausführung von Basisfunktionen und die Signalisierung von Zuständen mittels digitaler Ein- und Ausgänge. Neben der Steuerung durch eine SPS ermöglicht diese Schnittstelle auch die Bedienung durch Hardwaretaster.

Die SPS-Schnittstelle erfordert freie digitale Ein- bzw. Ausgänge. Sie reagiert auf steigende Flanken an Eingängen.

Folgende Eingabefunktionen werden unterstützt:

Parameter	Bedeutung
Motoren aktivieren / deaktivieren	Setzt die Steuerung zurück und Aktiviert dann die Motoren. Ist die Steuerung aktiv, wird sie deaktiviert.
Referenzierung starten	Startet die Referenzierung aller Achsen
Programm starten / stoppen	Startet die Ausführung des geladenen Programms, stoppt die Ausführung eines laufenden Programms oder führt ein pausiertes Programm fort
Programm pausieren / fortführen	Pausiert ein laufendes Programm oder führt ein pausiertes Programm fort
Ein-Tasten-Steuerung Start	Ein-Tasten-Steuerung: Führt bei mehrmaligen Drücken des Tasters nacheinander die folgenden Funktionen aus: Reset, Aktivieren, Referenzieren (wenn noch nicht erledigt), Programm starten
Ein-Tasten-Steuerung Stop	Ein-Tasten-Steuerung: Führt bei mehrmaligen Drücken des Tasters nacheinander die folgenden Funktionen aus: Pause, Stopp, Reset
Herunterfahren	Führt den Steuerungsrechner herunter
Plattformmission starten	Startet die Mission der mobilen Plattform
Achsanweisung hinzufügen	Achs-Anweisung zur aktuellen Position im Programmeditor hinzufügen
Linearanweisung hinzufügen	Linear-Anweisung zur aktuellen Position im Programmeditor hinzufügen



"Achs-Anweisung hinzufügen" und "Linear-Anweisung hinzufügen" können nur eingesetzt werden, wenn iRC mit dem Roboter verbunden ist.

Folgende Ausgabefunktionen werden unterstützt:

Parameter	Bedeutung
Kein-Fehler	Ausgang ist aktiv, wenn der Roboter im fehlerfreien Zustand ist

Parameter	Bedeutung
Fehler	Ausgang ist aktiv, wenn der Roboter im fehlerfreien Zustand ist
Roboter ist referenziert	Ausgang ist aktiv, wenn alle Achsen referenziert sind
Programm läuft	Ausgang ist aktiv, wenn ein Programm läuft
Programm läuft nicht	Ausgang ist aktiv, wenn kein Programm läuft
Plattform-Mission läuft	Die mobile Plattform führt eine Mission aus

Die Konfiguration der SPS-Schnittstelle erfolgt über den Konfigurationsbereich in iRC (Datei → Konfiguration → SPS-Schnittstelle). Um einen Roboter mit integrierter Steuerung zu konfigurieren muss dieser verbunden sein. Das Feld "Aktiv" aktiviert die Schnittstelle. Die Zahlenfelder zu den Ein- und Ausgängen entsprechen den Nummern der digitalen Ein-/Ausgänge. Um einzelne Funktionen zu deaktivieren, nutzen Sie die Checkbox neben der entsprechenden Funktion.

### 13.13.2 Programmwahl über digitale Eingänge

Über digitale Eingänge oder globale Signale können Roboterprogramme geladen und gestartet werden. Dies ist beispielsweise nützlich wenn über Taster oder die CRI-GSig-Anweisung ein Programm aus einer vorgegebenen Auswahl gewählt werden soll. Die Konfiguration dazu ist im Konfigurationsbereich der SPS-Schnittstelle im Abschnitt "Programmauslöser" zu finden.

Für jeden Programmauslöser können folgende Parameter angegeben werden:

Parameter	Bedeutung
Aktiv	Schaltet den Programmauslöser frei
Typ	Typ des Eingangssignals: digitaler Eingang (DIn) oder globales Signal (GSig)
Nummer	Nummer des Eingangs. Z.B. 21 für DIn21 bzw. GSig21
Programmtyp	Roboterprogramm oder Plattformmission
Programmdatei	Programm- oder Missionsdatei. Muss im Programs- oder Missions-Verzeichnis liegen

### 13.13.3 Modbus

Über die Modbus-TCP-Schnittstelle können beispielsweise SPS Daten und Anweisungen an die Robotersteuerung senden und Zustandsinformationen empfangen. Weitere Informationen zur Verwendung und Lizenzierung dieser Schnittstelle finden Sie in Abschnitt 14.

Die Modbus-Schnittstelle kann über "Datei → Konfiguration → Steuerungsschnittstellen" aktiviert werden. Anders als die anderen Konfigurationsparameter werden diese Einstellungen nicht mit dem angeschlossenen System gespiegelt, um zu verhindern dass die Konfiguration der integrierten Steuerung und des Steuerungssoftware auf dem Windows-PC miteinander kollidieren. Aus diesem Grund beziehen sich die angezeigten Parameter wenn eine Robotersteuerung angeschlossen ist auf diese, wenn keine angeschlossen ist auf die PC-Software.

Folgende Parameter können konfiguriert werden:

Parameter	Bedeutung
Aktiviert	Aktiviert den Modbus-Server
Port	TCP-Port des Modbus-Servers
Maximale Verbindungen	Maximale Anzahl gleichzeitiger Verbindungen zum Server (nur integrierte Steuerung)

#### 13.13.4 CRI-Schnittstelle

Die CRI-Schnittstelle ermöglicht das Senden komplexer Anweisungen und den Abruf von Informationen und Einstellungen über die Ethernetschnittstelle per TCP/IP. Die iRC verwendet diese Schnittstelle um sich mit Robotern mit integrierter Steuerung zu verbinden. Diese Schnittstelle ist am Roboter und in der Simulation dauerhaft aktiv und benötigt keine Konfiguration.



Eine Dokumentation aller unterstützten Anweisungen sowie Beispielcode kann unter folgendem Link gefunden werden:

[https://wiki.cpr-robots.com/index.php/CRI\\_Ethernet\\_Interface](https://wiki.cpr-robots.com/index.php/CRI_Ethernet_Interface)



Beachten Sie das Aktiv-/Passiv-System des CRI: Zu jeder Zeit kann nur ein CRI-Client den Roboter oder die Simulation steuern (aktiv), alle weiteren Clients können lediglich den Zustand beobachten (passiv). Dies schließt auch die grafische Oberfläche von iRC ein. Es wird immer der erste Client aktiv der sich mit der Robotersteuerung verbindet. Um eine Anwendung manuell aktiv zu schalten senden Sie die CRI-Anweisung SetActive (siehe CRI-Doku), iRC wird bei Klick auf "Zurücksetzen" aktiv.

#### 13.13.5 Positionsschnittstelle

Die Positionsschnittstelle erweitert die CRI-Schnittstelle um Funktionen für schnelle Positionsupdates. Sie lässt sich nur über die CRI-Schnittstelle aktivieren und verwenden. Nach dem aktivieren öffnet sie einen separaten TCP/IP-Port an dem sich maximal ein Client verbinden kann. Die Schnittstelle sendet die Ist-Position sobald diese neu berechnet wurde (je nach Roboter 10-20ms). Über eine CRI-Anweisung lässt sich die Positionsschnittstelle als Soll-Positionsquelle wählen, der Roboter folgt dann den regelmäßig (10-500ms) gesendeten Soll-Positionen.



Die Dokumentation für die Positionsschnittstelle finden Sie unter dem folgenden Link:

[https://wiki.cpr-robots.com/index.php/Position\\_Interface](https://wiki.cpr-robots.com/index.php/Position_Interface)



### 13.13.6 App-Schnittstelle

Die App-Schnittstelle ermöglicht die Erweiterung der Robotersteuerung um eigene oder fremde Softwarekomponenten. Diese Komponenten können auf Daten der Steuerung zugreifen und Funktionen definieren die aus Roboterprogrammen heraus aufgerufen werden können. Auch die Integration einer grafischen Benutzerschnittstelle in iRC ist möglich um mit der App zu interagieren oder um diese zu konfigurieren.



Apps und Informationen zur Programmierung eigener Apps finden Sie hier:

[https://wiki.cpr-robots.com/index.php/Apps\\_for\\_the\\_Robot\\_Control](https://wiki.cpr-robots.com/index.php/Apps_for_the_Robot_Control)



Um Apps hinzuzufügen, zu aktualisieren, zu aktivieren und deaktivieren oder zu entfernen öffnen Sie den App-Konfigurationsbereich unter "Datei" → "Konfiguration" → "Apps". Klicken Sie auf "App installieren..." und wählen Sie ein Zip-Archiv das eine App enthält. Die Installation kann einige Minuten dauern, danach sollte sie in der Liste erscheinen. Die Schaltfläche "App aktualisieren..." aktualisiert wichtige Dateien, behält aber Einstellungen der App. In der Liste finden Sie Informationen über den Zustand der App.

Falls die App eine grafische Benutzerschnittstelle definiert finden Sie diese in der Hauptansicht der iRC in den Reitern oben. Optional können Sie auch eine App neben der 3D-Ansicht anzeigen lassen. Um eine App-Funktion in einem Programm aufzurufen fügen Sie diese im Programmeditor unter "Sonderbefehle" → "Appfunktion" hinzu. Wählen Sie die App, die gewünschte Funktion und geben Sie ggf. die angezeigten Parameter ein. Abhängig von der App können dies beispielsweise Zahlenwerte, Koordinaten, Variablennamen, Zeichenketten oder beliebige andere Daten sein.



Da Apps beliebigen Code auf der Robotersteuerung ausführen können sollten Sie Apps sicherheitskritisch betrachten und nur Apps aus vertrauenswürdigen Quellen verwenden. Fehler können zu Schäden führen, böswillige Apps können bspw. Daten an Server Dritter senden oder das Netzwerk angreifen. Es ist empfehlenswert Roboter mit Apps nicht mit dem Firmennetzwerk oder dem Internet zu verbinden.

### 13.13.7 Kameraschnittstelle

Die Kameraschnittstelle ermöglicht die Verwendung von Objekterkennungs- und Videokameras. Objekterkennungskameras erkennen die Position und Klasse von Objekten und übertragen diese Daten an die Robotersteuerung, Bilddaten sind optional. Wenn die Kamera Positionen als Pixelkoordinaten liefert berechnet die Robotersteuerung die entsprechenden Positionen im Roboterkoordinatensystem. Reine Videokameras liefern nur Bilder und können daher nur zur Beobachtung des Arbeitsbereichs, jedoch nicht für die Objekterkennung verwendet werden.

Derzeit unterstützt die Robotersteuerung folgende Kameraarten:

- ifm Objekterkennungskameras der Serien O2D200 und O2D500 sowie Kameras die deren TCP/IP-Protokoll nachbilden können
- USB-Videokameras (z.B. Webcams und industrielle Kameras, die USB video class (UVC) unterstützen)

Die Kameras können über den Bereich "Kameras" der Schnittstellenkonfiguration zur Robotersteuerung hinzugefügt werden. Dort wird zwischen Kameras am PC und der integrierten Steuerung unterschieden. Wenn der Roboter von einer integrierten Steuerung gesteuert wird, dann müssen die Kameras dort konfiguriert werden.

Die folgenden Abschnitte beschreiben die Konfiguration beider Kameratypen.



Nicht jede integrierte Steuerung unterstützt USB-Videokameras. Der Abschnitt zu USB-Kameras unten erklärt die Kompatibilität.

### Kameraschnittstelle für Objekterkennung

Zur Objekterkennung werden derzeit Kameras von ifm der Serien O2D200 und O2D500 unterstützt sowie Kameras die deren TCP/IP Protokoll nachbilden können (wie hier beschrieben: [https://wiki.cpr-robots.com/index.php/Remote\\_Variable\\_Access#Protocol](https://wiki.cpr-robots.com/index.php/Remote_Variable_Access#Protocol)).



Eine Schritt-für-Schritt-Anleitung und Tipps zur Parametrierung der Kamera finden Sie in unserem Wiki-Artikel:

[https://wiki.cpr-robots.com/index.php/2D\\_Camera\\_Integration](https://wiki.cpr-robots.com/index.php/2D_Camera_Integration)



Damit die Kamera mit der Robotersteuerung kommunizieren kann stellen Sie zunächst die folgenden Parameter über die Software des Kameraherstellers ein:

- ifm O2D200
  - Format: ASCII
  - Protokollversion: V2 oder V3
  - Objektdetail-Ausgabe: an
  - Start-Zeichenkette: start oder star
  - Stop-Zeichenkette: stop
  - Trennzeichen: #
  - Bildausgabe: egal
  - Bildformat: Windows Bitmap
- ifm O2D500
  - Protokollversion: V3
  - Für den Modus "Kontur-Anwesenheitserkennung":
    - \* Modellergebnis: an
    - \* ROI Ergebnisse: an
    - \* Objektergebnisse: egal
    - \* Start: start oder star
    - \* Trennzeichen: #
    - \* Ende: stop
    - \* Anzahl Objekte: 1
  - Für den benutzerdefinierten Modus:
    - \* Verwenden Sie das bereitgestellte Protokoll-Preset

Um eine Kamera hinzuzufügen wählen Sie den Typ der Kamera ("IFM O2D") und klicken Sie "Kamera hinzufügen". Der Bereich "Allgemein" enthält folgende Parameter:

Parameter	Bedeutung
Bild aktiviert	Wenn dieses Feld aktiviert ist fragt die Robotersteuerung regelmäßig das aktuelle Kamerabild an falls die Kamera dieses nicht selbstständig sendet. Unterstützt werden Bilder im Format "Windows Bitmap" (O2D200) und "JPEG" (O2D500).
Name	Name der Kamera im Roboterprogramm
Beschreibung	Optionale Beschreibung
IP-Adresse	IP-Adresse der Kamera
Port	Portnummer der Kamera

Die Einträge im Bereich "Koordinatentransformation" bestimmen die Verarbeitung der von der Kamera gelieferten Positionsdaten. Diese werden je nach Einstellung der Kamera als Bildkoordinaten (Pixel-Position) oder Roboterkoordinaten (in mm) übertragen. Bildkoordinaten müssen von der Robotersteuerung in Roboterkoordinaten transformiert werden, dazu dienen die Werte im Bereich "Geometrie". Siehe dazu Abbildung 66.

Parameter	Bedeutung
Skalierung	Skaliert die Pixelposition
Ursprung	Position der Kamera im Roboterkoordinatensystem
Sichtrichtung	Sichtrichtung der Kamera. Eine nach unten gerichtete Kamera hat Z=-1
Aufwärts	X-Richtung der Kamera im Roboterkoordinatensystem
Z-Abstand	Abstand der Objekte von der Kamera

Der Simulationsbereich ermöglicht die Simulation der Kamera. Diese Funktion ist nicht in der integrierten Steuerung verfügbar.

Parameter	Bedeutung
X, Y, Z	Objektposition XY in Pixel (0-640 bzw. 0-480) oder XYZ in mm, abhängig von der Einstellung "Quellkoordinatentyp"
Orientierung	Drehung des Objekts in Grad
Modellklasse	Klasse des erkannten Objekts, der Wert -1 bedeutet dass kein Objekt erkannt wurde

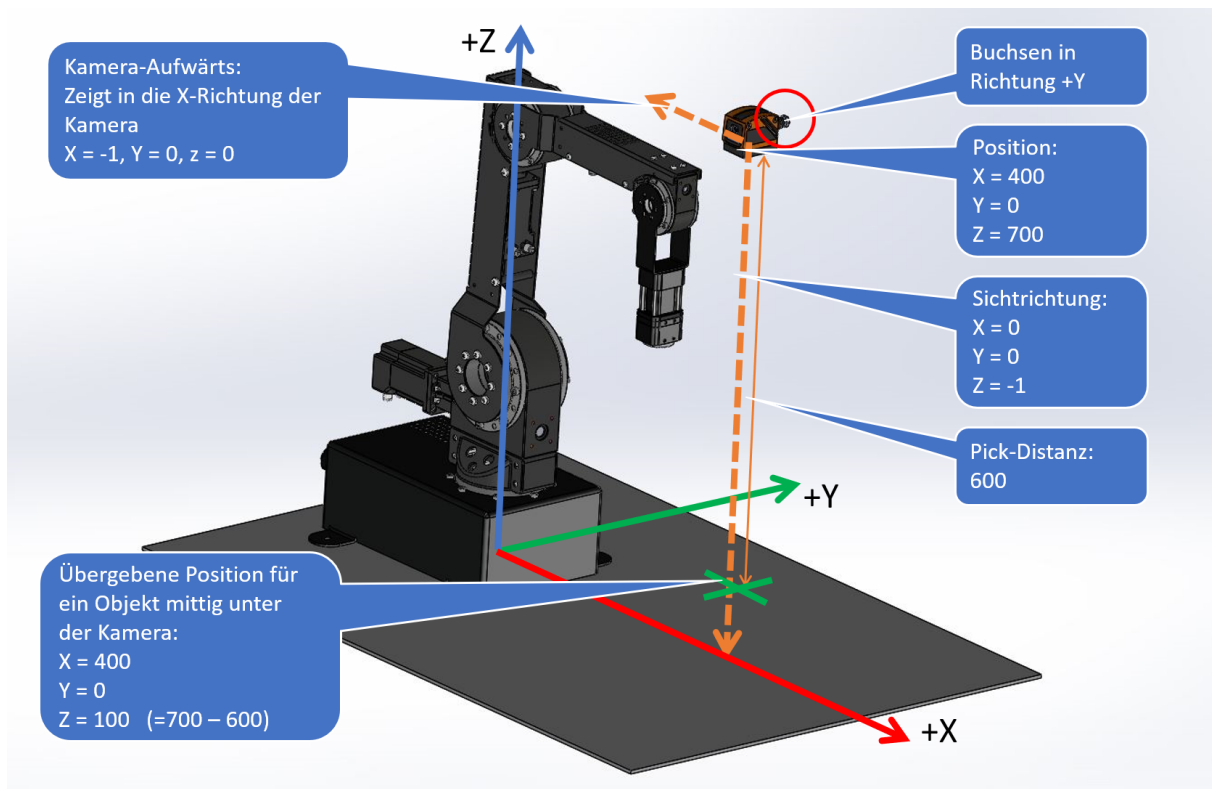
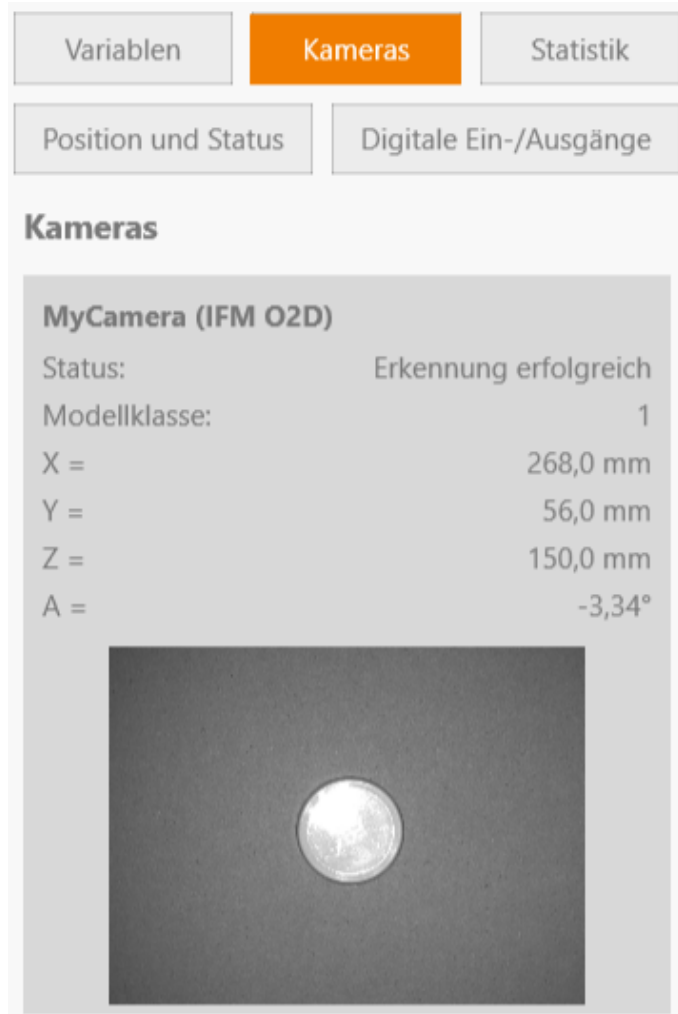


Abbildung 66: Vermessung der Kameraposition und -ausrichtung.

Nach der Konfiguration können die erkannten und berechneten Werte im Statusbereich beobachtet werden. Von der Kamera empfangene Bilder werden hier angezeigt.



Kameras	
<b>MyCamera (IFM O2D)</b>	
Status:	Erkennung erfolgreich
Modellklasse:	1
X =	268,0 mm
Y =	56,0 mm
Z =	150,0 mm
A =	-3,34°

Abbildung 67: Kamera-Statusbereich



Beachten Sie dass Kameras vom Typ O2D500 im Modus "Kontur-Anwesenheitserkennung" nach Abschluss des Einrichtungsassistenten keine Modell-IDs liefern. Bei nur einem Objekttyp sollte das kein Problem darstellen, wenn Sie jedoch Objekte unterscheiden möchten kann ein alternatives Protokoll-Preset mit Modell-IDs gewählt werden. Mehr Infos dazu finden Sie unter dem Link in dem vorherigen Hinweis.



Nach Änderungen an der Konfiguration der Kamera über die Software des Herstellers (z.B. Änderungen der Modelldefinition) klicken Sie im Kamera-Konfigurationsbereich von iRC einmal auf "Anwenden". Dadurch wird die Kamera neu verbunden. Andernfalls arbeitet die Kamera möglicherweise bis zum Neustart weiter mit der alten Konfiguration.

### Kameraschnittstelle für USB-Video

Zur Beobachtung des Arbeitsbereichs können USB-Kameras vom Typ USB video class (UVC) verwendet werden, beispielsweise Webcams oder industrielle USB-Videokameras. Die Robotersteuerung leitet die empfangenen Bilder über folgende Schnittstellen weiter:

- CRI-Schnittstelle: Das Bild wird über die Ethernet-Schnittstelle an die iRC übertragen und kann dort im Kamera-Statusbereich beobachtet werden (s. Abb. 67).
- Cloud-Schnittstelle: Wenn diese aktiviert und die Kamera zugewiesen ist wird das Bild an die RobotDimension-Cloud übertragen. Das Bild kann über die Webseite beobachtet werden (s. Abschnitt 13.13.9).



#### Kompatibilität:

Die USB-Kameraschnittstelle wird nur von integrierten Steuerungen unterstützt, die auf dem Raspberry Pi basieren. Ältere Robotersteuerungen die von einem Phytex-Modul gesteuert werden unterstützen keine USB-Kameras. Die Windows-Software iRC unterstützt diese Kameraschnittstelle ebenfalls nicht, USB-Kameras können hier direkt im Cloud-Konfigurationsbereich zugewiesen werden (s. Abschnitt 13.13.9).

Schließen Sie die Kamera an einen der USB-Ports am Steuerungsmodul an. Fügen Sie im Kamerakonfigurationsbereich von iRC eine Kamera vom Typ USB hinzu, geben Sie ihr einen eindeutigen Namen und wählen Sie die Gerätenummer. Gegebenenfalls müssen verschiedene Nummern im Bereich 0-10 ausprobiert werden. Im Kamera-Statusbereich wird der Verbindungsstatus und bei Erfolg das Kamerabild angezeigt. Um das Bild an die Cloud zu übertragen muss die Kamera anhand des angegebenen Namens im Cloud-Konfigurationsbereich.



Die Kameranummer kann sich nach einem Neustart der Steuerung oder wenn eine andere Kamera angeschlossen wurde ändern. Starten Sie die Steuerung nach der Konfiguration neu und korrigieren Sie gegebenenfalls die Nummer.

#### 13.13.8 Netzwerk

Über den Netzwerkkonfigurationsbereich kann die WLAN-Schnittstelle der integrierten Steuerung aktiviert und konfiguriert werden. Beim Öffnen des Konfigurationsbereichs wird automatisch geprüft, ob die Funktion unterstützt wird. Phytex-basierte Robotersteuerungen unterstützen generell kein WLAN, neuere Steuerungen benötigen ggf. ein Softwareupdate.

Im oberen Bereich der Netzwerkkonfiguration werden Verbindungsinformationen (u.a. das verbundene Netz und die Verbindungsqualität) oder Hinweise bei Konfigurationsproblemen angezeigt. Darunter befindet sich die eigentliche Konfiguration.

Um die Konfiguration zu ändern muss zunächst das Kästchen "Ändern" gesetzt werden. Daraufhin kann der Modus gewählt werden: "Deaktiviert" schaltet die WLAN-Schnittstelle ab, "Access Point" öffnet ein neues WLAN-Netzwerk, "Infrastruktur" lässt die Steuerung mit einem existierenden Netzwerk verbinden. Für eine Internetverbindung wird letzteres benötigt. Geben Sie dann die Parameter an (abhängig vom Modus werden nur die benötigten angezeigt):

Parameter	Bedeutung
SSID	WLAN-Netzwerkname

Parameter	Bedeutung
Passwort	WLAN-Passwort
IP-Adresse	WLAN-IP-Adresse der Steuerung. Über diese können Sie sich mit dem Roboter verbinden. Ist im Infrastruktur-Modus optional, in dem Fall wird sie automatisch zugewiesen.
Router-IP-Adresse	Die IP-Adresse des Routers im Infrastruktur-Modus, welcher die Route zum Internet bereitstellt. Dieser Eintrag ist optional und wird im Normalfall automatisch zugewiesen.

Die Einstellungen werden in der Regel direkt übernommen. Beim Aktivieren und Deaktivieren der Schnittstelle ist ein Neustart der Steuerung notwendig. Wenn die Schnittstelle aktiviert wird ist sie nach dem Neustart möglicherweise noch nicht freigeschaltet (ähnlich dem Flugzeugmodus mobiler Geräte), in diesem Fall zeigt der Konfigurationsbereich eine Schaltfläche, über die sie freigeschaltet werden kann. Falls beim Wechsel von Infrastruktur- zu Access-Point-Modus das neue WLAN-Netz nicht automatisch aufgebaut wird starten Sie die Steuerung neu.



Verbinden Sie den Roboter nur in sicheren Netzwerken. Es wird empfohlen, den Roboter nicht mit dem Firmennetzwerk oder dem Internet zu verbinden. Schützen Sie Ihr Netzwerk vor fremden Zugriffen (Firewall, sicheres Passwort, Zugang zu Netzwerkports, etc.) und überwachen Sie die Verbindungsprotokolle auf verdächtige Aktivitäten.

### 13.13.9 Cloud

Die Cloud-Schnittstelle ermöglicht die Überwachung des Roboters über RobotDimension. Nach Aktivieren und Anmelden sendet der Roboter grundlegende Zustandsinformationen und Kamerabilder an den Online-Dienst. Auf der Webseite kann der Nutzer seine Roboter auflisten und die Informationen abrufen. Die Steuerung des Roboters über das Internet ist aus Sicherheitsgründen nicht möglich.

<https://www.robotdimension.com>



Um die Cloud-Schnittstelle zu verwenden wird eine Steuerung mit Internetverbindung benötigt, beispielsweise eine integrierte Steuerung wie in Abschnitt 13.13.8 beschrieben oder ein PC mit iRC, welcher mit einem Roboter verbunden ist. Zusätzlich ist ein Konto bei RobotDimension nötig, dort muss neben dem Online-Passwort ein separates Roboterpasswort gesetzt werden.

Die Cloud-Schnittstelle kann über die Schnittstellenkonfiguration in iRC aktiviert werden. Wählen Sie oben ob die Cloud-Schnittstelle am PC oder der integrierten Steuerung konfiguriert werden soll. Um die Benutzerinformationen einzutragen muss das Feld "Anmeldedaten ändern" gesetzt werden.

Parameter	Bedeutung
Benutzername	Anmelde-Email-Adresse bei RobotDimension
Passwort	Roboterpasswort bei RobotDimension

Parameter	Bedeutung
Client-ID	Zur Unterscheidung mehrerer Roboter, wenn leer wird sie zufällig generiert

Die folgenden Daten können optional zur Cloud gesendet werden, sie dienen lediglich der Information.

Parameter	Bedeutung
Robotername	Identifizierender Name des Roboters
Roboterbesitzer	Besitzer oder Verantwortlicher des Roboters

Die Bilder von bis zu zwei Kameras können zur Cloud gesendet werden, beispielsweise um den Arbeitsbereich und die Umgebung zu überwachen. Bei einem Roboter der über iRC an die Cloud angebunden ist können am PC angeschlossene USB-Kameras über die Gerätenummer gewählt werden. Bei Erfolg wird ein Vorschaubild angezeigt. Im Fall eines Roboters mit integrierter Steuerung können USB- und Objekterkennungskameras im Kamerakonfigurationsbereich konfiguriert und hier anhand des dort angegebenen Namens zur Bildübertragung zugewiesen werden.

### 13.13.10 Gamepad

Im Abschnitt Gamepad kann die Tastenbelegung eines am PC angeschlossenen Gamepads konfiguriert werden.

Im Unterabschnitt "Allgemein" befindet sich das Feld "Benutzerdefiniertes Layout verwenden". Ist dieses gesetzt wird die benutzerdefinierte Konfiguration anstelle der Standardbelegung verwendet. Ist das Feld "Debugmodus aktivieren" gesetzt werden Tastendrucke ins Log geschrieben. Dies ist nützlich wenn die Tasten Ihres Gamepads nicht mit den Bezeichnungen in der iRC() übereinstimmen. Der Totbereich wird verwendet um driftende Thumbsticks (Joysticks) auszugleichen. Wenn sich die Achsen Ihres Roboters nach Verbinden des Gamepads selbstständig bewegen erhöhen Sie diesen Wert.

Darunter finden Sie die Thumbsticks und Tasten und deren zugewiesenen Funktionen. Ist einer Taste mehr als eine Funktion zugewiesen werden alle Funktionen gleichzeitig ausgelöst.



Da nur 3 Thumbstick-Achsen verwendet werden können bietet die Funktion "Jog-Bewegung" das Feld "Wenn Alternative gewählt ist". Weisen Sie einem Thumbstick zwei Bewegungsfunktionen zu, eine davon mit gesetzten Feld. Einer anderen Taste weisen Sie die Funktion "Achsalternative wechseln" zu. Abhängig davon ob diese Taste gedrückt ist wird die eine oder die andere Achse bewegt.



Eine Ausführliche Beschreibung aller Funktionen finden Sie im folgenden Wikiartikel:

[https://wiki.cpr-robots.com/index.php/Joystick\\_and\\_Gamepad](https://wiki.cpr-robots.com/index.php/Joystick_and_Gamepad)



### 13.13.11 Unterspannungsversorgung (USV/UPS)

Unter bestimmten Voraussetzungen kann es wünschenswert sein, eine Unterspannungsversorgung anzuschließen. Auf dem integrierten Steuerrechner ist der "apcupsd" daemon integriert, der die Steuerung bei Stromausfall und u.U. darauf folgendem, niedrigen Batteriestand sicher herunterfahren lässt. Unterstützt wird eine "APC Back-UPS Pro 1500". Die USV wird per USB Kabel an den Steuerrechner angeschlossen. Die Robotersteuerung wird per Schukostecker an die USV angeschlossen. Die USV wird an das Stromnetz angeschlossen. Eine weitere Konfiguration ist nicht erforderlich, da der "apcupsd" vorkonfiguriert ist. Sollte es erforderlich sein die Konfiguration zu ändern, wenn z.B. eine andere mit apcupsd kompatible USV angeschlossen wird, verweisen wir auf die apcupsd Dokumentation:

<http://www.apcupsd.org>



### 13.14 Zugriff auf Konfigurationsdateien

Die wichtigsten Einstellungen können und sollten wie oben beschrieben über die grafische Oberfläche vorgenommen werden. Für tiefer gehende Änderungen können die Konfigurationsdateien manuell geändert werden.

Nach manuellen Änderungen muss die Robotersteuerung immer neu gestartet werden. Dies kann über die iRC im Bereich "Datei" → "Herunterfahren und Neustarten" ausgelöst werden oder indem die Elektronik kurz abgeschaltet wird.

#### 13.14.1 Zugriff über iRC

Die Projekt-, Roboter- und Werkzeugkonfigurationsdateien können über iRC heruntergeladen, in einem einfachen Texteditor geändert und wieder hochgeladen werden. Öffnen Sie hierzu den Bereich "Datei" → "Konfigurationsdateien". Finden Sie den Abschnitt zu der Konfigurationsdatei die Sie herunterladen möchten, klicken Sie dort auf "Vom Roboter laden..." und speichern Sie die Datei an einem Ort an dem Sie sie wiederfinden.

Um die Datei zu ändern empfehlen wir einen einfachen Texteditor wie Notepad oder Notepad++.

Um die Datei wieder auf den Roboter zu kopieren klicken Sie in iRC auf die entsprechende Schaltfläche "Auf Roboter schreiben...".



Es ist empfehlenswert nach dem Anpassen der Konfigurationsdateien ein Backup der Dateien zu sichern, um im Falle von Problemen oder externer Manipulation die ursprünglichen Einstellungen wiederherstellen zu können.

#### 13.14.2 SFTP-Zugriff

Das "Secure Shell File Transfer Protocol" (SFTP) erlaubt den Zugriff auf die Dateien der integrierten Robotersteuerung beispielsweise über einen grafischen Dateieexplorer (SFTP-Client) wie z.B. FileZilla. Das Ändern von Systemdateien hierüber ist aufgrund von Zugriffsrechten nicht möglich, verwenden Sie hierzu eine SSH-Kommandozeile (siehe Abschnitt 13.14.3).

Eine bebilderte Anleitung ist hier verfügbar:

[https://wiki.cpr-robots.com/index.php/FTP\\_and\\_putty\\_Access](https://wiki.cpr-robots.com/index.php/FTP_and_putty_Access)



1. FileZilla Client herunterladen und installieren:

<https://filezilla-project.org>



2. Herstellen der Ethernet-Verbindung. LAN Kabel mit Windows PC und integriertem Computer verbinden. Der Windows-Netzwerkadapter muss die IP Adresse 192.168.3.1, Subnetz 255.255.255.0 haben. Der integrierte Computer hat standardmäßig die IP 192.168.3.11 im Subnetz 255.255.255.0.

3. FileZilla starten und folgendes angeben:

- Host: 192.168.3.11
- Username: robot
- Password: robot
- Port: 22

Dann auf Quickconnect klicken.

4. Die SFTP-Verbindung zum Roboter wird nun hergestellt:

- Im linken Fenster wird die lokale Verzeichnisstruktur des Windows-PC dargestellt.
- im rechten Fenster von FileZilla wird die Verzeichnisstruktur auf dem Linux Computer dargestellt.
- Die Steuerungssoftware auf der integrierten Steuerung heißt RobotControl. Sie liegt im Ordner ~/RobotControl
- Die Verzeichnisstruktur innerhalb des RobotControl Ordners ähnelt der von iRC.

### **Anpassen von Parametern in einer Roboter- oder Projektdatei**

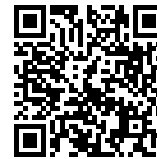
1. Dazu navigiert man in den entsprechenden Ordner und kopiert die Datei per "Drag and Drop" in einen lokalen Ordner.
2. Hier kann die Datei mit einem Standard-Texteditor wie z.B. Windows Notepad oder Notepad++ editiert werden.
3. Sobald alle Änderungen in der lokalen Datei gespeichert sind, kann diese durch "Drag and Drop" in den Zielordner auf dem Linux PC kopiert werden.
4. Dabei wird die Datei im Zielordner überschrieben "Overwrite"
5. Um die Konfigurationsänderung wirksam werden zu lassen, muss die Steuerung neu gestartet werden.

### 13.14.3 Secure Shell Zugriff

Auf den Steuerrechner kann z.B. zu Wartungszwecken per Secure Shell (SSH) zugegriffen werden, um z.B. Projektdateien oder Roboterdateien manuell zu ändern. Benutzername ist "robot", Passwort ist "robot".

Eine bebilderte Anleitung ist hier verfügbar:

[https://wiki.cpr-robots.com/index.php/FTP\\_and\\_putty\\_Access](https://wiki.cpr-robots.com/index.php/FTP_and_putty_Access)



Über einen SSH-Client wie z.B. PuTTY kann man sich mit der integrierten Steuerung verbinden und auf einer Kommandozeile arbeiten. Auf diese Weise kann man beispielsweise die Live-Ausgaben der Robotersteuerung anschauen oder Dateien ändern.



Für die Arbeit auf der Kommandozeile sind Linux-Kenntnisse notwendig! Fehler können dazu führen dass das System auf den Werkzustand zurückgesetzt werden muss.

1. Putty.exe herunterladen und starten.

<https://putty.org>



2. Herstellen der Ethernetverbindung. LAN Kabel mit Windows PC und integriertem Computer verbinden. Der Windows-Netzwerkadapter muss die IP-Adresse 192.168.3.1, Subnetz 255.255.255.0 haben. Der integrierte Computer hat die IP 192.168.3.11 im Subnetz 255.255.255.0.
3. Putty starten und im Feld "Host Name (or IP address)" die Adresse des integrierten Computers angeben: 192.168.3.11. "Port" auf 22 und "Connection Type" auf SSH stellen. Dann auf "Open" klicken. Ein Fenster öffnet sich. Möglicherweise kommt eine Sicherheitsabfrage, ob diesem Computer vertraut werden kann.
4. Login: robot
5. Password: robot
6. Nach dem Login befindet man sich im home Verzeichnis des "robot" Benutzers, in dem sich das RobotControl-Verzeichnis befindet, dessen Inhalt dem iRC-Verzeichnis in Windows ähnelt.
7. Roboterdateien befinden sich in  
~/RobotControl/Data/Robots/<Kategorie>/<Robotertyp>/<Robotertyp>.xml
8. Die Projektdatei, in der die Roboterdatei referenziert wird, befinden sich in  
~/RobotControl/Data/Projects/<Kategorie>/EmbeddedCtrl.prj

9. Als Editor sind nano, vim und vi vorinstalliert.
10. Nach dem Editieren und Speichern der Datei muss die Steuerung neu gestartet werden, damit Änderungen wirksam werden.
11. Um den RobotControl Log-Output live im Terminal anzuzeigen (das ist speziell nach Änderungen in den Dateien sinnvoll), muss RobotControl zunächst beendet werden: killall RobotControl
12. Danach navigiert man in das Verzeichnis ~/RobotControl und startet ./RobotControl
13. Der Prozess lässt sich per Strg+C beenden.
14. Nach einem Neustart der Steuerung startet RobotControl automatisch

### 13.15 Weitergehende Konfiguration

Darüber hinausgehende Dinge können in den Projekt- und Roboterdateien geändert werden. Der Zugriff auf die Konfigurationsdateien eines Roboters mit integrierter Steuerung ist über "Datei" → "Konfigurationsdateien" möglich.



Ändern Sie die Konfigurationsdateien nur wenn Sie wissen was Sie tun. Testen Sie den Roboter mit langsamer Geschwindigkeit, da er sich bei fehlerhafter Konfiguration unerwartet verhalten, sich zu schnell bewegen oder kollidieren kann.

## 14 Modbus

Das Modbus-TCP-Protokoll ermöglicht die Steuerung und den Abruf von Konfigurations- und Statusinformationen von Robotersteuerungen. Hierdurch lassen sich Roboter leicht durch speicherprogrammierbare Steuerungen (SPS) steuern und in Prozesse mit weiteren Geräten einbinden.

### 14.1 Konfiguration des Modbus-Servers

Der Modbus-Server kann über den Konfigurationsbereich in iRC konfiguriert werden. Verbinden Sie dazu iRC mit dem Roboter und öffnen Sie die Modbus-Konfiguration (Datei → Schnittstellenkonfiguration → Modbus). Der Modbus-Server wird aktiv, wenn das Kästchen "Aktiv" gesetzt ist und "Anwenden" oder "Projekt speichern" geklickt wird. Bei Bedarf kann der Port (Standard: 502) und die maximale Anzahl an Verbindungen geändert werden.

### 14.2 TIA-Portal Bibliothek

Für die Umsetzung der SPS Seite mit einer S7-1200 oder S7-1500 CPU steht dem Benutzer eine Bibliothek zur Verfügung. Die Bibliothek beinhaltet Datentypen, sowie Kommunikationsbausteine. Der Download der Bibliothek kann über nachfolgendem Link erfolgen.

[https://wiki.cpr-robots.com/index.php/Modbus\\_Server](https://wiki.cpr-robots.com/index.php/Modbus_Server)



Für das Einbinden der Bibliothek wenden Sie sich bitte an den Siemens Support.

<https://support.industry.siemens.com/cs/ww/de/view/37364723>



#### 14.2.1 Anlegen des Roboter Datenbausteins

Zunächst wird mit dem Einfügen eines Roboterdatenbausteins vom Typ "CPR\_ROBOT\_MODBUS" begonnen. Dieser Datenbaustein enthält alle wichtigen Informationen und Hilfen für die spätere Kommunikation mit dem Roboter. Nachfolgende Abbildung zeigt den Roboter Datentypen. An erster Stelle beinhaltet dieser ein "TCON\_IP\_v4" Objekt. Über dieses Objekt lässt sich die IP Adresse des Roboters, sowie der zuverwendende Port einstellen. Die Verbindungs ID kann ebenfalls eingestellt werden.

CPR_ROBOT_MODBUS								
	Name	Datentyp	Defaultwert	Err...	Sch...	Sic...	Einstellw...	Kommentar
1	▶ Robot_IP	TCON_IP_v4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	set robot ip and connection id
2	▶ Coils_OUT	Array[0..93] of Bool		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	do not change
3	▶ Coils_IN	Array[0..133] of Bool		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	do not change
4	▶ Holding_Information	Array[0..95] of Word		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	do not change
5	▶ Data	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Sollten Sie die IP Adresse ihres Roboters nicht verändert haben, so sind in den Defaultwerten bereits die entsprechenden Adressen eingetragen. Im Struct Data sind alle Einträge aus dem Modbusmapping erreichbar.



### Nomenklatur

Alle Daten zum Senden an den Roboter sind mit CMD oder OUT gekennzeichnet. Alle Daten, vom Roboter an die Steuerung, sind mit Info oder IN gekennzeichnet.

#### 14.2.2 Einfügen des Roboterkommunikations FB

Für die Kommunikation mit dem Roboter ist der FB CPR\_Robot verantwortlich. Dieser Baustein benötigt folgende Eingangssignale.

Signal	Datentyp	Beschreibung
Request_MB	Bool	Daten vom Roboter abrufen. Solange dieser Eingang gesetzt ist pflegt der FB eine aktive Kommunikation mit dem Roboter
Disconnect_MB	Bool	Trennt die TCP/IP Verbindung mit dem Roboter, kann für das Zurücksetzen von Fehlern verwendet werden
Reset	Bool	Setzt den Roboter zurück
Enable	Bool	Aktiviert den Roboter
Reference	Bool	Referenziert alle Roboter Achsen
StartProgram	Bool	Startet das Roboterprogramm
StopProgram	Bool	Stoppt das Roboterprogramm
Robot_Data	CPR_ROBOT_MODBUS	In/Out für Roboterdatenbaustein

Der Robot\_Data Input gibt dem Baustein alle Notwendigen Daten vor um mit dem Roboter zu kommunizieren. Durch den Einsatz mehrerer Datenbausteine und CPR\_Robot FB's ist es somit möglich mit mehreren Roboter gleichzeitig zu kommunizieren. Als Ausgänge stehen folgende Signale zur Verfügung.

Signal	Datentyp	Beschreibung
Enabled	Bool	Roboter ist aktiviert
Referenced	Bool	Roboter ist referenziert
ProgrammRunning	Bool	Das Roboterprogramm wird ausgeführt

#### 14.2.3 Datenzugriff

Für den Zugriff auf die Roboterdaten können die Daten im Roboter DB manipuliert werden. Diese werden im Anschluss automatisch an den Roboter übertragen und verarbeitet.

### 14.3 Adresszuordnung

Dieser Abschnitt beschreibt die Adresszuordnung um eigene Implementationen und Erweiterungen der SPS-Funktionsblöcke zu ermöglichen.

Modbus definiert vier Speicherbereiche, die durch verschiedene Nachrichten gelesen und geschrieben werden können. In der Adresszuordnung der igus-Roboter werden die Bereiche wie folgt verwendet. In einigen Fällen sind Informationen sowohl bitweise als auch als Register abrufbar.

Speicherbereich	Zugriff	Verwendung
Coils	1 Bit, Lesen und Schreiben	Aktionen und änderbare Zustände
Diskrete Eingänge	1 Bit, nur Lesen	Zustände, Informationen
Halteregeister	16 Bit, Lesen und Schreiben	Änderbare Werte oder Zustände
Eingaberegister	16 Bit, nur Lesen	Nicht änderbare Werte, Informationen

Tabelle 26: Verwendung der Modbus-Speicherbereiche

Folgende Modbus-Funktionscodes können zum Lesen und Schreiben der Speicherbereiche verwendet werden.

Speicherbereich	Lesen	Schreiben
Coils	1	5 (einzel), 15 (mehrere)
Diskrete Eingänge	2	-
Halteregeister	3	6 (einzel), 16 (mehrere), 22 (maskiert), 23 (lesen und schreiben)
Eingaberegister	4	-

Tabelle 27: Unterstützte Modbus-Funktionscodes (dezimal)

Da Modbus lediglich 1-Bit und 16-Bit-Speicherbereiche definiert werden komplexe Datentypen und Aktionen hier wie folgt definiert:

Datentyp	Beschreibung
boolean	Bit "0" oder "1" schreiben startet die entsprechende Aktion bzw. setzt den Zustand.
enum	Aufzählung. Bedeutung hängt vom Register ab, siehe Abschnitt 14.3.5.
Info	Information, nicht schreibbar
int32 / uint32	Zwei 16-Bit Register, niederwertiges Register zuerst.
steigende Flanke $\lrcorner$	Aktion wird ausgeführt, wenn zunächst eine 0 und darauf eine 1 geschrieben wird. Achtung: Einige Coils geben beim Lesen den tatsächlichen Wert zurück, nicht den zuletzt geschriebenen. Bei doppelter Belegung (z.B. Motoren aktivieren/deaktivieren) wird die Aktion abhängig vom tatsächlichen Zustand gewählt.

Datentyp	Beschreibung
string	Zeichenkette. Zwei 8-Bit-Zeichen je Register, niederwertiges Byte zuerst. Die Zeichenkette endet mit einem Null-Byte oder wenn die maximale Registeranzahl erreicht ist.

Tabelle 28: Definition komplexer Datentypen



Positionswerte (X, Y, Z, A, B, C, Achswinkel) werden, wo erforderlich, als 32-Bit-Werte angegeben und belegen daher 2 Register. Das erste Register enthält die niederwertigen, das zweite die höherwertigen Bits. Bei negativen Werten müssen beide Register entsprechend gesetzt werden.

Unter folgendem Link finden Sie ein Rechenbeispiel um den Roboter an eine Positionsvorgabe fahren zu lassen.

[https://wiki.cpr-robots.com/index.php/Moving\\_Robots\\_via\\_Modbus](https://wiki.cpr-robots.com/index.php/Moving_Robots_via_Modbus)



Die folgenden Tabellen beschreiben die Modbus-Adresszuordnung Version 3 (siehe Eingaberegister 3).

### 14.3.1 1 Bit Lesezugriff (Coils und diskrete Eingänge)

Über Coils und diskrete Eingänge sind 1-Bit-Zugriffe möglich. Dies wird hier verwendet, um Ein- und Ausgänge sowie einfache Zustände abzufragen und um Aktionen auszulösen.



Damit zum Lesen weniger Modbus-Nachrichten gesendet werden müssen ist der Lesezugriff auf 1-Bit-Daten sowohl als Coils als auch als diskrete Eingänge möglich. Dabei sollte vermieden werden Werte als diskreten Eingang zu lesen, die zuvor als Coil geschrieben wurden. Es bietet sich daher an nur den Coil-Zugriff zu verwenden.

Adressen	Datentyp	Beschreibung
10	Info	Hat Roboterachsen
11	Info	Hat externe Achsen
12	Info	Hat Greiferachsen
13	Info	Hat Plattformachsen
14	Info	Hat digitale Ein-/Ausgabemodule
20	Info	Module - Kein Fehler
21	Info	Modulfehler - Temperatur
22	Info	Modulfehler - Notaus / Unterspannung
23	Info	Modulfehler - Motor nicht aktiviert

Adressen	Datentyp	Beschreibung
24	Info	Modulfehler - Kommunikation
25	Info	Modulfehler - Schleppfehler
26	Info	Modulfehler - Encoderfehler
27	Info	Modulfehler - Überstrom
28	Info	Modulfehler - Treiberfehler
29	Info	Modulfehler - Bus tot
30	Info	Modulfehler - Modul tot
31-36	Info	Modulfehler - reserviert für künftige Fehler
37	Info	Kinematik - Kein Fehler
38	Info	Kinematik - Achslimit Min
39	Info	Kinematik - Achslimit Max
40	Info	Kinematik - Zentralachsensingularität
41	Info	Kinematik - Außer Reichweite
42	Info	Kinematik - Handgelenkssingularität
43	Info	Kinematik - Virtuelle Box erreicht
44	Info	Kinematik - Bewegung nicht erlaubt
45-49	Info	Kinematik - reserviert für künftige Fehler
50	<input type="checkbox"/>	Ist CAN-Bus verbunden? / Verbinden (1) / Verbindung trennen (0) (Verbinden / Trennen ist derzeit nicht möglich)
51	<input type="checkbox"/>	Steuerungsrechner Herunterfahren
52	<input type="checkbox"/>	Roboter Zurücksetzen
53	<input type="checkbox"/>	Sind die Motoren aktiv? / Motoren aktivieren (1) / deaktivieren (0)
54	Info	Normaler Betrieb (siehe Betriebsmodus, Tabelle 35)
60	<input type="checkbox"/>	Sind alle Achsen referenziert? / referenzieren
61-66	<input type="checkbox"/>	Ist Roboterachse referenziert? / referenzieren
67-69	<input type="checkbox"/>	Ist externe Achse referenziert? / referenzieren
70-72	<input type="checkbox"/>	Ist Greiferachse referenziert? / referenzieren
73	<input type="checkbox"/>	Alle Achsen auf 0 setzen
74	<input type="checkbox"/>	Starte das Referenzierprogramm, danach "Referenziere Alle". Muss referenziert und die Motoren aktiviert sein.
100	<input type="checkbox"/>	MoveTo starten - kartesisch
101	<input type="checkbox"/>	MoveTo starten - kartesisch relativ Basiskoordinaten
102	<input type="checkbox"/>	MoveTo starten - kartesisch relativ Werkzeugkoordinaten
103	<input type="checkbox"/>	MoveTo starten - Achsbewegung
104	<input type="checkbox"/>	MoveTo starten - Achsbewegung relativ
110	Info	Ist Zero-Torque (Handführungsmodus) verfügbar?
111	boolean	Ist Zero-Torque aktiviert? / aktivieren (1) / deaktivieren (0)

Adressen	Datentyp	Beschreibung
112	Info	Bewegt sich der Roboter?
120	Info	Ist ein Roboterprogramm geladen?
121	Info	Ist ein Logikprogramm geladen?
122	┘	Läuft das Roboterprogramm? / starten / fortführen
123	┘	Ist Roboterprogramm pausiert? / pausieren
124	┘	Ist das Roboterprogramm gestoppt? / stoppen
130	┘	Nächsten Verzeichniseintrag wählen
131	┘	Vorherigen Verzeichniseintrag wählen
132	Info	Ist der gewählte Verzeichniseintrag eine Programmdatei
133	┘	Gewählten Verzeichniseintrag als Roboterprogramm laden / Verzeichnis öffnen
134	┘	Gehe ins Basisverzeichnis (.../Data/Programs)
135	┘	Roboterprogramm entladen
136	┘	Logikprogramm entladen
150	boolean	Byteorder von Zeichenketten-Registern umkehren (Standard: false, erster Buchstabe im niedrigen Byte). Ab Mapping Version 3.
200-299	boolean	Globale Signale
300-363	boolean	Digitale Ausgänge
364-427	Info	Digitale Eingänge

Tabelle 29: Zuweisung der Coils und der diskreten Eingänge

### 14.3.2 16 Bit Lesezugriff (Eingaberegister)

Die Eingaberegister bieten Lesezugriff auf Konfigurations-, Zustands- und Statistikinformationen. Um Zahlenwerte in die korrekte Einheit umzurechnen muss mit dem unter Einheit angegebenen Faktor multipliziert werden. Die Bedeutung der Zustandsregister mit Datentyp enum ist im Abschnitt 14.3.5 beschrieben.

Adressen	Datentyp	Einheit	Beschreibung
0	uint16		Software-ID (902=iRC (Simulation V12+V13), 980=Robot-Control/TinyCtrl)
1	uint16		Software Major-Version (z.B. 12)
2	uint16		Software Minor-Version (z.B. 6)
3	uint16		Modbus Mapping Version
4-5	uint32	Minuten	Uptime komplett
6-7	uint32	Minuten	Uptime zuletzt
8-9	uint32	Minuten	Uptime aktiviert
10-11	uint32	Minuten	Uptime Bewegung

Adressen	Datentyp	Einheit	Beschreibung
12	uint16		Programmstarts
13	uint16	0.1ms	Zykluszeit-Soll
14	uint16	0.1ms	Zykluszeit-Max (letzte 50 Zyklen)
15	uint16	0.01Hz	Zyklusfrequenz (Durchschnitt)
16	uint16	0.01%	Arbeitslast
20	uint16		Anzahl Roboterachsen
21	uint16		Anzahl externe Achsen
22	uint16		Anzahl Greiferachsen
23	uint16		Anzahl Plattformachsen
24	uint16		Anzahl Ein-/Ausgabemodule
25-30	Bitfeld		Modulfehlercodes Roboterachsen
31-33	Bitfeld		Modulfehlercodes externe Achsen
34-36	Bitfeld		Modulfehlercodes Greiferachsen
37-40	Bitfeld		Modulfehlercodes Plattformachsen
41-43	Bitfeld		Modulfehlercodes Ein-/Ausgabemodule
44-49	int16	0.1°C	Temperatur Elektronik Roboterachsen
50-52	int16	0.1°C	Temperatur Elektronik externe Achsen
53-55	int16	0.1°C	Temperatur Elektronik Greiferachsen
56-59	int16	0.1°C	Temperatur Elektronik Plattformachsen
60-65	int16	0.1°C	Temperatur Motoren Roboterachsen
66-68	int16	0.1°C	Temperatur Motoren externe Achsen
69-71	int16	0.1°C	Temperatur Motoren Greiferachsen
72-75	int16	0.1°C	Temperatur Motoren Plattformachsen
76-81	uint16	mA	Ströme Roboterachsen
82-84	uint16	mA	Ströme externe Achsen
85-87	uint16	mA	Ströme Greiferachsen
88-91	uint16	mA	Ströme Plattformachsen
92	uint16	0.01V	Spannung
93	uint16	mA	Gesamtstrom
94	uint16	0.1%	Batterieladezustand (derzeit nicht implementiert)
95	uint16	enum	Kinematik - Fehlercode
96	uint16	enum	Betriebsmodus
130-135	int32	0.01mm	Aktuelle kartesische Position
136-141	int16	0.01°	Aktuelle kartesische Orientierung
142-153	int32	0.01	Aktuelle Achsposition Roboterachsen
154-159	int32	0.01	Aktuelle Achsposition ext. Achsen

Adressen	Datentyp	Einheit	Beschreibung
160-165	int32	0.01	Aktuelle Achsposition Greiferachsen
166-173	int32	0.01	Aktuelle Achsposition Plattform
262	uint16		Anzahl geladener Roboterprogramme
263	int16		Nummer des aktuellen Programms, 0 für Hauptprogramm
264	uint16		Anzahl Anweisungen in aktuellem Programm
265	int16		Nummer der aktuellen Anweisung, -1 wenn Programm nicht läuft
266	enum		Grund für letzten Programmstopp oder -pause
331	uint16		Anzahl Einträge im aktuellen Verzeichnis
333-364	string		Name des gewählten Verzeichniseintrags
365-396	string		Name des aktuellen Verzeichnisses
207-210	Bitfeld		Digitale Eingänge
400-431	string		Info/Fehlernachricht kurz (wie auf Handbediengerät)
440-455	int16		Zahlenvariablen mb_num_r1 - mb_num_r16
456-711	int16	0.1	Positionsvariablen mb_pos_r1 - mb_pos_r16 (s. Abs. 14.3.4)

Tabelle 30: Zuweisung der Eingaberegister

### 14.3.3 16 Bit Lese- und Schreibzugriff (Halteregister)

Über die Halteregister können Zielpositionen und Variablen, sowie der Name eines zu ladenden Programms geschrieben werden.

Adressen	Datentyp	Einheit	Beschreibung
130-135	int32	0.01mm	Zielposition kartesisch
136-141	int32	0.01°	Zielorientierung kartesisch
142-153	int32	0.01	Zielposition Roboterachsen
154-159	int32	0.01	Zielposition externe Achsen
174-177	int32	0.01mm	Zielposition Plattform
178-179	int32	0.01°	Zielorientierung Plattform
180	int16	0.1	Geschwindigkeit für MoveTo (Prozent oder mm/s)
181-186	int32	0.1	Zielgeschwindigkeit der ext. Achsen im Velocity-Modus
187	uint16	0.01%	Geschwindigkeits-Override
188	enum		Jog-Modus
260	enum		Roboterprogramm RunState
261	enum		Roboterprogramm Replay-Modus
267-298	string		Name des geladenen Roboterprogramms / bei Schreiben laden

Adressen	Datentyp	Einheit	Beschreibung
299-330	string		Name des geladenen Logikprogramms / bei Schreiben laden
332	uint16		Nummer des gewählten Verzeichniseintrags
200-206	Bitfeld		Globale Signale
207-210	Bitfeld		Digitale Ausgänge
440-455	int16		Zahlenvariablen mb_num_w1 - mb_num_w16
456-711	int16	0.1	Positionsvariablen mb_pos_w1 - mb_pos_w16 (s. Abs. 14.3.4)

Tabelle 31: Zuweisung der Haltereister

#### 14.3.4 Zahlen- und Positionsvariablen

Zur Kommunikation mit Roboter- und Logikprogrammen können neben den globalen Signalen auch vordefinierte Programmvariablen verwendet werden. Dazu stehen jeweils 16 lesbare und 16 schreibbare Zahlen- und Positionsvariablen zur Verfügung.

Name	Typ	Zugriff über Modbus
mb_num_r1 - mb_num_r16	Zahlenvariable	nur Lesen (Eingaberegister)
mb_num_w1 - mb_num_w16	Zahlenvariable	Lesen und Schreiben (Halteregister)
mb_pos_r1 - mb_pos_r16	Positionsvariable	nur Lesen (Eingaberegister)
mb_pos_w1 - mb_pos_w16	Positionsvariable	Lesen und Schreiben (Halteregister)

Tabelle 32: Programmvariablen für die Kommunikation über Modbus



Verwenden Sie bei Einsatz der SPS-Funktionsblöcke die schreibbaren Variablen nur zum Senden von Werten von der SPS zum Roboter. Ändern Sie diese Variablen nicht im Roboterprogramm, da sie von der SPS regelmäßig überschrieben werden.



Anders als normale Programmvariablen stehen die Modbus-Variablen immer zur Verfügung. Es muss kein Programm gestartet sein und die Variablen müssen nicht mit der Store-Anweisung deklariert werden.

Jede Zahlenvariable wird in einem Register abgebildet. Da hier nur Ganzzahlen unterstützt werden muss das Roboterprogramm gegebenenfalls durch Multiplikation oder Division auf den gewünschten Wertebereich konvertieren (Math-Anweisung).

Positionsvariablen bestehen aus jeweils 16 Registern, deren Werte in Zehntelgenauigkeit angegeben sind:

- 9 Register für Roboter- und externe Achsen (A1-A6, E1-E3)
- 3 Register für die kartesische Position (X, Y, Z)

- 3 Register für die kartesische Orientierung (A, B, C)
- 1 Register zur Wahl der Übersetzungsart

Entsprechend der Übersetzungsart konvertiert die Kinematik von Achswinkeln zu kartesischen Koordinaten oder umgekehrt. Dies kann hilfreich sein, wenn beispielsweise Zielpositionen nur als Koordinaten vorliegen, der Roboter aber per Joint-Anweisung fahren soll.

Wert	Bedeutung
0	Keine Konvertierung, Achspositionen und kartesische Koordinaten werden ohne Änderung übernommen (Standard)
1	Die kartesischen Koordinaten und Orientierung wird aus den Achspositionen berechnet
2	Die Achspositionen werden aus den kartesischen Koordinaten und der Orientierung berechnet

Tabelle 33: Übersetzungsart



Beachten Sie, dass möglicherweise nicht alle Positionen von der Kinematik erreicht werden können. Prüfen Sie die Werte daher auf Plausibilität.

### 14.3.5 Bedeutung der Aufzählungswerte

Die folgenden Tabellen beschreiben die Bedeutung der Aufzählungswerte.

Wert	Bedeutung
0	Kein Fehler
13	Achslimit Min
14	Achslimit Max
21	Zentralachsensingularität
22	Außer Reichweite
23	Handgelenkssingularität
30	Virtuelle Box in X+ berührt
31	Virtuelle Box in X- berührt
32	Virtuelle Box in Y+ berührt
33	Virtuelle Box in Y- berührt
34	Virtuelle Box in Z+ berührt
35	Virtuelle Box in Z- berührt
50	NAN in Berechnung
90	Bewegung nicht erlaubt
65535 (0xFFFF)	Unbekannter Fehler

Tabelle 34: Kinematikfehlercode

Wert	Bedeutung
0	Standard - normaler Betrieb
1	Schwerer Fehler, Steuerung muss neu gestartet werden
2	CAN-Bridge (CRI, z.B. Abruf der Firmwareparameter)

Tabelle 35: Betriebsmodus

Wert	Bedeutung
0	Achsen
1	Kartesisch Basiskoordinatensystem
2	Kartesisch Werkzeugkoordinaten
3	Plattform
0xFFFF	Ungültig

Tabelle 36: Jog-Modus

Wert	Bedeutung
0	Programm läuft nicht
1	Programm läuft
2	Programm pausiert

Tabelle 37: Programmzustände (RunState)

Wert	Bedeutung
0	Programm einmal ausführen
1	Programm wiederholen
2	Anweisungen schrittweise ausführen
3	Schnell (nicht verwendet)

Tabelle 38: Replay-Modus

Wert	Bedeutung
0	Benutzer (Bediengerät, CRI, Modbus, etc.)
1	PLC
2	Programm (Stopp-/Pause-Anweisung)
3	Replay Step (Schrittbetrieb)
4	Shutdown (System fährt herunter)

---

<b>Wert</b>	<b>Bedeutung</b>
100	Fehler
101	Pfadgeneratorfehler 1
102	Pfadgeneratorfehler 2
103	Fehler in Zustandsmaschine

---

Tabelle 39: Grund für letzten Stopp/Pause des Programms

## **15 Wartung**

### **15.1 Reinigung**

- Nachdem die Steuerung vom Stromnetz getrennt ist, kann sie mit einem feuchten Tuch abgewischt werden.
- Nachdem die Steuerung vom Stromnetz getrennt ist, können Lüfter oder Lüftungsschlitze mit einem feuchten Tuch oder leichter Druckluft vorsichtig gereinigt werden. Dabei müssen die Rotoren der Lüfter festgehalten werden, damit diese durch zu grosse Geschwindigkeit (rpm) keinen Lagerschaden erhalten.

## 16 Fehlerbehebung

### 16.1 Häufig gestellte Fragen

Antworten zu häufig gestellten Fragen finden Sie in unserem Wiki:

<https://wiki.cpr-robots.com>



### 16.2 Fehlercodes und Wege zur Behebung

Unser Troubleshooting-Guide hilft schrittweise bei der Identifikation und Lösung von Problemen:

<https://wiki.cpr-robots.com/index.php/Troubleshooting>



Bei Verbindungsproblemen mit dem Roboter schauen Sie bitte in diesen Artikel:

[https://wiki.cpr-robots.com/index.php/Troubleshooting\\_Connection\\_Issues](https://wiki.cpr-robots.com/index.php/Troubleshooting_Connection_Issues)



Fehlercodes und Probleme mit Hardware und Elektronik sind in folgendem Artikel beschrieben:

[https://wiki.cpr-robots.com/index.php/Robot\\_Hardware\\_Troubleshooting](https://wiki.cpr-robots.com/index.php/Robot_Hardware_Troubleshooting)



Lösungen zu häufigen Softwareproblemen sind in folgendem Artikel beschrieben:

[https://wiki.cpr-robots.com/index.php/Common\\_Software\\_Troubleshooting](https://wiki.cpr-robots.com/index.php/Common_Software_Troubleshooting)



### 16.3 Testsoftware Module Control

Um Achsen einzeln und ohne Einfluss der darüber liegenden Robotersteuerung zu testen kann die Software Module Control verwendet werden. Sie ermöglicht unter anderem das Bewegen und Referenzieren der Achse sowie das Auslesen und Ändern von Parametern (siehe Abschnitt 12.5).

Unter folgendem Link kann Module Control heruntergeladen werden. Dort ist ebenfalls die Bedienung genauer beschrieben.

[https://wiki.cpr-robots.com/index.php/Config\\_Software\\_ModuleCtrl](https://wiki.cpr-robots.com/index.php/Config_Software_ModuleCtrl)



## 16.4 Support-Kontakt

Bei Problemen helfen wir gerne!

- igus Support-Landingpage:

<https://www.igus.de/info/igus-low-cost-automation>



- E-Mail: [ww-robot-control@igus.net](mailto:ww-robot-control@igus.net)



Bei Softwareproblemen senden Sie uns bitte die Logdateien der iRC - igus Robot Control und der integrierten Steuerung. Klicken Sie dazu einfach auf das Fragezeichen unten rechts im 3D-Bereich um alle relevanten Dateien automatisch an eine E-Mail anzuhängen

- Telefon: +49(0)2203 / 96498-255