

LOW  
COST  
AUTOMATION  
by igus®

igus® ReBeL®  
Benutzerhandbuch  
Software und Elektrik



---

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Kontakt . . . . .	4
1.2	Bestimmungsgemäße Verwendung . . . . .	4
1.3	Zielgruppe und Qualifikation . . . . .	4
1.4	Verwendete Symbole . . . . .	4
1.5	Produktsicherheit . . . . .	5
1.6	Vorschriften . . . . .	5
<b>2</b>	<b>Übersicht über das System</b>	<b>6</b>
2.1	Spezifikationen . . . . .	6
2.2	Mechanische Abmessungen . . . . .	6
<b>3</b>	<b>Sicherheitshinweise</b>	<b>7</b>
<b>4</b>	<b>Anforderungen</b>	<b>8</b>
4.1	Umgebungsbedingungen . . . . .	8
4.2	Softwareanforderungen . . . . .	8
<b>5</b>	<b>Schnellstart-Anleitung</b>	<b>9</b>
5.1	Einrichten und Anschließen . . . . .	9
5.2	Einschalten . . . . .	9
5.3	Verbinden und Bewegen des Roboters . . . . .	9
5.3.1	Vorbereitung mit dem integrierten Computer . . . . .	9
5.3.2	Erstes Bewegen des Roboters . . . . .	10
<b>6</b>	<b>Bewegen eines Roboters mit iRC</b>	<b>12</b>
6.1	Die grafische Benutzeroberfläche . . . . .	12
6.1.1	Auswählen des Robotertyps . . . . .	13
6.1.2	Navigation und Bewegen des Roboters in der 3D-Ansicht . . . . .	14
6.2	Anschließen des Roboters . . . . .	14
6.2.1	Verbindung zur Hardware . . . . .	15
6.2.2	Den Roboter bewegen . . . . .	15
6.3	Referenzieren des Roboters . . . . .	16
6.3.1	Schrittweise Anleitung der Referenzierung . . . . .	16
6.4	Bewegen des Roboters . . . . .	16
6.5	Starten von Roboterprogrammen . . . . .	18
6.6	Digitale Ein- und Ausgänge . . . . .	19
6.7	Software-Schnittstellen . . . . .	19
6.8	Aktualisieren der Software . . . . .	20

<b>7</b>	<b>Programmierung eines Roboters mit iRC</b>	<b>21</b>
7.1	Programmeditor	21
7.1.1	Ändern der Befehlssequenz	21
7.1.2	Position nachbessern	22
7.1.3	Startbefehl festlegen	22
7.2	Kommentare und Informationen im Programm	23
7.2.1	Informationen zum Programm	23
7.2.2	Beschreibungen	23
7.2.3	Kommentare	23
7.3	Bewegung	23
7.3.1	Abbruchbedingungen	23
7.3.2	Beschleunigung und Glättung	23
7.3.3	Achsbewegung	24
7.3.4	Lineare Bewegung	25
7.3.5	Relative Bewegung	26
7.3.6	Kreisbewegung	26
7.3.7	Dauerbewegung	28
7.4	Greifer und digitale Ein-/Ausgänge	28
7.4.1	Digitale Eingänge	28
7.4.2	Digitale Ausgänge	28
7.4.3	Öffnen/Schließen des Greifers	28
7.5	Programmfluss	28
7.5.1	Bedingungen	28
7.5.2	Stop	29
7.5.3	Pause	29
7.5.4	Wait	29
7.5.5	If-then-else	29
7.5.6	Schleifen	30
7.5.7	Rasterbewegungen / Palettierung	30
7.5.8	Unterprogramme	32
7.6	Variablen und Variablenzugriff	32
7.6.1	Benutzervariablen	32
7.6.2	Systemvariablen	32
7.6.3	Zugriff auf Elemente	34
7.6.4	Berechnungen mit Variablen	34
7.6.5	Variablen beobachten	34
7.7	Kamera	34
<b>8</b>	<b>Hardwarekonfiguration</b>	<b>36</b>
8.1	Ein-/Ausgänge	36
8.1.1	Elektrische Integration	36
8.1.2	Software-Konfiguration	37
8.1.3	Sensoren und Taster an der Basis anschließen	38
8.1.4	Sensoren und Taster am Arm anschließen	38
8.1.5	Aktoren am Arm anschließen	38
8.2	Motorbremse	38
8.3	Konfiguration der Motorsteuerungen	38



<b>9 Softwarekonfiguration</b>	<b>40</b>
9.1 Projektkonfiguration	41
9.1.1 Programm	41
9.1.2 Werkzeug	41
9.1.3 Ein-/Ausgänge	41
9.1.4 Virtuelle Box	41
9.2 Schnittstellen	42
9.2.1 SPS-Schnittstelle	42
9.2.2 Programmwahl über digitale Eingänge	43
9.2.3 Modbus	43
9.2.4 CRI-Schnittstelle	44
9.2.5 Kameraschnittstelle	44
9.2.6 Netzwerk	47
9.2.7 Cloud	47
9.2.8 Secure Shell Zugriff	48
9.2.9 SFTP Zugriff	50
9.2.10 Unterspannungsversorgung (USV/UPS)	51
9.3 Weitergehende Konfiguration	51
<b>10 Modbus</b>	<b>52</b>
10.1 Lizenzierung	52
10.2 Konfiguration des Modbus-Servers	52
10.3 TIA-Portal Bibliothek	52
10.3.1 Anlegen des Roboter Datenbausteins	52
10.3.2 Einfügen des Roboterkommunikations FB	53
10.3.3 Datenzugriff	53
10.4 Adresszuordnung	54
10.4.1 1 Bit Lesezugriff (Coils und diskrete Eingänge)	55
10.4.2 16 Bit Lesezugriff (Eingaberegister)	57
10.4.3 16 Bit Lese- und Schreibzugriff (Halteregister)	58
10.4.4 Zahlen- und Positionsvariablen	59
10.4.5 Bedeutung der Aufzählungswerte	60
<b>11 Wartung</b>	<b>63</b>
11.1 Reinigung	63
<b>12 Fehlerbehebung</b>	<b>64</b>
12.1 Häufig gestellte Fragen	64
12.2 Fehlercodes und Wege zur Behebung	64
12.3 Testsoftware Module Control	64
12.4 Support-Kontakt	65

---

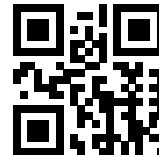
# 1 Einleitung

## 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
2. **Robotersteuerung:** bestehend aus u.a. Embedded Computer, Achsmodule und IO-Module.
3. **Robotersteuerungssoftware:** Steuerungssoftware zum Bewegen des Roboters und dem Ausführen von Roboterprogrammen.
4. **Programmierungsumgebung:** Grafische Software zum Einrichten von Roboterprogrammen.

### 2.1 Spezifikationen

<b>Modulare Steuerung</b>	<b>Eigenschaft</b>
Stromversorgung	24V / 11,7A durch externes Netzteil
Typ	Integrierte Steuerungselektronik
Kommunikation mit PC/SPS	CRI oder Modbus via Ethernet
Interne Kommunikation	CAN
Achsmodule	Closed-Loop-Motormodule Absolutencoder integrierte Bremse
Digitale IO-Module	In der Basis: 4 digitale Eingänge Im Arm: 2 Ein- und Ausgänge Halbleiterrelais, max. 500 mA

Tabelle 2: Robotersteuerung - Komponenten

<b>Steuerrechner</b>	<b>Eigenschaft</b>
Typ	Single Board Computer mit Tochtermodul
Betriebssystem	Linux-basiert
Software	TinyCtrl Robotersteuerungssoftware
Schnittstellen	CAN-Bus (Verbindung zu den Modulen) Ethernet (Verbindung zu Windows PC) USB Digitale Eingänge für Zustimmschalter

Tabelle 3: Integrierter Computer - Spezifikationen

### 2.2 Mechanische Abmessungen

Die mechanischen Abmessungen entnehmen Sie bitte der ReBeL-Betriebsanleitung.

### 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-Ausschalter, 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 Anforderungen

### 4.1 Umgebungsbedingungen

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

Tabelle 4: Umgebungsbedingungen

### 4.2 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 4.7.2 oder neuer
- mindestens 500MB freier Speicherplatz
- Grafikkarte (integriert oder dediziert)
  - OpenGL 3.0 oder neuer
  - Herstellertreiber (der Standardtreiber von Microsoft wird nicht unterstützt)
- einen freien Ethernet-Port

## 5 Schnellstart-Anleitung

### 5.1 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 4.1.
- 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. Der Roboter ist nach Abschluss dieser Schritte bereit zum Einschalten.

### 5.2 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 ein.

### 5.3 Verbinden und Bewegen des Roboters

#### 5.3.1 Vorbereitung mit dem integrierten Computer

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

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 der Robotersteuerung befindet.

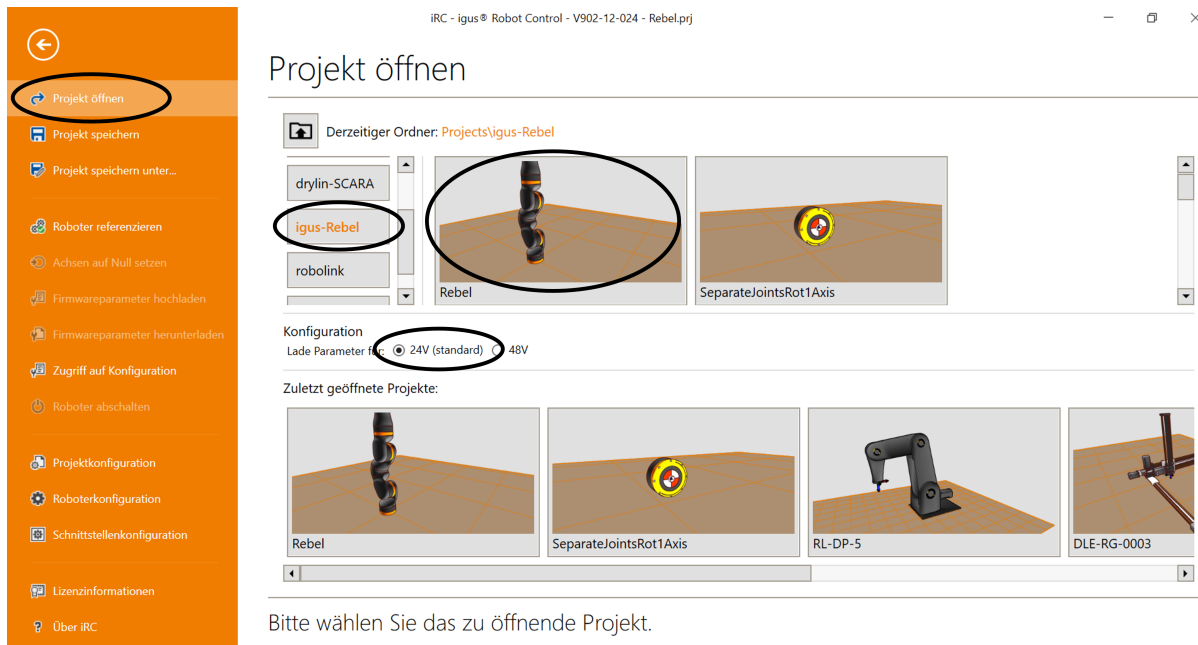


Abbildung 1: Projektauswahl

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".

### 5.3.2 Erstes Bewegen des Roboters

1. Installieren Sie die iRC Software auf Ihrem PC.
2. Starten Sie die iRC Software. Beim Start können Sie das auf Ihren Roboter zutreffende Projekt auswählen. Bitte beziehen Sie sich auf die Produktnummer bzw. den Produktnamen, die Projektnamen basieren auf diesen (siehe Abb. 1).
3. Sie können den Roboter jetzt aktivieren, indem Sie nachfolgenden Schaltflächen in der angegebenen Reihenfolge drücken (siehe auch Abb. 2):
  - 3.1 "Verbinden"
  - 3.2 "Zurücksetzen"
  - 3.3 "Aktivieren"
4. Jetzt sollte die Status-LED-Leuchte links in iRC grün sein und der Status "Kein Fehler" anzeigen.
5. Sie können nun die Gelenke des Roboters mit Hilfe der Schaltflächen auf der Registerkarte "Jogging" bewegen (siehe Abb. 2).



#### Kartesische Bewegungen und Programme

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



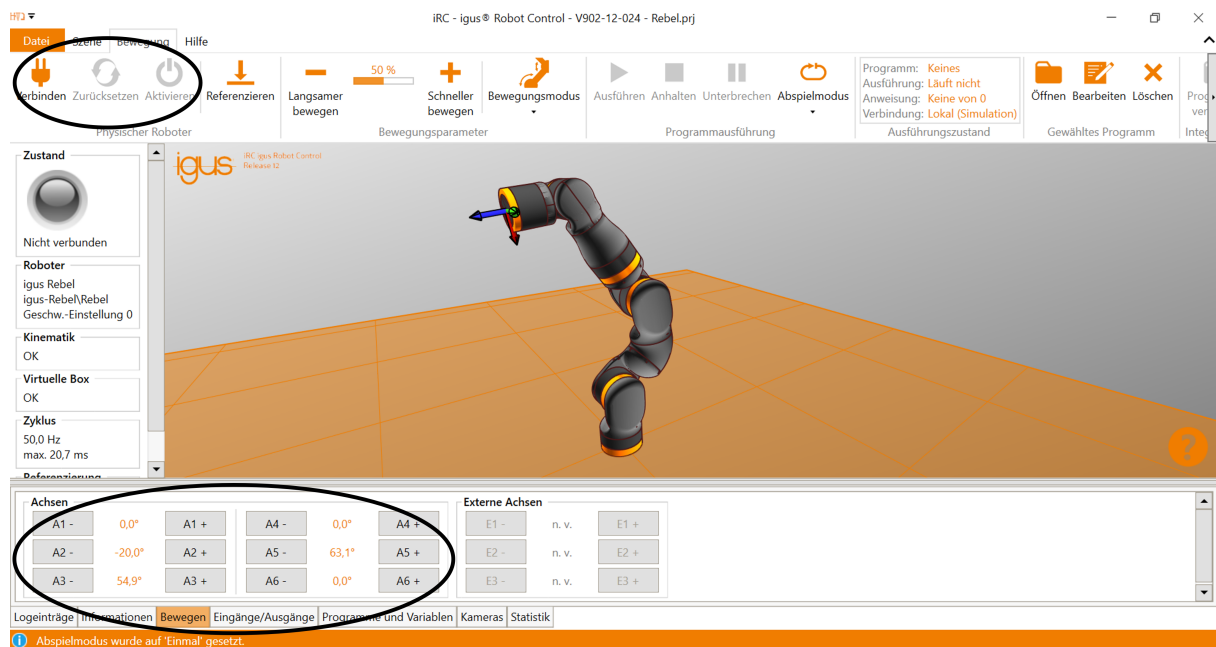


Abbildung 2: Jog-Befehle

## 6 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.

### 6.1 Die grafische Benutzeroberfläche

Dieser Abschnitt erklärt die iRC Software. Auch ohne einen angeschlossenen Roboter können alle Schritte simuliert werden. In Abschnitt 6.2 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.

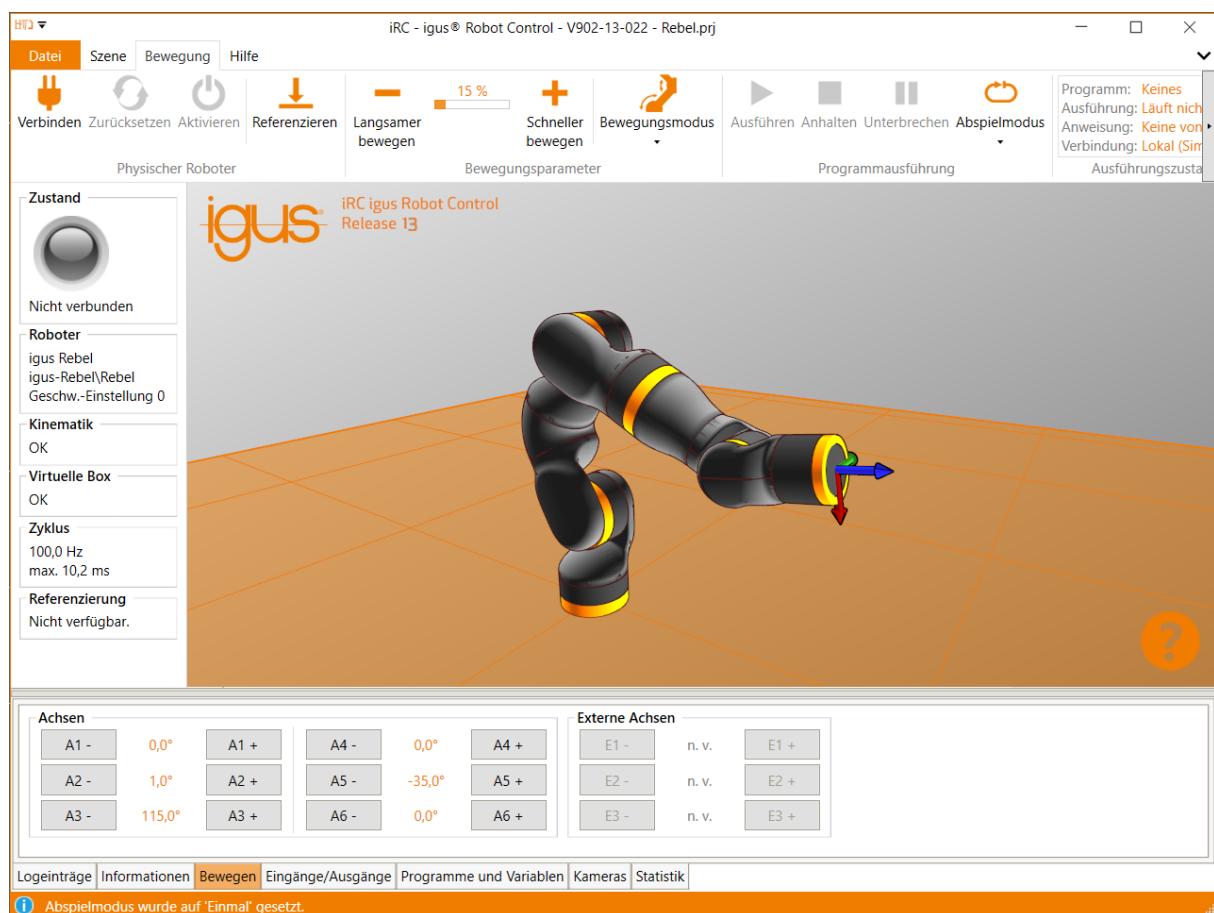


Abbildung 3: Benutzeroberfläche der iRC - igus Robot Control

In der linken oberen Ecke bieten die Registerkarten "Datei", "Szene", "Bewegung" und "Hilfe" Zugang zu den Hauptfunktionen. In der linken Ecke werden Informationen über den aktuellen Zustand des physischen Roboters angezeigt. Zusätzliche Funktionen wie das Laden eines anderen Projekts ("Projekt öffnen") oder "Roboter Referenzieren" finden Sie im Register "Datei" (s. Abb. 3).

Es gibt sieben Registerkarten am unteren Rand des Fensters:

- "Logeinträge": Meldungen des Programms über Status oder Fehler.
- "Informationen": zeigt die Achswerte, die kartesische Position und weitere Informationen an.
- "Bewegen": Tasten zum Bewegen des Roboters.
- "Eingänge/Ausgänge": Anzeigen und Einstellen der DIO-Schnittstellen der Robotersteuerung.

- "Programme & 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



Das "Hilfe"-Symbol in der rechten unteren Ecke enthält Links zu den Wiki-Seiten ("Online-Dokumentation", "Software-Updates", "Beispiele", "Fehlerbehebung"), eine Schaltfläche um den Support per E-Mail zu kontaktieren und eine Schaltfläche um die Logdateien von iRC und der integrierten Steuerung abzurufen.

### 6.1.1 Auswählen des Robotertyps

iRC bietet projektbezogene Einstellungen für verschiedene Robotertypen, wie z.B. Portale, Roboterarme oder Deltaroboter. Abbildung 4 zeigt den Projekt-Öffnen-Bereich, über den das entsprechende Projekt geladen werden kann.

1. Klicken Sie auf die Registerkarte "Datei" in der linken oberen Ecke und wählen Sie "Projekt öffnen".
2. Wählen Sie die Robotergruppe, z.B. "roboLink".
3. Stellen Sie sicher, dass der zur Betriebsspannung des Roboters passende Parametersatz gewählt ist. Dies beeinflusst die Geschwindigkeit und Beschleunigung.
4. Klicken Sie nun auf den Eintrag der zu Ihrem Roboter passt, z.B. RL-DP-5.

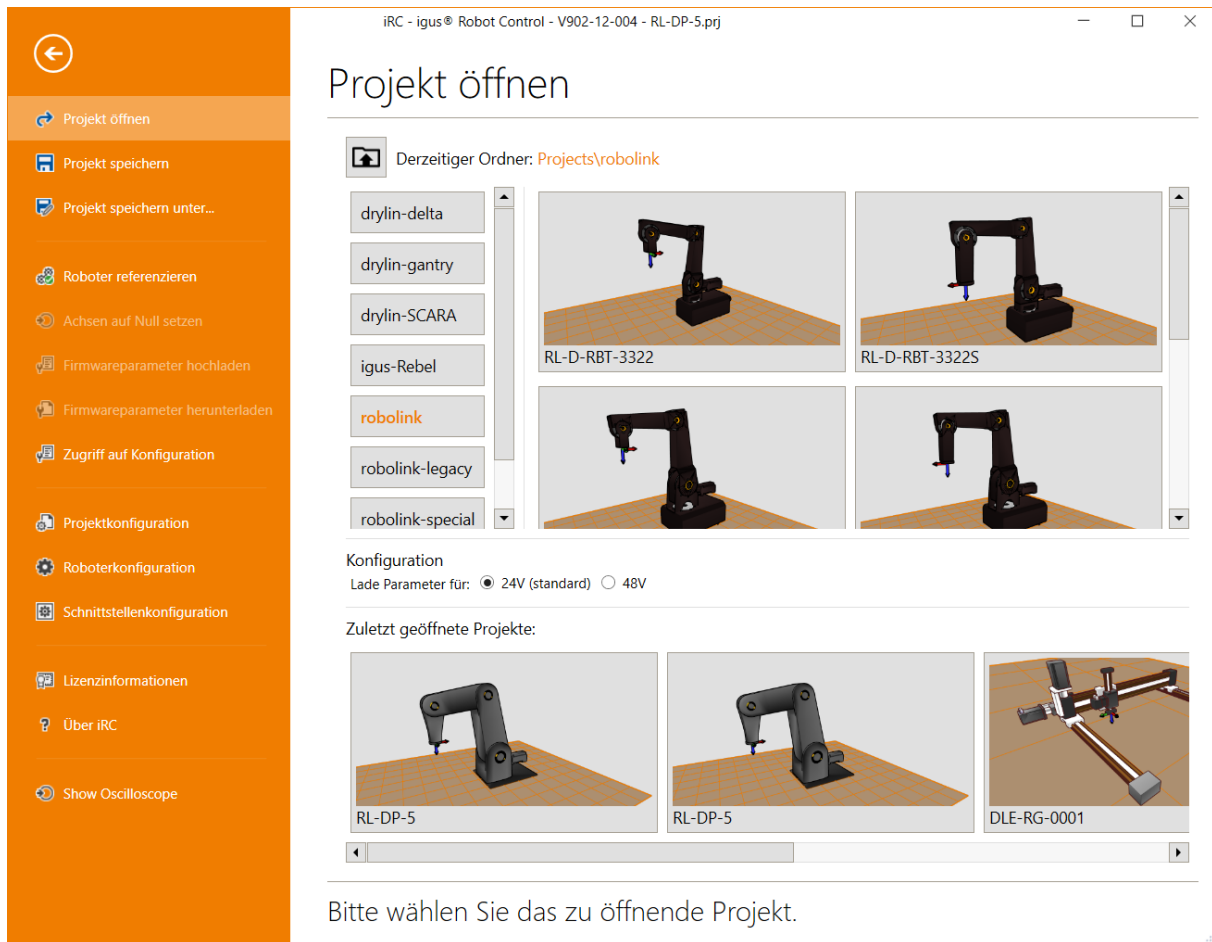
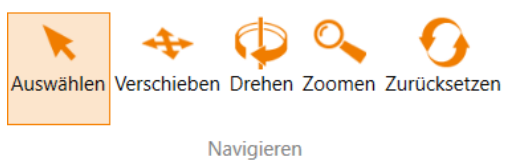


Abbildung 4: Auswahl des Robotertyps über den Menüpunkt "Datei" → "Projekt öffnen"

### 6.1.2 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:

- 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 "Szene" unter "Navigieren" geändert werden. Zur Auswahl stehen die Bewegungsoptionen "Auswählen", "Verschieben", "Drehen" und "Zoomen". "Zurücksetzen" bringt Sie zurück in die Startansicht.

## 6.2 Anschließen des Roboters


### 6.2.1 Verbindung zur Hardware

Der reale Roboter kann wie der simulierte gesteuert werden, nur die Hardware muss zuerst durch Klicken auf die Symbole "Verbinden", "Zurücksetzen" und "Aktivieren" in der Schaltflächengruppe "Physikalischer Roboter" im Register "Bewegung" verbunden und aktiviert werden (siehe Abb. 5). Danach muss der Roboter referenziert werden.



Abbildung 5: 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 die Verbindung initialisiert (meist per Ethernet, in einigen Fällen per USB-CAN-Adapter).
  - Die "Status"-Anzeige auf der linken Seite wechselt von grau zu rot.
  - Mehrere Fehlermeldungen werden unterhalb des "Status"-Indikators angezeigt.
2. "Zurücksetzen": 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, die in Abschnitt 6.3 beschrieben wird.

  - Die "Status"-Anzeige bleibt 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.
  - Die "Status"-Anzeige ist jetzt grün.

### 6.2.2 Den Roboter bewegen

Es ist jetzt möglich, den Roboter über die Jog-Tasten, mit einer Maus in der Benutzeroberfläche oder einem Gamepad zu bewegen, siehe Abschnitt 6.4.

### 6.3 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.

#### 6.3.1 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. 5).
3. Klicken Sie auf die Schaltfläche "Referenzieren" (s. Abb. 5) in der Schaltflächen-Gruppe "Physischer Roboter" im Register "Bewegung" (oder "Roboter Referenzieren" 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. 6. Mehrere Gelenke können die Referenzierung parallel durchführen.
5. Sie können auch auf "Referenziere alle" 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.

### 6.4 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

Die wichtigsten Einstellungen finden Sie im Register "Bewegung" (s. Abb. 7):

- "Eingabe": Verbinden eines Gamepads
- "Bewegungsparameter": Auswahl von Bewegungsmodi und Geschwindigkeit

Ein Gamepad ist möglicherweise die intuitivste Art den Roboter zu bewegen. Die Abb. 8 zeigt die 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. Weitere Informationen zur Verbindung finden Sie im Register "LogMessages" im unteren Bereich Fensterbereich.



Abbildung 6: Referenzierung der Achsen.

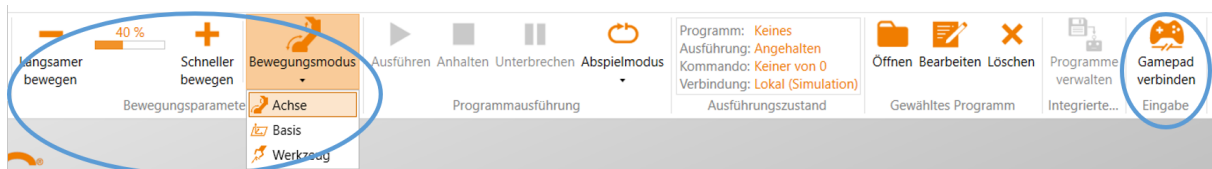


Abbildung 7: Bedienfelder zum Bewegen des Roboters (blau markiert).



1. Bewegungsmodus ändern
2. 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.
3. Programmanweisung Joint hinzufügen
4. Programmanweisung Linear hinzufügen

Abbildung 8: Tastenbelegung des Gamepads

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. 7):

- "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. 9).

- "Basis": (kartesischer Modus) bewegt den Roboter in geraden Linien entlang der X-, Y- und Z-Achse des Basiskoordinatensystems.
- "Werkzeug": (kartesischer Modus) bewegt den Roboter in X, Y und Z des aktuellen Werkzeugkoordinatensystems.

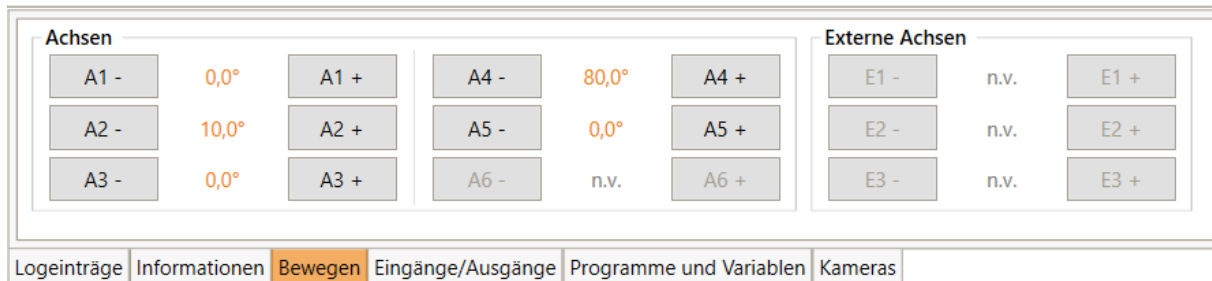


Abbildung 9: Die Softwareschaltflächen für "Achse"-Bewegungen. In beiden kartesischen Modi wechseln die Schaltflächen auf X, Y, Z, A, B und C.

## 6.5 Starten von Roboterprogrammen

Ein Roboterprogramm kann wie folgt geladen und gestartet werden:

1. Zum Laden des Programms klicken Sie auf das Ordnersymbol "Öffnen" in der Gruppe "Gewähltes Programm" der Registerkarte "Bewegung" und wählen Sie ein Programm, z.B. "testRobolink.xml".

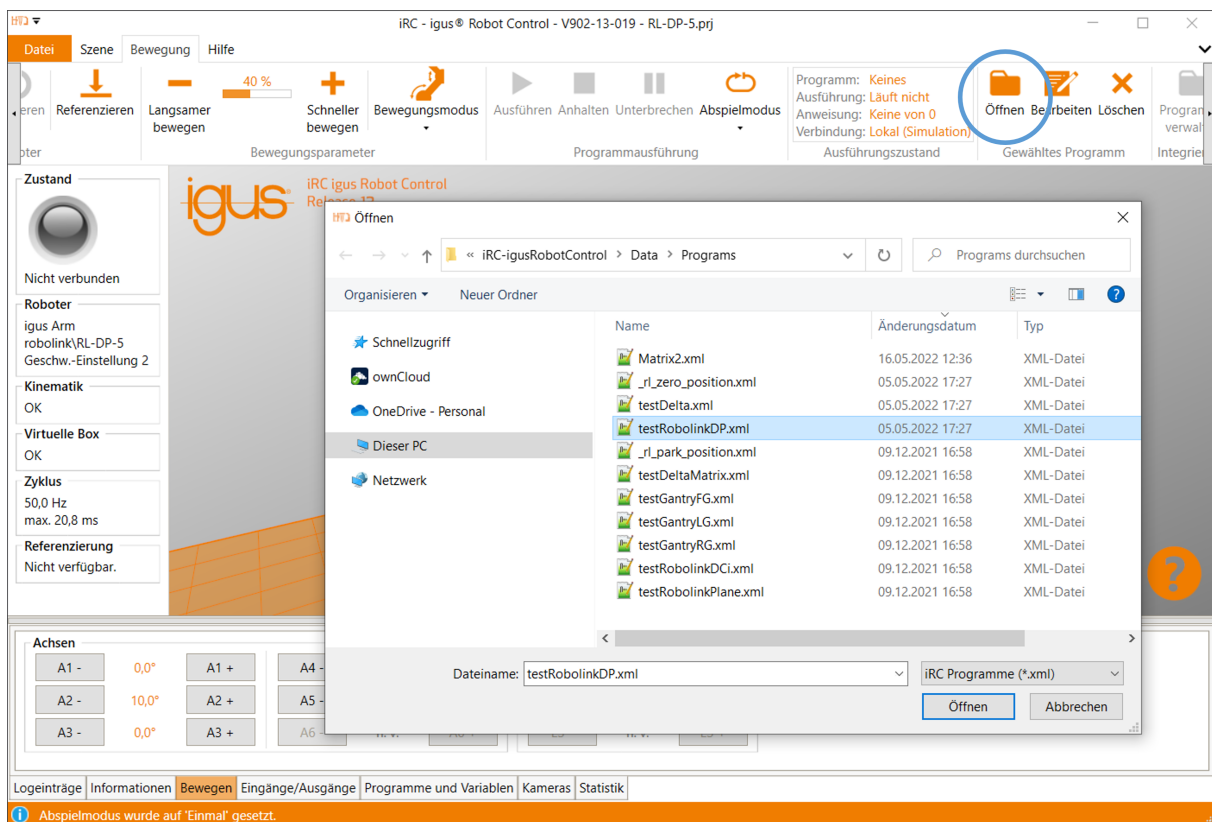
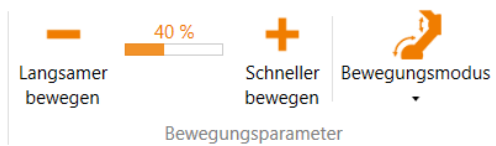


Abbildung 10: Laden eines Programms (blau markiert).

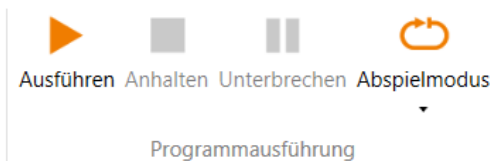


## 2. Stellen Sie die Grundgeschwindigkeit ein:



- 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 "Ausführen" in der Schaltflächen-Gruppe "Programmausführung" der Registerkarte "Bewegung".

## 4. Das Programm anhalten oder unterbrechen:

- Nachdem Sie das Symbol "Unterbrechen" gedrückt haben, kann der Roboter mit dem Programm fortfahren, indem Sie erneut auf das Symbol "Ausführen" klicken.
- Nach dem Drücken des "Anhalten"-Symbols startet das Programm mit dem ersten Befehl, wenn das "Ausführen"-Symbol erneut angeklickt wird.
- Der "Abspielmodus" kann auf drei verschiedene Werte eingestellt werden:
  - Einmal (das Programm stoppt nach einem einzigen Zyklus).
  - Wiederholen (das Programm stoppt nur durch "Anhalten" oder "Unterbrechen").
  - Einzelschritt (Dies ist nützlich für die Fehlersuche in einem Programm).

## 6.6 Digitale Ein- und Ausgänge

Der Zustand der Ein- und Ausgänge kann unter "Eingänge/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.

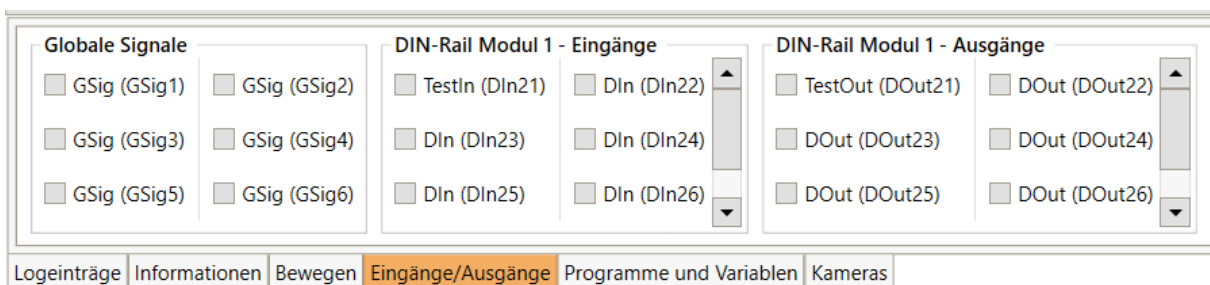


Abbildung 11: Ein-/Ausgabebereich der iRC - igus Robot Control.

Die Konfiguration der Ein- und Ausgänge ist in Abschnitt 8.1 beschrieben.

## 6.7 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 Implementation.
- 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

Siehe Abschnitt 9.2 für die Konfiguration dieser Schnittstellen.

## 6.8 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

## 7 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.

### 7.1 Programmierer

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 Programmierer wird mit der Schaltfläche "Bearbeiten" im "Bewegung"-Tab von iRC geöffnet.

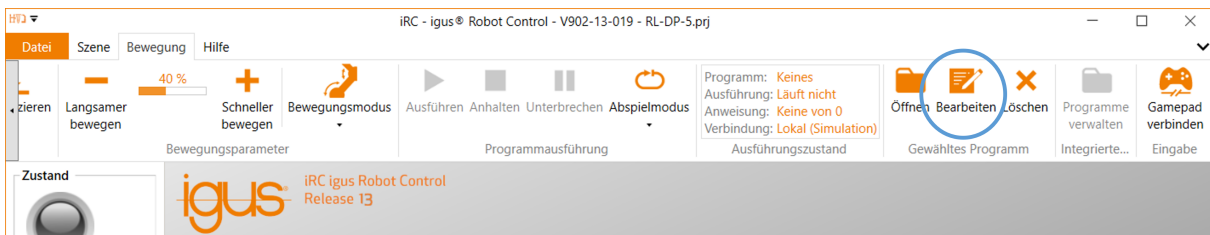


Abbildung 12: Öffnen des Programmierers mit der Schaltfläche "Bearbeiten".

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

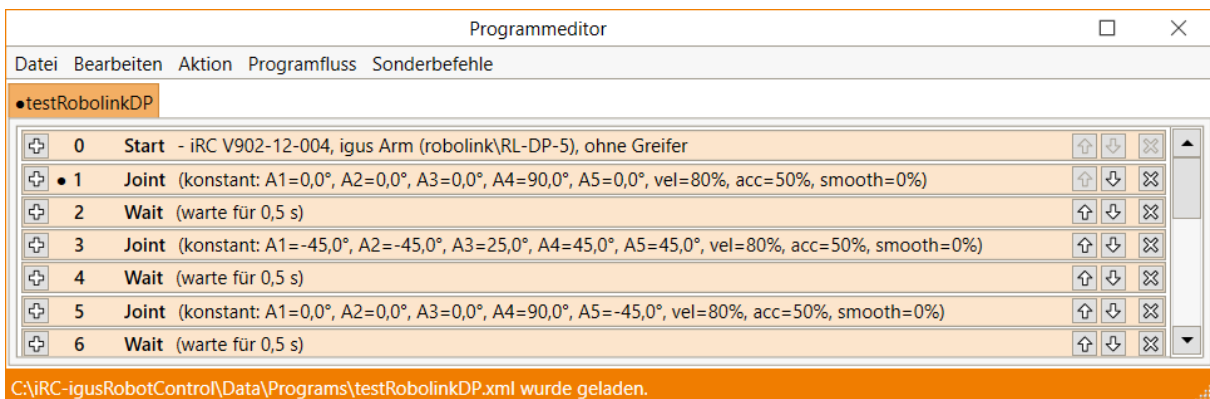


Abbildung 13: Programmierer mit einem kurzen Programm

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

#### 7.1.1 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. 14).

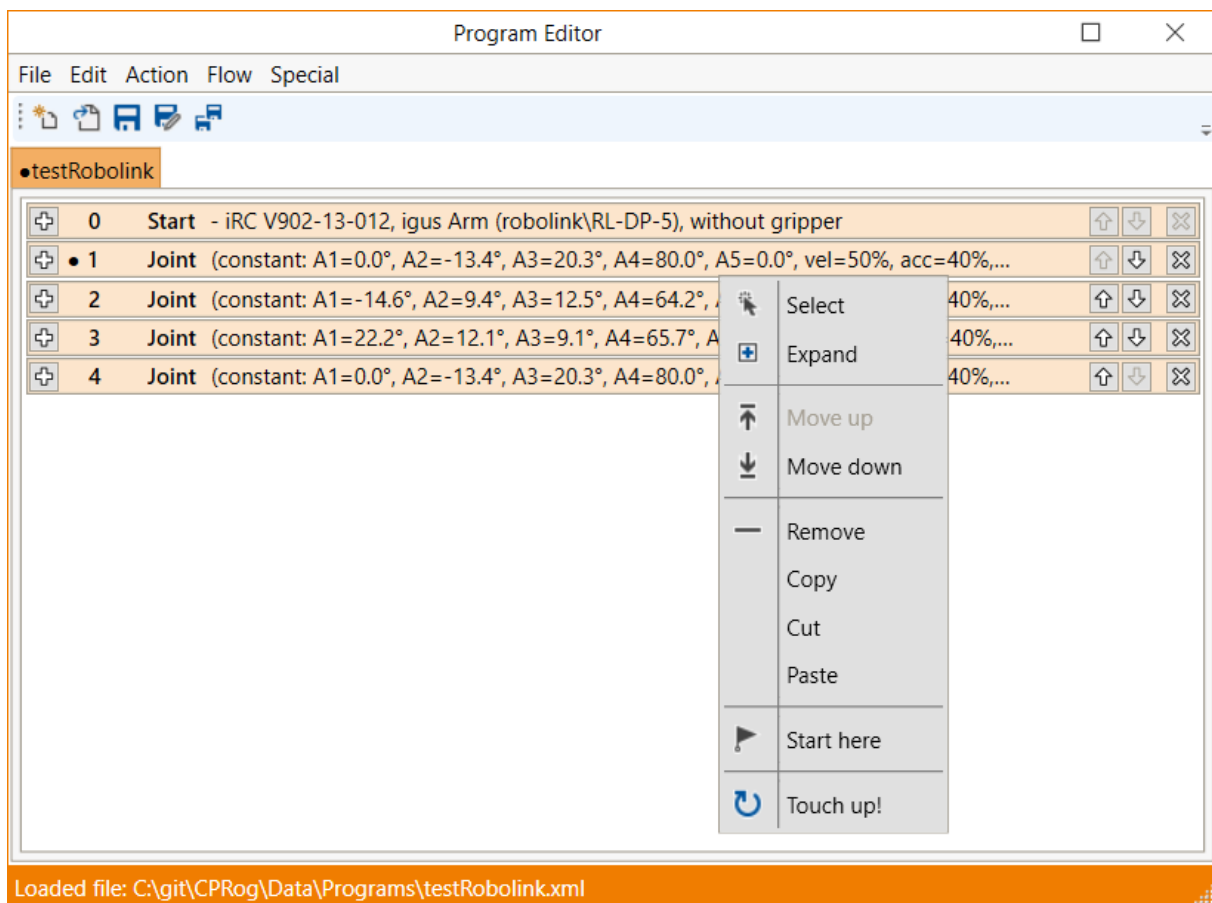


Abbildung 14: Das Kontextmenü einer Befehlszeile

Der Programmierer 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.

### 7.1.2 Position nachbessern

Bestimmte Befehle erfordern Positionswerte als Parameter. Häufig ist es wünschenswert, die aktuelle Position des Roboters 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. 14).

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

### 7.1.3 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 Programmierer 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. 14).

## 7.2 Kommentare und Informationen im Programm

### 7.2.1 Informationen zum Programm

Der Programmierer 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.

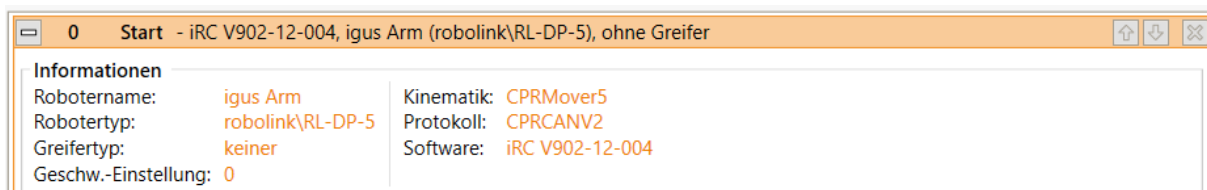


Abbildung 15: Die Start-Zeile enthält Informationen über die aktuelle Hardware.

### 7.2.2 Beschreibungen

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

### 7.2.3 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 Programmierer im Menüeintrag "Sonderbefehle" → "Kommentar" zu finden.

## 7.3 Bewegung

### 7.3.1 Abbruchbedingungen

Jeder Bewegungsbefehl kann mit einer Abbruchbedingung versehen werden. Es handelt sich um einen bedingten Ausdruck, der in Abschnitt 7.5.1 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.

### 7.3.2 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.

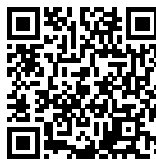
Das Überschleifen ist nur bei direkt auf einander folgenden Bewegungsanweisungen gleicher Art möglich. Z.B. können Linear und Kreisbewegungen mit einander überschleift werden und Achsbewegungen mit sich selbst. Wird eine Bewegungssequenz von einer Bewegung anderer Art oder einer

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 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".

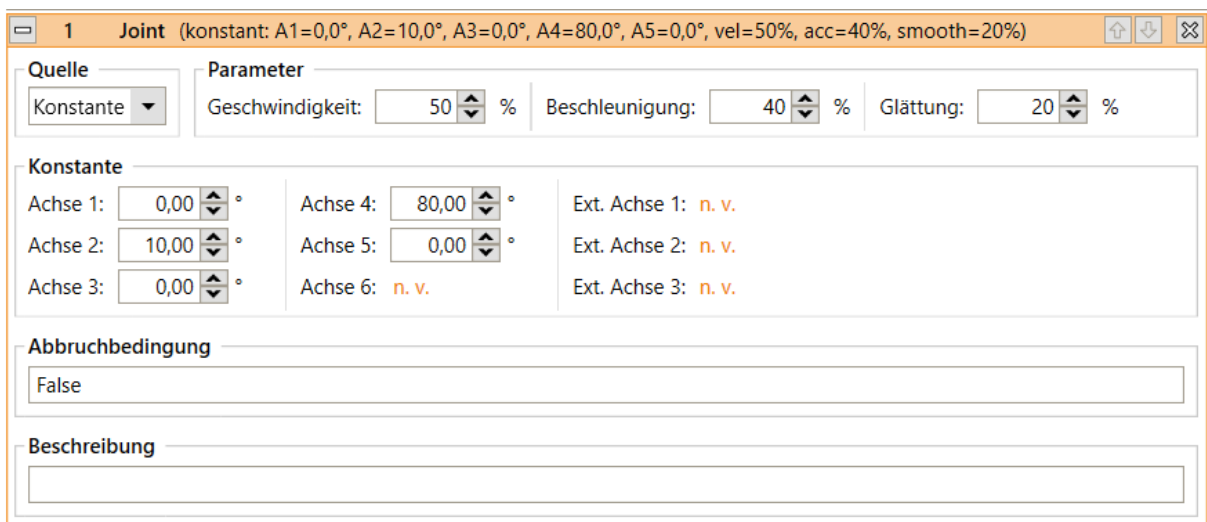
[https://wiki.cpr-robots.com/index.php/Motion\\_Something](https://wiki.cpr-robots.com/index.php/Motion_Something)



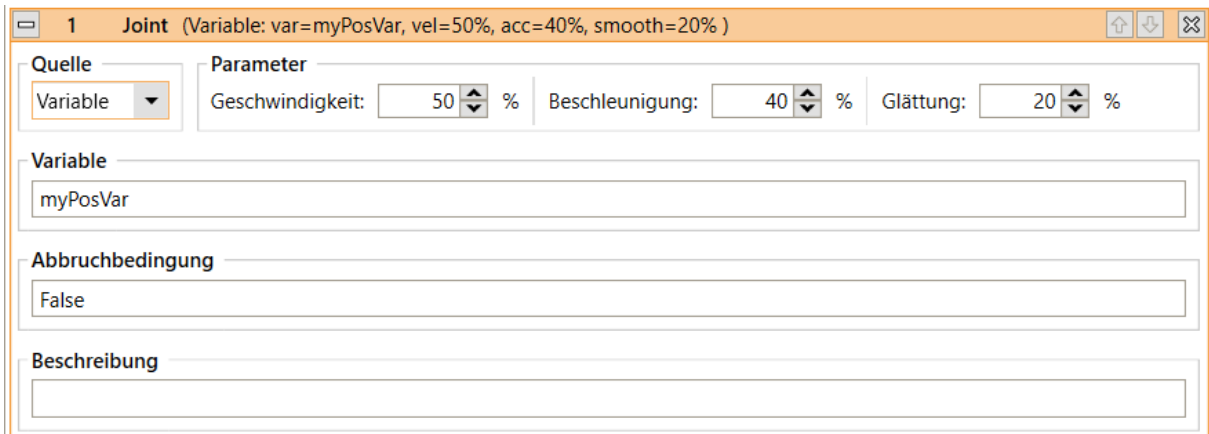
### 7.3.3 Achsbewegung

Der Befehl Joint 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.



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



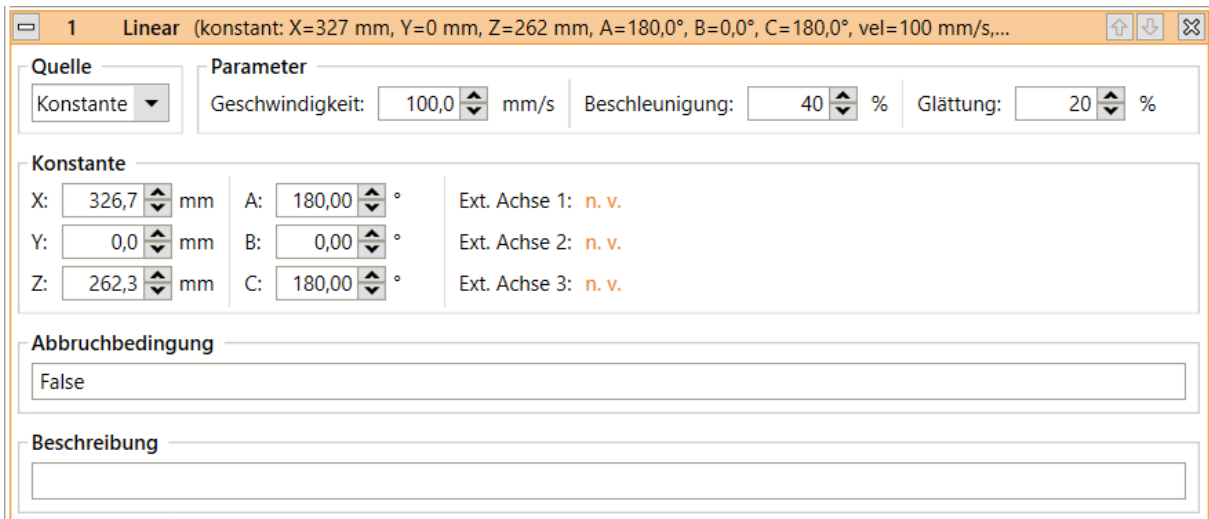
Die Bewegungsgeschwindigkeit wird durch "Geschwindigkeit" angegeben. Sie wird in Prozent der maximal erlaubten Bewegungsgeschwindigkeit für die jeweiligen Roboterachsen gemessen.

Der Joint-Befehl kann im Programmierer unter den Menüeinträgen "Aktion" → "Achsbewegung" und "Aktion" → "Variable Bewegung" → "Achsbewegung" aufgerufen werden.

#### 7.3.4 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.



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

1 Linear (Variable: var=myPosVar, vel=100 mm/s, acc=40%, smooth=20%)
↑ ↓ ✕

<b>Quelle</b> <input type="text" value="Variable"/>	<b>Parameter</b> Geschwindigkeit: <input style="width: 50px;" type="text" value="100,0"/> mm/s           Beschleunigung: <input style="width: 50px;" type="text" value="40"/> %           Glättung: <input style="width: 50px;" type="text" value="20"/> %
<b>Variable</b> <input style="width: 100%;" type="text" value="myPosVar"/>	
<b>Abbruchbedingung</b> <input style="width: 100%;" type="text" value="False"/>	
<b>Beschreibung</b> <input style="width: 100%; height: 20px;" type="text"/>	

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 Linear-Befehl kann im Programmierer unter "Aktion" → "Linearbewegung" und "Aktion" → "Variable Bewegung" → "Linearbewegung" aufgerufen werden.

### 7.3.5 Relative Bewegung

Der Befehl Relativ erlaubt es, den Roboter relativ zu seiner aktuellen Position zu bewegen. Er kann über die Menüpunkte unter "Aktion" → "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. Das für den Versatz verwendete Koordinatensystem ist das Roboterkoordinatensystem. 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.

### 7.3.6 Kreisbewegung

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



8 Circular (konstant: P1: X=380 mm, Y=50 mm, Z=260 mm, P2: X=430 mm, Y=0 mm, Z=260 mm,...

<b>Quelle</b>	<b>Parameter</b>		
Konstante ▼	Geschwindigkeit: 100,0 mm/s	Beschleunigung: 40 %	Glättung: 20 %

**Positions**

Der Kreis beginnt an der vorherigen Position. Setzen Sie zwei weitere Punkte auf dem Kreis um ihn zu definieren:

Position 1:	X: 380 mm	Y: 50 mm	Z: 260 mm
Position 2:	X: 430 mm	Y: 0 mm	Z: 260 mm
Zielorientierung:	A: 0 °	B: 0 °	C: 180 °
Ziel externe Achsen:	E1: 0 Einheiten	E2: 0 Einheiten	E3: 0 Einheiten

**Endposition**

Bewegungsende:  Position 2  Winkel: -180 °

**Abbruchbedingung**

False

**Beschreibung**

Die Kreisbahn wird definiert durch drei Punkte und optional einem Zielwinkel. Der Anfangspunkt ist die Position vor Start der Kreisbewegung, in der Regel die Zielposition der vorherigen Anweisung. Position 1 ist eine Hilfsposition auf der Kreisbahn, die überfahren wird. Position 2 ist die Zielposition auf der Kreisbahn. Wenn ein Zielwinkel angegeben ist kann die Bewegung vor Erreichen der Punkte stoppen oder darüber hinaus gehen. Bei Angabe von Winkeln größer 360° wird der Kreis mehrfach gefahren, negative Winkel kehren die Drehrichtung um.



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

Darüber hinaus können die Zielorientierung des Werkzeugs und die Zielpositionen der externen Achsen angegeben werden. Diese werden über die Kreisstrecke linear interpoliert.

Die Angabe der Positionen ist konstant oder über zwei Positionsvariablen möglich. Dabei sind die Zielorientierung und die externen Achsen in der Variable für Position 2 (Zielposition) anzugeben.



#### 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 Kreisbewegung 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.



### Programmierung einer Kreisbewegung durch Berechnung der Kreispunkte

Für einfache, an den Achsen des Koordinatensystems ausgerichtete Kreisbewegungen empfiehlt es sich 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.

#### 7.3.7 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 (siehe Abschnitt ??).

## 7.4 Greifer und digitale Ein-/Ausgänge

### 7.4.1 Digitale Eingänge

Digitale Eingänge und globale Signale sind als reservierte Wörter in bedingten Anweisungen verfügbar (siehe Abschnitt 7.5.1) Bei globalen Signalen sind dies GSig1, GSig2, etc. bei digitalen Eingängen Din1, Din2, etc. Je nach Zustand werden sie als "wahr" oder "falsch" ausgewertet.

### 7.4.2 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 Programmreditor unter "Aktion" → "Digitaler Ausgang" zugänglich.

### 7.4.3 Öffnen/Schließen des Greifers

Der Befehl Gripper ermöglicht die Steuerung des Greifers des Roboters. Er ist im Programmreditor über den Menüeintrag "Aktion" → "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%.

## 7.5 Programmfluss

### 7.5.1 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. Beispiele:

DIn23	Wahr, wenn Digitaleingang 23 gesetzt ist.
DIn23 AND !DIn27	Wahr, wenn Digitaleingang 23 gesetzt und 27 nicht gesetzt ist.
modelclass = 31	Wahr, wenn die Variable "modelclass" 31 ist
mempos.x > 350.0	Wahr, wenn die x-Komponente der Positionsvariablen "mempos" größer als 350 ist.

Die Syntax der bedingten Anweisungen ist durch die folgende EBNF-Definition gegeben:

Ausdruck	:= ["!"] <Boolean> <Boolescher Operator> <Boolean> ...
Boolean	:= <Boolesche Konstante>   <Ausdruck>   "(" <Ausdruck> ")"   Ver- glAusdruck   "(" <VerglAusdruck> ")"   <Digitaleingänge>   "(" <Digi- taleingänge> ")"
BoolescherOperator	:= "And"   "Or"
BoolescheKonstante	:= "True"   "False"
Digitaleingänge	:= <KanalTyp> <Kanalld>
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	:= "="   ">"   "<"   ">="   "<="

### 7.5.2 Stop

Der Befehl "Stop" stoppt die Programmausführung.

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

### 7.5.3 Pause

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

### 7.5.4 Wait

Der Befehl Wait 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.

### 7.5.5 If-then-else

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

Die angegebene Bedingung muss der in Abschnitt 7.5.1 beschriebenen Syntax entsprechen. Die Anweisungen zwischen "If" und "Else" werden ausgeführt, wenn die Bedingung als wahr ausgewertet wird. Andernfalls werden die Anweisungen zwischen "Else" und "EndIf" ausgeführt.

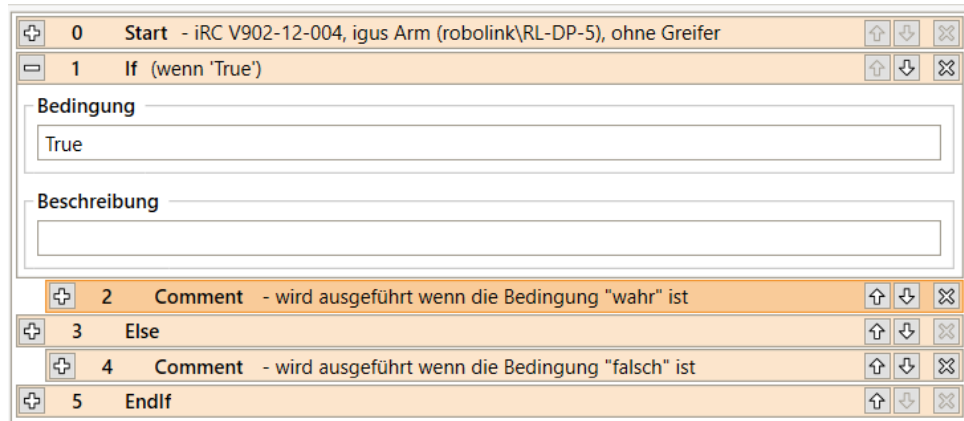


Abbildung 16: Die If-Anweisung verzweigt den Programmfluss.

### 7.5.6 Schleifen

Der Befehl Loop 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 7.5.1 beschriebenen Syntax entsprechen.
- "Zähler": Die Schleife wird die unter "Wiederholungen" angegebene Anzahl von Malen wiederholen.

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

### 7.5.7 Rasterbewegungen / Palettierung

Die Rasteranweisungen berechnen an einem Raster ausgerichtete Positionen, z.B. als Greif- oder Ablageposition für Palettierungsaufgaben. Abbildung 17 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" → "Schleife" → "Raster" hinzugefügt werden. Rasterdefinition und Rasterabfrage befinden sich im Menü "Sonderbefehle" → "Raster".

Die als "Punkt A", "Punkt B" und "Punkt C" angegebenen Positionsvariablen definieren die Ecken des Bereichs, der von der Rasterschleife abgedeckt wird (siehe Abbildung 17). Die Anzahl der durchzufüh-

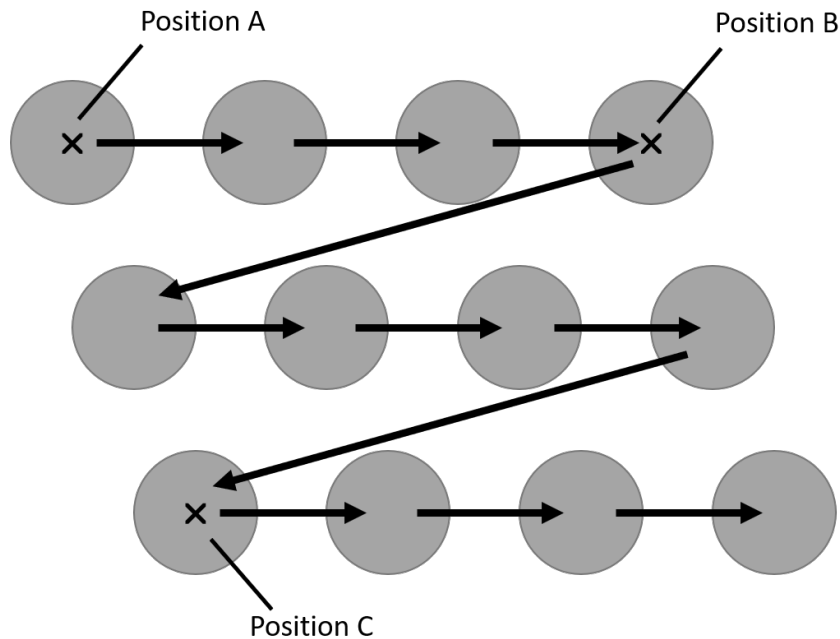


Abbildung 17: Die Matrixbewegung erfolgt von Punkt A nach B, dann in Richtung C versetzt.

renden 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  $Y=3$ .

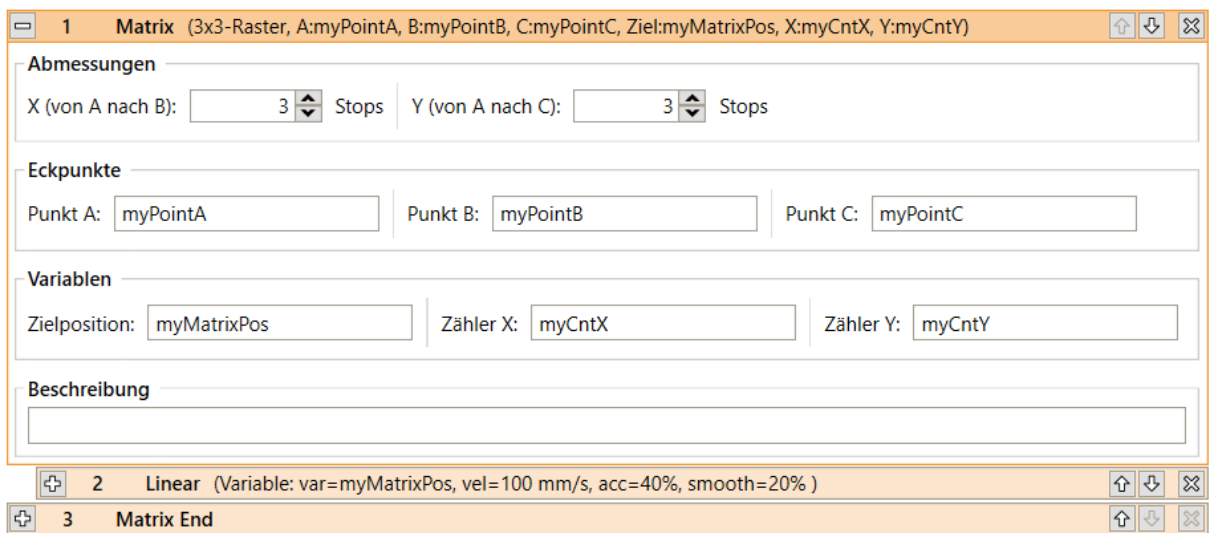


Abbildung 18: Definition einer Rasterschleife.

Der Block zwischen "Matrix" und "Matrix End" wird für jeden 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.

Die Anweisungen Rasterdefinition und -abfrage 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.

+	1	Matrix Definition (Name=myMatrix, Dimensionen=3x3, A=myPointA, B=myPointB, C=myPointC)	↑	↓	✕
+	2	Loop (wiederhole bis 'myIndex < 10')	↑	↓	✕
	+	3 Matrix Query (Name=myMatrix Index=myIndex, Ziel=myMatrixPos, X=myCntX, Y=myCntY)	↑	↓	✕
	+	4 Math (addiere 1 zu myindex)	↑	↓	✕
	+	5 Linear (variabel: var=mymatrixpos, vel=100 mm/s, acc=40%, smooth=0% )	↑	↓	✕
+	6	EndLoop	↑	↓	✕

Abbildung 19: Verwendung der Rasterdefinition- und Rasterabfrageanweisung. Die Definition der Variablen wurde ausgelassen.

### 7.5.8 Unterprogramme

Mit dem Sub-Befehl können Unterprogramme aufgerufen werden.

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

## 7.6 Variablen und Variablenzugriff

In Programmen für iRC und TinyCtrl 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.

### 7.6.1 Benutzervariablen

Es ist möglich, Benutzer-Variablen mit dem Befehl Store zu definieren, der im Programmeditor über die Menüeinträge unter "Sonderbefehle" → "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. 20).
- "Positionskonstante":  
Eine Positionsvariable wird mit den unter "Kartesische Position", "Achspanposition" und "Externe Achsen" angegebenen Konstanten initialisiert (s. Abb. 21). Je nach dem kinematischen Modell des aktuellen Roboters kann es sein, dass bestimmte Achsen nicht verfügbar sind.

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.

### 7.6.2 Systemvariablen

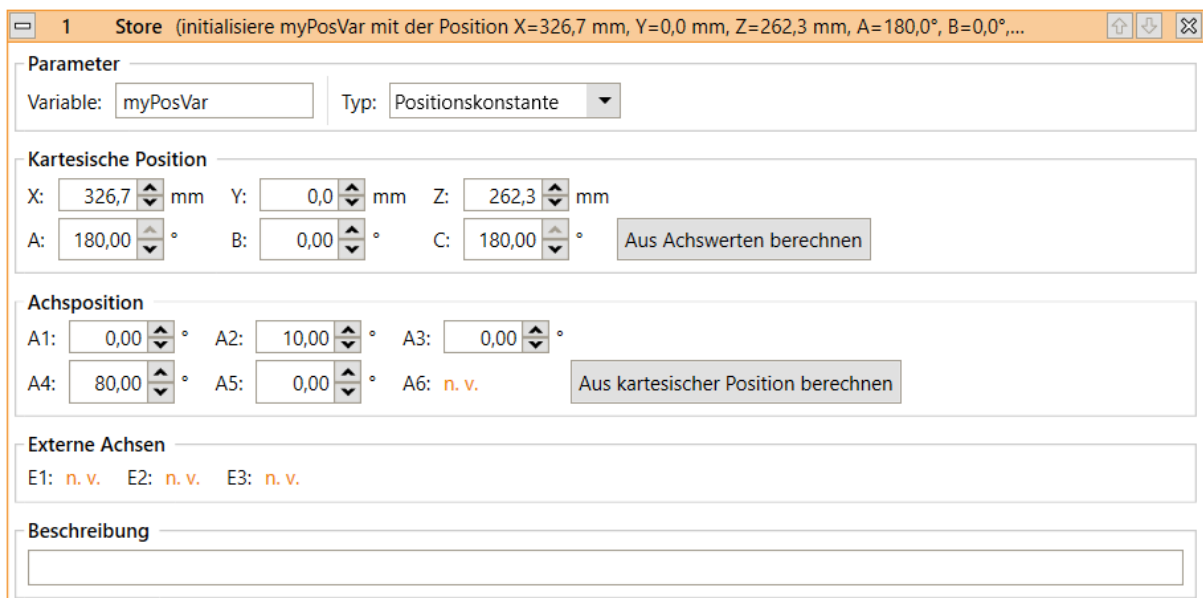
Die folgenden vordefinierten Variablen stehen zur Verfügung, ohne dass sie definiert werden müssen:

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



The screenshot shows a dialog box titled "1 Store (initialisiere myPosVar mit der Zahl 0)". It has a "Parameter" section with a "Variable:" field containing "myPosVar", a "Typ:" dropdown menu set to "Zahlenkonstante", and a "Wert:" field with a spinner set to "0,00". Below this is a "Beschreibung" section with an empty text area.

Abbildung 20: Definition einer Zahlenvariable.



The screenshot shows a dialog box titled "1 Store (initialisiere myPosVar mit der Position X=326,7 mm, Y=0,0 mm, Z=262,3 mm, A=180,0°, B=0,0°,...)". It has a "Parameter" section with a "Variable:" field containing "myPosVar" and a "Typ:" dropdown menu set to "Positionskonstante". Below this are two sections for position definition: "Kartesische Position" with fields for X (326,7 mm), Y (0,0 mm), Z (262,3 mm), A (180,00°), B (0,00°), and C (180,00°), and a button "Aus Achswerten berechnen"; and "Achspannung" with fields for A1 (0,00°), A2 (10,00°), A3 (0,00°), A4 (80,00°), A5 (0,00°), and A6 (n. v.), and a button "Aus kartesischer Position berechnen". At the bottom, there is an "Externe Achsen" section with fields for E1, E2, and E3, all set to "n. v.". A "Beschreibung" section with an empty text area is at the very bottom.

Abbildung 21: Definition einer Positionsvariable.

- #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 "#"

### 7.6.3 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".

### 7.6.4 Berechnungen mit Variablen

Berechnungen mit Variablen können mit dem Befehl Math ausgeführt werden, der im Programmmeditor über die Menüeinträge "Sonderbefehle" → "Variablenoperation" 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						

### 7.6.5 Variablen beobachten

Sie können die aktuellen Werte aller definierten Variablen in iRC in der Registerkarte "Programme und Variablen" im Statusbereich beobachten.

## 7.7 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 9.2.5). Die Programmanweisung kann über den Menüeintrag "Sonderbefehle" → "Kamera" hinzugefügt werden.



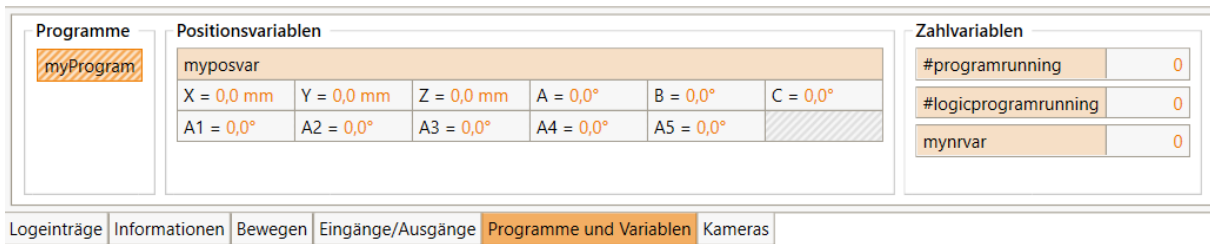


Abbildung 22: Die Werte der Variablen werden im Infobereich angezeigt.

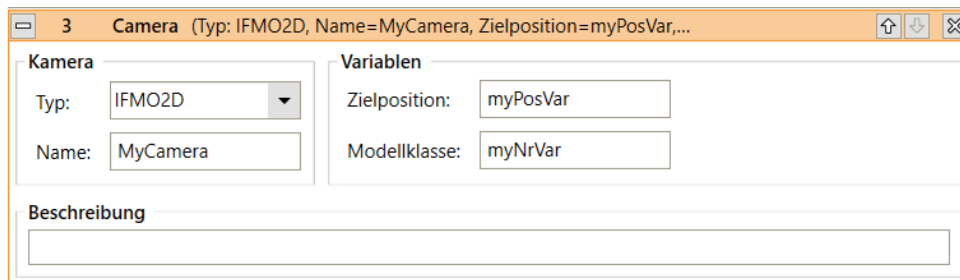


Abbildung 23: Kameraanweisung im Programmeditor.

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".



Die Kameraanweisung wartet nicht, wenn kein Objekt erkannt wurde. Prüfen Sie durch eine If-Bedingung oder eine Bedingungsschleife, 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.

## 8 Hardwarekonfiguration

Um zusätzliche Hardware wie Schalter, SPS oder Aktoren 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 9.

### 8.1 Ein-/Ausgänge

Die Steuerung unterstützt verschiedene Arten von digitalen Ein- und Ausgängen (DIO), die im Projektkonfigurationsbereich 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. Die Rebel-Steuerung ignoriert sie.
- DIN-Rail Ein-/Ausgänge: Der Rebel besitzt zwei integrierte Ein-/Ausgabemodule. Bei DIN-Rail-Steuerungen können bis zu 3 Ein-/Ausgabemodule angeschlossen und konfiguriert werden.
- 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.

#### 8.1.1 Elektrische Integration

Der einfachste Weg, Schalter, Aktoren oder SPS anzuschließen ist über digitale Ein- und Ausgänge. Der Rebel wird mit zwei integrierten DIO-Modulen geliefert; an der Basis stehen vier digitale Eingänge zur Verfügung, am Arm je zwei Ein- und Ausgänge.

Die Eingänge an der Basis sind von der Robotersteuerung galvanisch getrennt. Um sie anzusteuern muss eine Spannung (ca. 24V) zwischen einem der DIn-Pinne und External Ground angelegt werden. Die Ein- und Ausgänge am Arm sind im Gegensatz dazu nicht galvanisch getrennt, hier steht am Stecker 24V und GND aus der Robotersteuerung zur Verfügung.

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.

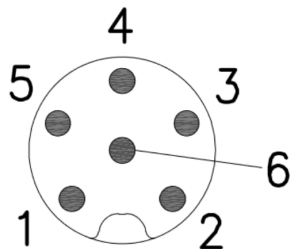


Abbildung 24: Belegung der DIO-Buchse am Arm

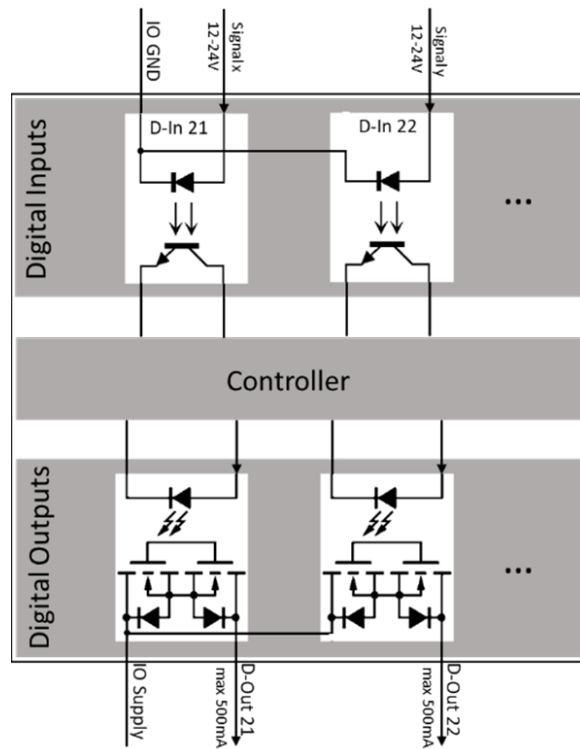


Abbildung 25: Interner Aufbau eines DIO-Moduls

Pin	Belegung
1	DIn21
2	DIn22
3	DIn23
4	DIn34
5	External GND

Tabelle 7: Belegung der DIO-Buchse an der Basis (links nach rechts)

Pin	Belegung	Kabelfarbe (entsprechend binder-Kabel)
1	+24VDC (max 500mA)	Braun
2	DIn32	Weiß
3	GND	Blau
4	DIn31	Schwarz
5	DOut32	Grau
6	DOut31	Rosa

Tabelle 8: Belegung der DIO-Buchse am Arm (s. Abb. 24)

### 8.1.2 Software-Konfiguration

Die Software und integrierte Steuerung sind für die beiden DIO-Module des Rebel vorkonfiguriert.



Ändern Sie im Konfigurationsbereich (Datei → Projektkonfiguration → Ein-/Ausgänge) nicht die Anzahl der DIN-Rail Ein-/Ausgänge! Wenn die Module aus der Konfiguration entfernt und wieder hinzugefügt werden weist die automatische Konfiguration IDs nach einem mit dem Rebel inkompatiblen Schema zu. Dies kann nur manuell oder durch ein Werksreset per TinyCtrl-Updater zurückgesetzt werden.

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 welchen Zustand ein Ausgang bei Reset wechselt, der Fehlerzustand wird angenommen, wenn ein Fehler auftritt.

### 8.1.3 Sensoren und Taster an der Basis anschließen

- Pin 5 (External GND) muss mit dem negativen Pol (-) der Stromversorgung des Sensors verbunden werden.
- Das Sensorsignal (positiv) muss an einen Eingangspin angeschlossen werden.
- Die positive Seite des Sensors (+) muss an die Stromversorgung des Sensors angeschlossen werden.
- Der Status der Eingänge kann im Register "Input/Output" unten in iRC überwacht werden.
- Ein Roboterprogramm kann Eingaben abfragen und darauf reagieren, z.B. mit einer if-then-else-Anweisung

### 8.1.4 Sensoren und Taster am Arm anschließen

- Pin 1 (+24V) des DIn-Anschlusses muss mit dem positiven Pol des Sensors verbunden werden.
- Das Sensorsignal muss an einen Eingangspin angeschlossen werden.
- Der Status der Eingänge kann im Register "Input/Output" unten in iRC überwacht werden.
- Ein Roboterprogramm kann Eingaben abfragen und darauf reagieren, z.B. mit einer if-then-else-Anweisung

### 8.1.5 Aktoren am Arm anschließen

- Pin 3 (GND) des Steckers muss mit GND des Aktors verbunden werden.
- Der Aktor (Relais usw.) wird dann über einen freien DOut-Pin mit Strom versorgt.
- Sie können die Ausgaben manuell im Register "Eingabe / Ausgabe" unten in iRC einstellen.
- Ein Roboterprogramm kann den Zustand der Ausgänge mit der Digital-Out-Anweisung einstellen.

## 8.2 Motorbremse

Um ein Absacken der Achsen zu verhindern sind im Rebel elektromagnetische Bremsen verbaut. Durch Anlegen einer Spannung werden die Achse freigegeben. Ohne Spannung fallen sie durch Federn in den Bremszustand.

Die Bremsen des Rebel werden über die Motorelektronik gesteuert, daher ist keine zusätzliche Konfiguration nötig. Die Einstellungen im Konfigurationsbereich von iRC haben keinen Einfluss.

## 8.3 Konfiguration der Motorsteuerungen

Für die Feineinstellung der Bewegung und der Referenzierung enthält jedes Achsmodul einen eigenen Konfigurationssatz. Dieser kann über die Schaltflächen "Datei" → "Firmwareparameter herunterladen" bzw. "Firmwareparameter hochladen" abgerufen und geändert werden. Nach dem Herun-

terladen wird der Konfigurationssatz im Installationsverzeichnis von iRC unter Data\Backup angelegt.

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.

## 9 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" geöffnet werden kann (siehe Abb. 26). Das Projekt betreffende Einstellungen finden Sie dabei unter "Projektkonfiguration", projektübergreifende Einstellungen unter "Roboterkonfiguration". Die Schnittstellen können über "Schnittstellenkonfiguration" ebenfalls projektweise konfiguriert werden.

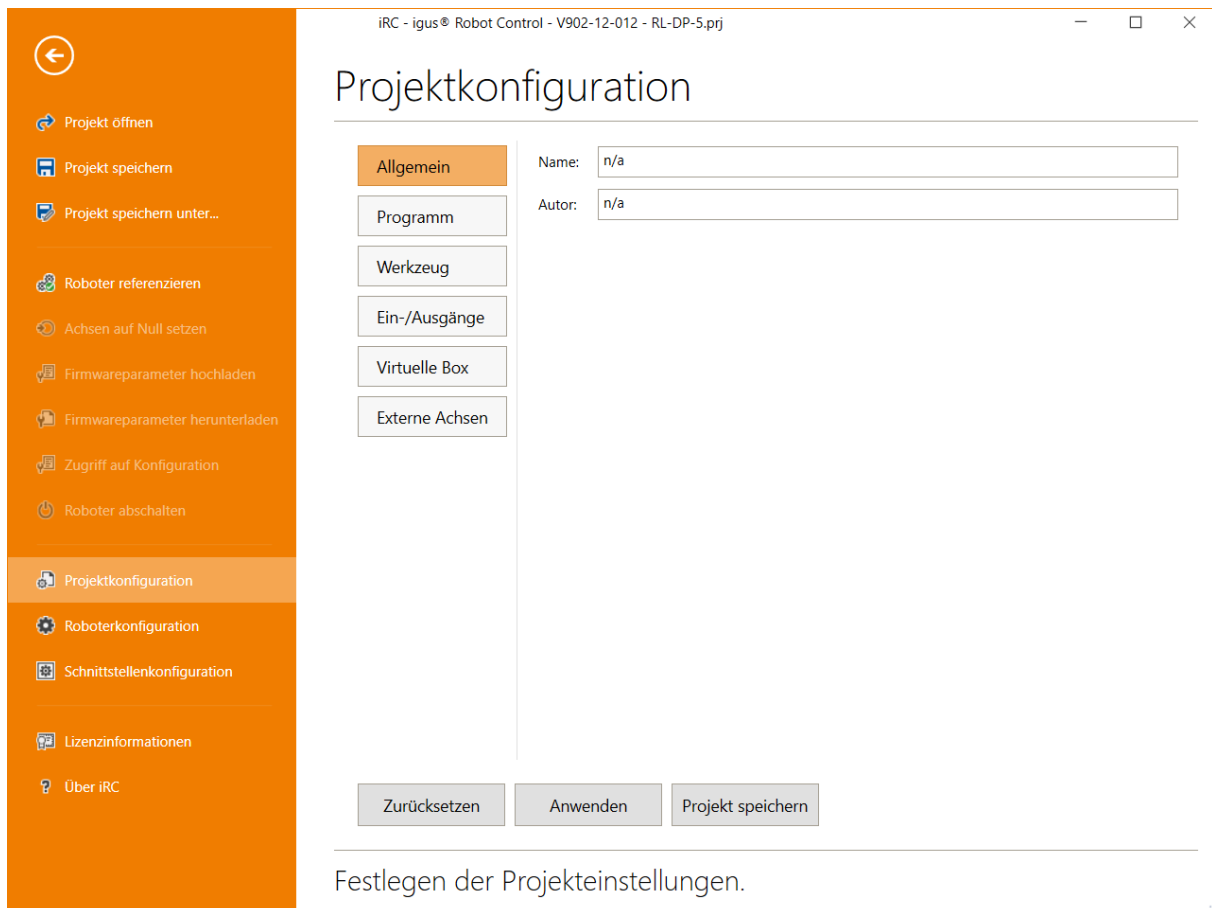


Abbildung 26: Der Projektkonfigurationsbereich.

Speziellere Einstellungen können über die Projekt-, Roboter-, und Werkzeugkonfigurationsdateien vorgenommen werden. Die Einstellungen der Achsmodule lassen sich über "Firmwareparameter hoch-/herunterladen" abrufen und ändern (s. Abs. 8.3).



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!

Falls Sie einen Roboter mit integrierter Steuerung (TinyCtrl) verwenden müssen die Änderungen auch dort vorgenommen werden. Bei Änderungen über den Konfigurationsbereich verbinden Sie vorher den Roboter. Bei Klick auf "Anwenden" oder "Projekt speichern" werden die Änderungen automatisch mit der Robotersteuerung synchronisiert. Änderungen an den Konfigurationsdateien müssen manuell übertragen werden, verwenden Sie dazu den Bereich "Zugriff auf Konfiguration".



Einige Änderungen der integrierten Robotersteuerung werden erst nach einem Neustart übernommen. Warten Sie nach dem Übertragen mindestens 20 Sekunden und starten Sie den Roboter neu.

## 9.1 Projektkonfiguration

### 9.1.1 Programm

Hier können Sie das Roboter- und Logikprogramm, die Bewegungsgeschwindigkeit (in Prozent der Höchstgeschwindigkeit), den Abspielmodus und die Reaktion bei Programmfehlern festlegen.

### 9.1.2 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 öffnen Sie den Bereich "Datei" → "Werkzeugkonfiguration", klicken Sie die Schaltfläche "Hinzufügen" im Bereich "Werkzeugkonfiguration" und wählen Sie die neue Werkzeugkonfigurationsdatei.

### 9.1.3 Ein-/Ausgänge

Hier kann die Anzahl der Ein-/Ausgabemodule und das Verhalten der Ein-/Ausgänge eingestellt werden. Die Basis-Ein-/Ausgänge sind nur für Roboter der Mover-Serie relevant, alle anderen Roboter werden über den Bereich "DIN-Rail Ein-/Ausgänge" konfiguriert. 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.

### 9.1.4 Virtuelle Box

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

## 9.2 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.

### 9.2.1 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
Aktivierung	Setzt die Steuerung zurück und Aktiviert dann die Motoren
Referenzierung	Startet die Referenzierung aller Achsen
Start	Startet die Ausführung des geladenen Programms, stoppt die Ausführung eines laufenden Programms oder führt ein pausiertes Programm fort
Pause	Pausiert ein laufendes Programm oder führt ein pausiertes Programm fort
Alt-Start-Programm	Ein-Tasten-Steuerung: Führt bei mehrmaligen Drücken des Tasters nacheinander die folgenden Funktionen aus: Reset, Aktivieren, Referenzieren (wenn noch nicht erledigt), Programm starten
Alt-Stop-Programm	Ein-Tasten-Steuerung: Führt bei mehrmaligen Drücken des Tasters nacheinander die folgenden Funktionen aus: Pause, Stopp, Reset
Herunterfahren	Führt den Steuerungsrechner herunter
Starte Plattform-Mission	Startet die Mission der mobilen Plattform
Achs-Anweisung hinzufügen	Achs-Anweisung zur aktuellen Position im Programmeditor hinzufügen
Linear-Anweisung 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
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



Parameter	Bedeutung
Plattform-Mission läuft	Die mobile Plattform führt eine Mission aus

Die Konfiguration der SPS-Schnittstelle erfolgt über den Konfigurationsbereich in iRC (Datei → Schnittstellenkonfiguration → SPS-Schnittstelle). Um einen Roboter mit integrierter Steuerung zu konfigurieren muss dieser verbunden sein. Das Feld "Aktiv" aktiviert die Schnittstelle, "Automatisch verbinden" führt dazu, dass iRC versucht sich automatisch mit dem Roboter zu verbinden. Die Zahlenfelder zu den Ein- und Ausgängen entsprechen den Nummern der digitalen Ein-/Ausgänge. Um einzelne Funktionen zu deaktivieren wählen Sie eine Nummer, die an keinem Hardwaremodul vorhanden ist, z.B. "1".

### 9.2.2 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 Eingangsignals: 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

### 9.2.3 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 10.

Die Modbus-Schnittstelle kann über die Schnittstellenkonfiguration 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)

### 9.2.4 CRI-Schnittstelle

Die CRI-Schnittstelle ermöglicht das Senden komplexer Anweisungen und den Abruf von Informationen und Einstellungen über die Ethernetschnittstelle per TCP/IP. iRC verwendet diese Schnittstelle um sich mit Robotern mit integrierter Steuerung oder anderen Instanzen von iRC zu verbinden. Standardmäßig ist diese Schnittstelle in iRC deaktiviert.

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)



Um die Schnittstelle in iRC zu aktivieren öffnen Sie den Konfigurationsbereich (Datei → Schnittstellenkonfiguration → CRI-Schnittstelle). Die Schaltfläche "Starten" startet den CRI-Server, "Anhalten" stoppt ihn. Das Statusfeld zeigt die gebundene IP-Adresse und Port sowie die Anzahl verbundener Clients an. Falls eine bestimmte IP-Adresse verwendet werden soll kann Sie im Feld darunter eingetragen werden.

### 9.2.5 Kameraschnittstelle

Die Kameraschnittstelle ermöglicht die Verwendung von Objekterkennungs- und Videokameras. Objekterkennungskameras erkennen die Position und Klasse von Objekten und übertragen diese, optional mit einem Videobild, an die Steuerung. Aus den Objektpositionen der Kamera berechnet die Steuerung 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 O2D Objekterkennungskameras und 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 unten erklärt die Kompatibilität.

#### Kameraschnittstelle für Objekterkennung

Zur Objekterkennung werden derzeit Kameras von ifm der Serie O2D200 unterstützt sowie Kameras die das 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)).

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

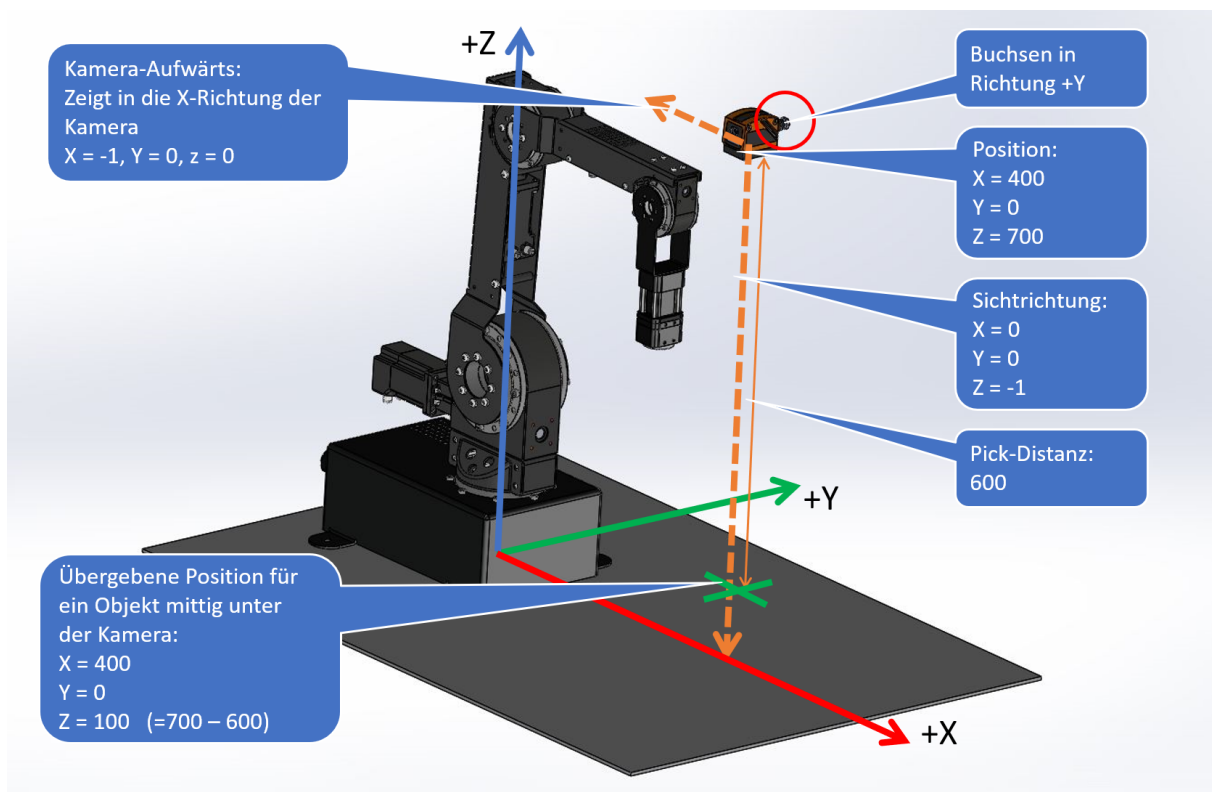


Abbildung 27: Vermessung der Kameraposition und -ausrichtung.

Parameter	Bedeutung
Aktiviert	Aktiviert oder deaktiviert die Kamera. Im deaktivierten Zustand werden die Werte aus dem Simulationsbereich verwendet (nur iRC)
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".
Name	Name der Kamera im Roboterprogramm
Beschreibung	Optionale Beschreibung
IP-Adresse	IP-Adresse der Kamera
Port	Portnummer der Kamera

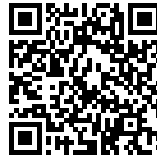
Die Einträge im Geometrie-Bereich bestimmen die Verarbeitung der von der Kamera gelieferten Positionen (Pixelposition des Kamerabilds) zu Positionen relativ zum Roboter (s. Abb. 27):

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



Eine Schritt-für-Schritt-Anleitung und Tipps zur Parametrierung 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)



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

Parameter	Bedeutung
SimX, SimY	Objektposition in Pixel (0-640 bzw. 0-480)
SimA	Orientierung in Grad
Modellklasse	Klasse des erkannten Objekts, der Wert -1 bedeutet dass kein Objekt erkannt wurde

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

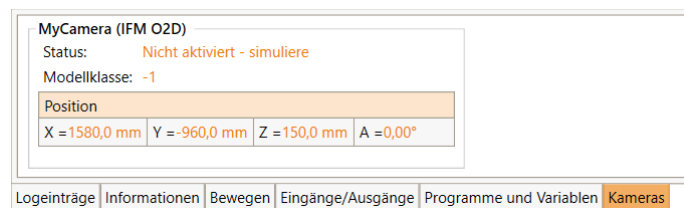


Abbildung 28: Kamera-Statusbereich

### 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 iRC übertragen und kann dort im Kamera-Statusbereich beobachtet werden (s. Abb. 28).
- 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 9.2.7).



#### Kompatibilität:

Die USB-Kameraschnittstelle wird nur von integrierten Steuerungen unterstützt, die auf dem Raspberry Pi basieren. Ältere Robotersteuerungen die von einem Phytec-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 9.2.7).

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.

### 9.2.6 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. Phytec-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
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.

### 9.2.7 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 9.2.6 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
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.

### 9.2.8 Secure Shell Zugriff

Auf den Steuerrechner kann z.B. zu Wartungszwecken per Secure Shell (ssh) (Port 22) 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)





Die Windows-Software iRC und die integrierte Steuersoftware TinyCtrl benutzen Konfigurationsdateien. Wenn auf einem dieser Systeme manuell (z.B. in einem Texteditor) Änderungen in den Konfigurationsdateien gemacht werden, müssen diese auch auf der anderen Seite durchgeführt werden. Sonst verhalten sich die Steuerungen nicht identisch, es kann zu Kollisionen kommen!

Über die Software PuTTY kann man sich mit dem Linux-Board verbinden und auf einer Kommandozeile arbeiten. Auf diese Weise kann man etwa die Live-Ausgaben der Robotersteuerung anschauen. Für die Arbeit auf der Kommandozeile sind Linux-Kenntnisse notwendig!

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 TinyCtrl Verzeichnis befindet, dessen Inhalt dem iRC-Verzeichnis in Windows ähnelt.
7. Roboterdateien befinden sich in  
~/TinyCtrl/Data/Robots/<Robotertyp>/<Robotermodell>/<Robotermodell>.xml
8. Die Projektdatei, in der die Roboterdatei referenziert wird, befinden sich in  
~/TinyCtrl/Data/Projects/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 TinyCtrl Log-Output live im Terminal anzuzeigen (das ist speziell nach Änderungen in den Dateien sinnvoll), muss TinyCtrl zunächst beendet werden: killall TinyCtrl
12. Danach navigiert man in das Verzeichnis ~/TinyCtrl und startet ./TinyCtrl
13. Der Prozess lässt sich per Strg+C beenden.
14. Nach einem Neustart der Steuerung startet TinyCtrl automatisch

### 9.2.9 SFTP Zugriff

Der "Secure Shell File Transfer Protocol" (SFTP)-Zugriff funktioniert prinzipiell wie der SSH-Zugriff im Abschnitt oben, ermöglicht dem Benutzer jedoch zu Wartungszwecken Änderungen an den Projekt und Roboterdateien zu machen, ohne ein Terminal benutzen zu müssen.

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)



Die Windows-Software iRC und die Linux-Steuersoftware TinyCtrl benutzen Konfigurationsdateien. Wenn auf einem dieser Systeme manuell (z.B. in einem Texteditor) Änderungen in den Konfigurationsdateien gemacht werden, müssen diese auch auf der anderen Seite durchgeführt werden. Sonst verhalten sich die Steuerungen nicht identisch, es kann zu Kollisionen kommen!

SFTP ist das SSH File Transfer Protocol. Mit ihm lassen sich Dateien von einem Computer auf den anderen übertragen oder anpassen. FileZilla ist ein kostenloses FTP-Programm.

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 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.
- Das Äquivalent zur iRC-Steuersoftware auf der integrierten Steuerung heißt TinyCtrl. Sie liegt im Ordner ~/TinyCtrl
- Die Verzeichnisstruktur innerhalb des TinyCtrl Ordners erinnert stark an die von iRC.

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

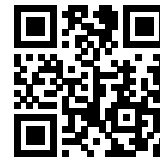


1. Dazu navigiert man in den entsprechenden Ordner und kopiert die Datei per "Drag and Drop" in einen lokalen Ordner.
2. Hier kann die Datei mit einem standard-Texteditor wie z.B. Windows Notepad editiert werden. (Notepad++ ist ein besserer Texteditor <https://notepad-plus-plus.org/>).
3. Sobald alle 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.

### 9.2.10 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>



## 9.3 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" → "Zugriff auf Konfiguration" 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.

## 10 Modbus

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

### 10.1 Lizenzierung

Der Einsatz der Modbus-Funktionen erfordert eine Lizenz. Für eine Dauer von 30 Minuten sind die Funktionen zu Testzwecken frei verwendbar, danach kann ein weiterer Testzeitraum durch einen Neustart der Steuerung gestartet werden. Die Installation einer Lizenzdatei ist in der iRC-Dokumentation beschrieben. Weitere Informationen erhalten Sie unter

<https://shop.cpr-robots.com/?product=modbus-tcp-ip-schnittstelle>



### 10.2 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.

### 10.3 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>



#### 10.3.1 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 zu verwendende 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.

### 10.3.2 Einfügen des Roboterkommunikations FB

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

### 10.3.3 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.

## 10.4 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 21: 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 22: 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 10.4.5.
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.
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 23: 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 1 (siehe Eingaberegister 3).

#### 10.4.1 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
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

Adressen	Datentyp	Beschreibung
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 mit TinyCtrl 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 30)
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
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)
112	Info	Bewegt sich der Roboter?
120	Info	Ist ein Roboterprogramm geladen?
121	Info	Ist ein Logikprogramm geladen?
122	<input type="checkbox"/>	Läuft das Roboterprogramm? / starten / fortführen
123	<input type="checkbox"/>	Ist Roboterprogramm pausiert? / pausieren
124	<input type="checkbox"/>	Ist das Roboterprogramm gestoppt? / stoppen
130	<input type="checkbox"/>	Nächsten Verzeichniseintrag wählen
131	<input type="checkbox"/>	Vorherigen Verzeichniseintrag wählen
132	Info	Ist der gewählte Verzeichniseintrag eine Programmdatei
133	<input type="checkbox"/>	Gewählten Verzeichniseintrag als Roboterprogramm laden / Verzeichnis öffnen
134	<input type="checkbox"/>	Gehe ins Basisverzeichnis (.../Data/Programs)
135	<input type="checkbox"/>	Roboterprogramm entladen

Adressen	Datentyp	Beschreibung
136	┌	Logikprogramm entladen
200-299	boolean	Globale Signale
300-363	boolean	Digitale Ausgänge
364-427	Info	Digitale Eingänge

Tabelle 24: Zuweisung der Coils und der diskreten Eingänge

#### 10.4.2 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 10.4.5 beschrieben.

Adressen	Datentyp	Einheit	Beschreibung
0	uint16		Software-ID (902=iRC, 980=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
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

Adressen	Datentyp	Einheit	Beschreibung
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 (nicht in TinyCtrl)
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
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. 10.4.4)

Tabelle 25: Zuweisung der Eingaberegister

### 10.4.3 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
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. 10.4.4)

Tabelle 26: Zuweisung der Halteregeister

#### 10.4.4 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 27: 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 28: Übersetzungsart



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

#### 10.4.5 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

Wert	Bedeutung
35	Virtuelle Box in Z- berührt
50	NAN in Berechnung
90	Bewegung nicht erlaubt
65535 (0xFFFF)	Unbekannter Fehler

Tabelle 29: 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 30: Betriebsmodus

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

Tabelle 31: Jog-Modus

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

Tabelle 32: Programmzustände (RunState)

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

Tabelle 33: Replay-Modus

---

<b>Wert</b>	<b>Bedeutung</b>
0	Benutzer (Bediengerät, CRI, Modbus, etc.)
1	PLC
2	Programm (Stopp-/Pause-Anweisung)
3	Replay Step (Schrittbetrieb)
4	Shutdown (System fährt herunter)
100	Fehler
101	Pfadgeneratorfehler 1
102	Pfadgeneratorfehler 2
103	Fehler in Zustandsmaschine

---

Tabelle 34: Grund für letzten Stopp/Pause des Programms

## **11** **Wartung**

### **11.1** **Reinigung**

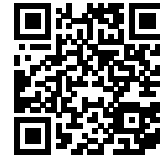
- Nachdem die Steuerung vom Stromnetz getrennt ist, kann sie mit einem feuchten Tuch abgewischt werden.
- Nachdem die Steuerung vom Stromnetz getrennt ist, können Lüfter oder Lüftungsschlitze mit einem feuchten Tuch oder leichter Druckluft vorsichtig gereinigt werden. Dabei müssen die Rotoren der Lüfter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

## 12 Fehlerbehebung

### 12.1 Häufig gestellte Fragen

Antworten zu häufig gestellten Fragen finden Sie in unserem Wiki:

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



### 12.2 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>



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 Softwareprobleme sind in folgendem Artikel beschrieben:

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



### 12.3 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 8.3).

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)



## 12.4 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