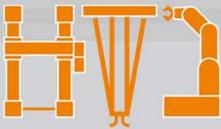


igus[®]



**LOW
COST
AUTOMATION**
by igus[®]



**Benutzerhandbuch
drylin[®] Delta-Roboter
mit iRC Steuerung**

Version V1.1-DE, September 2020

© Commonplace Robotics GmbH, 2011 - 2020

igus®, drylin® und robolink® sind eingetragene Warenzeichen der igus GmbH.

igus® GmbH
Spicher Str. 1a
51147 Köln
Deutschland
Telefonische Unterstützung: +49 (0) 2203-96498-255
ww-robot-control@igus.net
www.igus.de, www.igus.eu

Inhaltsverzeichnis

1.	Sicherheitshinweise.....	5
2.	Schnellstart-Anleitung.....	6
2.1.	Einrichten und Anschließen.....	6
2.2.	Einschalten.....	7
2.3.	Verbindung mit einem Window PC.....	7
3.	Einführung.....	9
3.1.	Übersicht über das System.....	9
3.2.	Glossary and Abbreviations.....	10
3.3.	Spezifikationen.....	12
3.4.	Mechanische Dimensionen.....	13
4.	Elektrische Verbindungen	14
4.1.	Übersicht.....	14
4.2.	Pinbelegung: Schrittmotormodul.....	15
4.3.	Pinbelegung: Supportmodul.....	16
4.4.	Pinbelegung: Digitales Ein-/Ausgabe-Modul.....	17
4.5.	Sensoren und Aktoren an das DIO-Modul anschließen	18
4.6.	Option: Schaltschrank.....	20
4.7.	Option: Integrierter Computer	21
4.8.	Option: Handbediengerät.....	24
5.	Sicherheit	25
5.1.	Sicherheitsmerkmale der modularen Robotersteuerung.....	25
5.2.	CE-Zertifizierung.....	25
5.3.	Integration von SIL-bewerteten Sicherheitskomponenten	25
6.	Software-Installation.....	27
6.1.	Installation der iRC - igus® Robot Control.....	27
6.2.	Lizenzierung.....	29
6.3.	Einrichten der Ethernet-Verbindung zum integrierten Computer.....	29
6.4.	Installieren des CAN-to-USB-Treibers.....	30
7.	Bewegen des Roboters mit iRC.....	31
7.1.	Die grafische Benutzeroberfläche des iRC	31
7.2.	Anschließen des Roboters.....	34
7.3.	Referenzierung des Roboters	35
7.4.	Bewegen des Roboters mit Gamepad oder Software-Tasten.....	36
7.5.	Starten von Roboterprogrammen	37
7.6.	Digitale Ein- und Ausgänge.....	39

7.7.	Software-Schnittstellen.....	39
7.8.	Aktualisieren der Software.....	40
8.	Programmierung des Roboters mit iRC.....	41
8.1.	Der Programm-Editor	41
8.2.	Kommentare und Informationen in Programmen.....	44
8.3.	Variablen und Variablenzugriff.....	45
8.4.	Ablauf der Programmausführung.....	49
8.5.	Bewegung.....	57
8.6.	Greifer und digitale Ein- / Ausgänge	63
8.7.	Kamera.....	64
9.	Stand-alone-Betrieb mit integriertem Computer und dem Handbediengerät... 65	
9.1.	Fehler zurücksetzen/Roboter aktivieren.....	66
9.2.	Bewegen des Roboters mit dem 3-Achsen-Joystick.....	66
9.3.	Referenzierung	67
9.4.	Starten und Stoppen eines Programms.....	68
9.5.	Manuelle Einstellung der digitalen Ein-/Ausgänge.....	68
9.6.	Anzeige von Statusinformationen	68
9.7.	Programme auf dem integrierten Computer organisieren	69
10.	Projekt-Konfiguration.....	70
10.1.	Programm	70
10.2.	Werkzeug.....	70
10.3.	Eingänge / Ausgänge	71
10.4.	Virtuelle Box.....	72
11.	Robot Configuration.....	73
12.	Konfiguration der Schnittstellen.....	74
12.1.	Kameras.....	74
12.2.	PLC-Schnittstelle.....	74
12.3.	CRI-Ethernet-Schnittstelle	75
13.	Fehlerbehebung.....	76
13.1.	Support Kontakt.....	76
13.2.	Online-Tool - Fehleridentifizierung und -behebung.....	76
13.3.	Konfiguration der Stepper-Module.....	77
13.4.	Kalibrierung des Roboters.....	78
13.5.	Fehler-Codes.....	79

1. Sicherheitshinweise



Bedienen Sie den Roboter vorsichtig!

Achten Sie bei der Bedienung eines Deltaroboters 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 "CE-Kennzeichnung" unten und Abschnitt 5.
- CE-Kennzeichnung: Der Deltaroboter 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, das je nach Konfiguration selbst Netzspannung (120 / 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 / 240 V), wenn Sie im Schaltschrank oder an der Elektronik arbeiten, die an die Robotersteuerung angeschlossen ist.
- KEIN Hot-Plugging! Dies kann zu dauerhaften Schäden an den Motormodulen führen. Installieren oder entfernen Sie keine Module oder Steckverbinder (z.B. Handbediengerät, Not-Aus-Schalter, DIO-Module oder externe Relais, Motoranschlüsse...), während Sie eingeschaltet sind.
- Der Deltaroboter 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 igus® Robotersteuerungssoftware.

2. Schnellstart-Anleitung

2.1. Einrichten und Anschließen

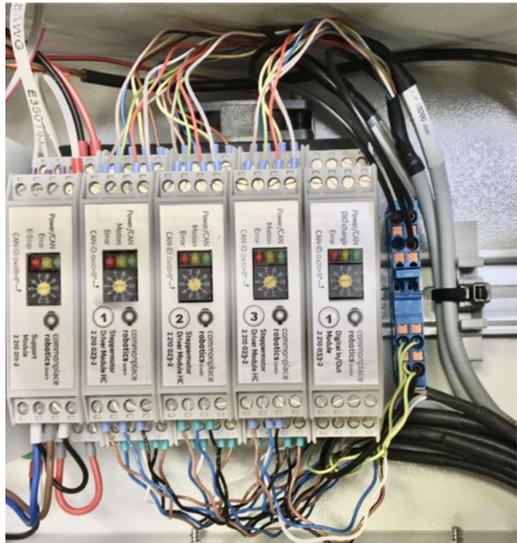
- Befolgen Sie die Sicherheitsanweisungen in Abschnitt 1.
- Vergewissern Sie sich, dass der Ein-/Aus-Schalter am Schaltschrank auf "Aus" steht.
- Montieren Sie den Deltaroboter auf einer geeigneten Fläche. Stellen Sie sicher, dass die Kabel nicht unter Spannung stehen.
- Führen Sie die Roboterkabel durch das große kreisförmige Loch im Schaltschrank und stecken Sie sie an die Schrittmotormodule. Jeder Motor ist über 4 Steckverbinder mit seinem Schrittmotormodul verbunden.

Alle Anschlüsse sind beschriftet und kodiert, um diesen Prozess zu unterstützen (siehe Abschnitt 4):

- Motorkabel
(mit der Bezeichnung *Motor*)
- Encoderkabel
(2 Anschlüsse mit der Bezeichnung *ENC-1* und *ENC-2*)
- Referenzsensor
(mit der Bezeichnung *End-Stop*)

Verbinden Sie auch die Abschirmung und die Erdung der Motor- und Encoderkabel mit GND:

- Motor Erdung: gelb-grün
- Encoder Abschirmung: schwarz



Schließen Sie diese 6 Drähte im Erdungsanschlussblock auf der rechten Seite der Elektronikmodule an.

- Sichern Sie die Roboterkabel gegen Spannung, z.B. mit einem Kabelbinder an einem der Löcher im Schaltschrank. Falls vorhanden, stecken Sie das Displaykabel ein und sichern Sie es über die Schraubverbindung.
- Danach kann der Roboter an das Stromnetz angeschlossen werden.

2.2. Einschalten

- Schalten Sie den Roboter mit dem Ein/Aus-Schalter am Schaltschrank ein.
- Die grünen Leuchtdioden (LEDs) auf den Modulen leuchten nun, ebenso die meisten roten LEDs und möglicherweise auch einige der gelben LEDs.
- Optional: Bei Auslieferung mit einem integrierten Computer beginnen die grünen LEDs nach ca. 20 s zu blinken. Dies zeigt die Kommunikation an, nun ist die Robotersteuerung in Betrieb. Falls verfügbar, können Sie den Roboter jetzt über das Handbediengerät bewegen (optional). Details finden Sie in Abschnitt 8.



2.3. Verbindung mit einem Window PC

- 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.
- Stellen Sie die IP-Adresse des PCs auf: statisch und 192.168.3.1 mit einer Subnetzmaske von 255.255.255.0



2.3.1. Verbinden und Bewegen des Roboters

- Installieren Sie die igus® Robot Control Software auf Ihrem PC, siehe Abschnitt 6.1.
- Starten Sie die igus® Robot Control Software. Beim Start können Sie das auf Ihren Roboter zutreffende Projekt auswählen. Bitte beziehen Sie sich auf die igus-Produktnummer, die Projektnamen basieren auf diesen.
- Sie können den Roboter jetzt aktivieren, indem Sie
 - "Verbinden",
 - "Zurücksetzen" und



3. Einführung

3.1. Übersicht über das System

Das Robotersystem besteht aus vier Grundkomponenten:

1. **Deltaroboter** : die mechanischen Linearachsen mit Motoren und Schaltern
2. **Robotersteuerung** : modulare Motorsteuerung und DIO-Module
3. **Integrierter Computer**: In den Schaltschrank integrierter Klein-Computer mit Steuerungssoftware zur Ausführung von Roboterprogrammen
4. **Programmierungsumgebung**: grafische Software zum Einrichten von Roboterprogrammen, zur Ausführung auf Windows-Computern.

Die Steuerung verwendet einen integrierten Computer mit Robotersteuerungssoftware, um die modulare Robotersteuerung anzusteuern. Diese Konfiguration kann Roboterprogramme ausführen, ohne dass ein externer Windows-PC angeschlossen ist. Zur Programmierung kann ein externer PC über Ethernet angeschlossen werden.

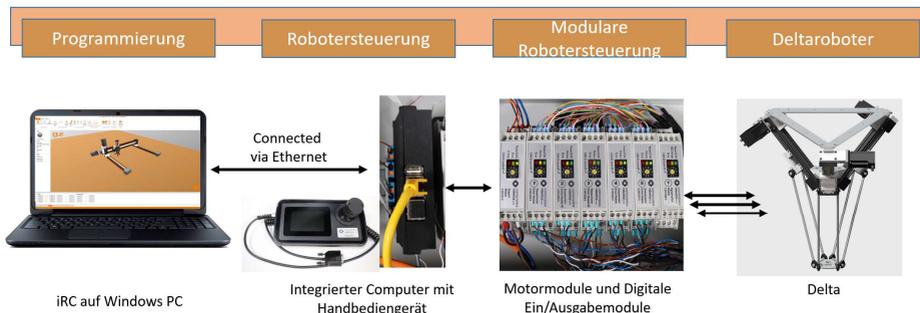
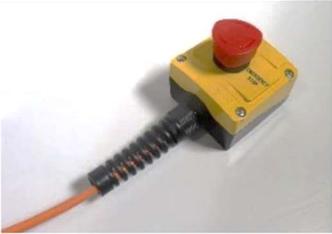


Abb. 3.1: Deltaroboter mit integriertem Computer, auf dem die Robotersteuerungssoftware läuft.

In Ausbildungsumgebungen kann die Robotersteuerung über einen USB-to-CAN-Adapter direkt vom Windows-PC aus gesteuert werden. Dieser Aufbau wird hier nicht behandelt.

3.2. Glossary and Abbreviations

Bild	Name	Abkürzung	Beschreibung
	igus® Delta-Roboter	Delta-Roboter	Mechanischer Deltaroboter einschließlich Struktur, Motoren und Kabel
	Modulare Robotersteuerung	Robotersteuerung	Besteht aus: <ul style="list-style-type: none"> • 1 Support-Modul • 3 oder mehr Stepper-Module • 1-3 DIO-Module
	Schaltschrank	-	Stahlschrank, in dem die Robotersteuerung montiert werden kann. Enthält eine 24V / 10A Stromversorgung.
	integrierter Computer	-	Optionaler integrierter Computer mit Linux-Betriebssystem und der TinyCtrl-Robotersteuerungssoftware
	Handbediengerät	-	Optionales Handbediengerät mit kapazitivem 3,5"-Touch-screen-Display und 3-Achsen-Joystick

	<p>Not-Aus-Schalter</p>	<p>-</p>	<p>Einkanaliger Notstopp</p>
<p>Elektronische Steuerungsmodule</p>			
	<p>Digitales Ein-/Ausgabe-Modul</p>	<p>DIO Modul</p>	<p>Liest 7 digitale Eingänge auf 24V-Pegel. Setzt 7 digitale Ausgangskanäle auf der Basis von Solid State Relais.</p>
	<p>Schrittmotor-modul</p>	<p>Stepper-Modul</p>	<p>Versionen:</p> <ul style="list-style-type: none"> • Motor-Encoder (ME): - Hochstrom (HC) - Schwachstrom (LC) • Ausgabe-Encoder (AE): - Hochstrom (HC) - Schwachstrom (LC)
	<p>Versorgungs-Modul</p>	<p>-</p>	<p>Liefert 5V Logikspannung, max. 2 A. Schnittstelle für einkanalige Not-Aus-Taste. Schnittstelle für zusätzliche z.B. zweikanalige Sicherheitsrelais.</p>

3.3. Spezifikationen

Deltaroboter	
Typ	igus® drylin® Delta-Roboter
Anzahl der Achsen	Je nach Ausführung
Nutzlast	Je nach Ausführung
Modulare Robotersteuerung	
Stromversorgung	24 V bei 10 A
Typ	Hutschienenmodule im ME-Format mit 5-poligem Busverbinder
Kommunikation	Controller Area Network (CAN) Feldbus 500 kBit USB-zu-CAN-Adapter PCAN-USB von Peak Systems Ethernet
Support-Module	Bereitstellung von 5 V Logikspannung: max. 2 A. SoftStart, um eine Überlastung der Stromversorgung zu verhindern. 1-kanalige Not-Aus-Funktion ohne Sicherheitsklassifizierung Vorkehrungen zum Anschluss externer Sicherheitsrelais.
Schrittmotor-Module	Zum Betrieb eines bipolaren Schrittmotors mit Quadratur-Encoder RS422 24 V Referenzschalter-Eingang
Digitale Ein-/Ausgangsmodule (DIO Module)	7 digitale Eingänge, 12 - 24 V, optogekoppelt 7 digitale Ausgänge, Halbleiterrelais, max. 500 mA
Integrierter Computer	
Plattform	Phytec Regor oder vergleichbar, CPU z.B. Texas Instruments AM3352
Betriebssystem	Linux
Software	TinyCtrl-Robotersteuerungs-Software
Schnittstellen	Steuerung der Antriebe und DIO-Module über den CAN-Bus Verbindung zu igus® Robot Control über Ethernet RS232-Display-Anschluss
Software für Programmierung und Robotersteuerung	
igus® Robot Control	
Empfohlene Systemanforderungen	PC mit z. B. Intel i5-CPU (mindestens i3) und Windows 10, freier USB 2.0-Anschluss, Ethernet-Anschluss, 500 MB Festplattenspeicher

3.4. Mechanische Dimensionen

Weitere Informationen finden Sie in der igus® Dokumentation des Delta-Roboters.

4. Elektrische Verbindungen

4.1. Übersicht

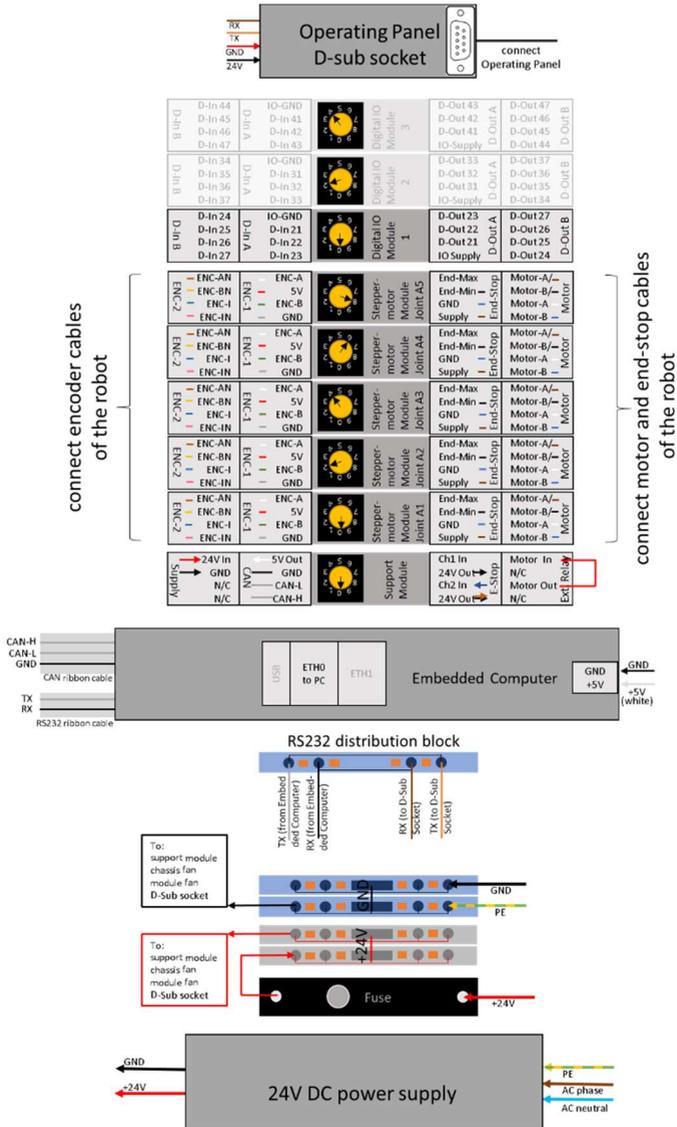


Abb. 4.1 Übersicht über die elektrischen Anschlüsse. Die Anzahl der Motortreiber hängt von der Anzahl der Gelenke des Deltaroboters ab.

4.2. Pinbelegung: Schrittmotormodul

Jedes Schrittmotormodul treibt einen bipolaren Schrittmotor mit Motor-Encodern an. Die Encodersignale werden von einem RS422-Baustein ausgewertet.

Die Signale für jede Achse laufen über drei Kabel: Motorkabel, Encoderkabel und Referenzschalterkabel.

Das Motorkabel wird an einen Stecker mit der Bezeichnung „Motor“, das Encoderkabel an zwei Stecker mit der Bezeichnung „ENC-1“ und „ENC-2“ und das Referenzschalterkabel an den Stecker mit der Bezeichnung „End-Stop“ angeschlossen.



Motorstecker:

Verbindet einen bipolaren Schrittmotor.

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



Encoderstecker 1 (ENC-1):

Verbindet einen Quadratur-Encoder mit einem RS422-Baustein.

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

Encoderstecker 2 (ENC-2):

Verbindet einen Quadratur-Encoder mit einem RS422-Baustein.

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

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



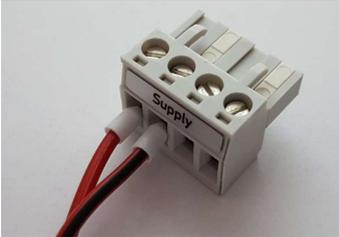
End-Stop-Stecker:

Wird an einen End-Stop- oder Referenzschalter angeschlossen.

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

4.3. Pinbelegung: Supportmodul

Das Supportmodul stellt 5 V Logikspannung, ein einkanaliges Not-Aus-Relais und ein SoftStart-Relais zur Verfügung. Es speist die Signale in das DIN-Schienen-Bussystem ein.



Versorgungsspannungs -Stecker:

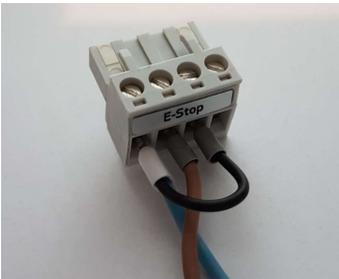
Pin 1 (links):	rot	24 V
Pin 2:	schwarz	GND
Pin 3:	nicht verbinden	-
Pin 4:	nicht verbinden	-



Controller Area Network (CAN) -Stecker:

Verbindet den Controller und die CAN-Schnittstelle des integrierten Computers.

Pin 1 (links):	5 V an integriertem PC
Pin 2:	schwarz CAN-GND
Pin 3:	orange CAN-L
Pin 4:	rot CAN-H



Anschluss für Not -Aus (E-Stop):

Verbindet den Not-Aus-Knopf.

Pin 1 (links):	blau	Not-Aus-Kanal 1
Pin 2:	braun	24 V Ausgangssignal
Pin 3:	schwarz	E-Stop-Kanal 2
Stift 4:	nicht verbinden	-

Dies ist ein einkanaliger Aufbau. Passen Sie diesen an Ihre Sicherheitsanforderungen an! Die Möglichkeiten zum Anschluss eines Sicherheitsrelais werden unten und in Abschnitt 5 beschrieben.



Brücke für Motorleistung:

Ermöglicht die Unterbrechung des Motorstroms durch externe Sicherheitsschalter, siehe Abschnitt 5.

Pin 1 (links):	Motorleistungs-Ausgang
Pin 2:	nicht anschließen
Pin 3:	Motorleistungs-Eingang
Pin 4:	nicht verbinden

4.4. Pinbelegung: Digitales Ein-/Ausgabe-Modul

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

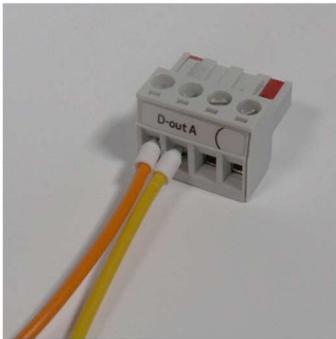


Ein von den Ausgangsrelais geschalteter Stromkreis darf keine größeren Kondensatoren enthalten.

Wenn der Strom 500 mA übersteigt, können die Halbleiterrelais beschädigt werden.

Aus Sicherheitsgründen sind die Ein- und Ausgänge des DIO-Moduls galvanisch getrennt. Das bedeutet, dass die Schaltkreise der Ein- und Ausgänge nicht mit den internen Schaltkreisen des Controllers verbunden sind.

Es ist daher notwendig, eine Versorgungsspannung für die Ausgänge und eine Masseleitung für die Eingänge anzuschließen. Hierfür können die in der Robotersteuerung verfügbaren 24 V, aber auch eine externe unabhängige Spannungsquelle verwendet werden. In der Software sind die Ein- und Ausgänge des ersten DIO-Moduls mit den Nummern 21-27, des zweiten DIO-Moduls (falls mitgeliefert) mit den Nummern 31-37 und des dritten mit 41-47 nummeriert.



Digital Out-Stecker A (D-out A):

Die Ausgangsrelais verbinden den Pin der Stromversorgung mit den entsprechenden Ausgangspins.

Pin 1 (links): Eingangsspannung (für alle 7 Kanäle)

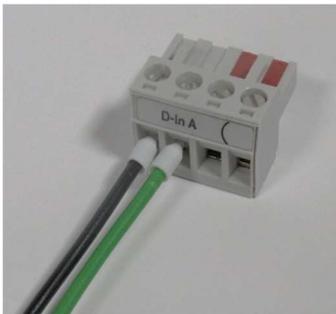
Pin 2: D-Out-Kanal 1 (in der Software: Dout21)

Pin 3: D-Out-Kanal 2 (in der Software: Dout22)

Pin 4: D-Out-Kanal 3 (in der Software: Dout23)

Digital Out-Stecker B (D-out B):

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



Digital In-Stecker A (D-in A):

Pin 1 von D-In A ist der entsprechende GND-Pin für alle Eingangspins.

Pin 1 (links): Signal GND (für alle 7 Kanäle)

Pin 2: D-In Kanal 1 (in der Software Din21)

Pin 3: D-In-Kanal 2 (in der Software Din22)

Pin 4: D-In-Kanal 3 (in der Software Din23)

Digital In-Stecker B (D-in B):

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

4.5. Sensoren und Aktoren an das DIO-Modul anschließen

Der einfachste Weg, eine speicherprogrammierbare Steuerung (SPS) anzuschließen, ist über digitale Ein- und Ausgänge. Jede Robotersteuerung wird mit einem DIO-Modul geliefert. Dieses bietet 7 Eingänge und 7 Ausgänge (siehe Abschnitt 4.2). Wenn zusätzliche Ein- und Ausgänge erforderlich sind, können bis zu zwei zusätzliche DIO-Module integriert werden, siehe Abschnitt 10.1. Insgesamt können bis zu 3 DIO-Module angesteuert werden.

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.

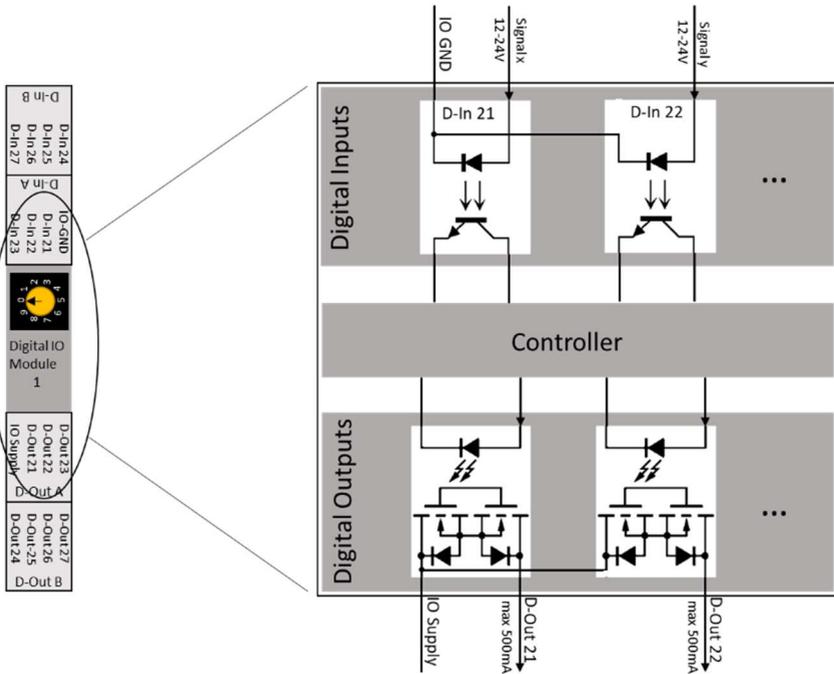


Abb. 4.2: Interner Aufbau eines DIO-Moduls

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

4.5.1.1. Einen Sensor anschließen

- Pin 1 (GND) des D-in-1-Anschlusses muss mit GND der Stromversorgung des Sensors verbunden werden.
- Das Sensorsignal (positiv) muss an einen Eingangspin D-in 1 Stecker Pins 2-4 oder D-in 2 Stecker Pins 1-4 angeschlossen werden. Die positive Seite des Sensors muss an V_{dd} der Stromversorgung angeschlossen werden.
- Der Status der Eingänge kann im Register „Input/Output“ unten in iRC überwacht werden, siehe Abschnitt 7.1
- In einem Roboterprogramm können Eingaben abgefragt und darauf reagiert werden, z.B. in einem if-then-else-Befehl, siehe Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** .

4.5.1.2. Einen Aktor anschließen (LED, Pneumatikventil, Relais...)

- Pin 1 (Versorgungsspannung) des Steckers D-out 1 muss mit einer Stromversorgung (z.B. 24V) verbunden werden.
- Der Aktor (Relais usw.) wird dann über einen freien Pin der D-out-Stecker (D-out 1 Pin 2-4 und D-out 2 Pin 1-4) mit Strom versorgt.
- Sie können die Ausgaben manuell im Register "Eingabe / Ausgabe" unten in iRC einstellen, siehe Abschnitt 7.1
- In einem Roboterprogramm können Sie den Zustand der Ausgänge mit dem Digital-Out-Befehl einstellen, siehe Abschnitt 8.6.

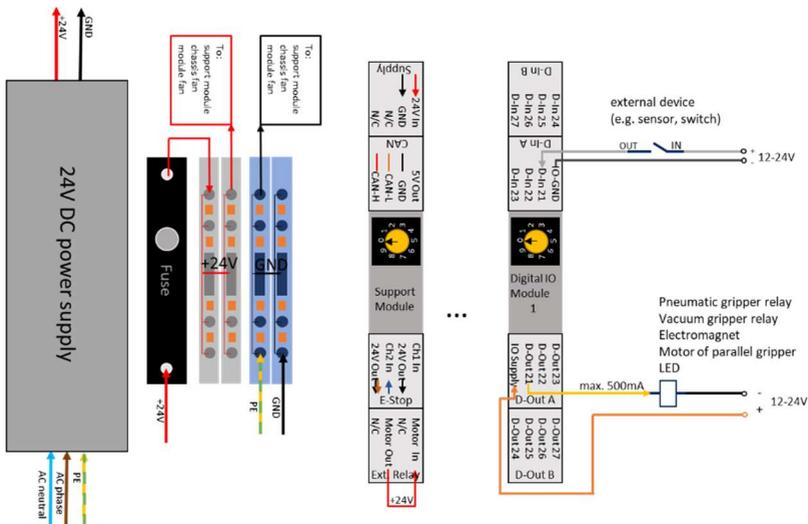


Abb. 4.3: Anschluss eines Ein- und Ausgangs an das DIO-Modul

4.6. Option: Schaltschrank

Die modulare Robotersteuerung kann in einem Schaltschrank bestellt werden. Der Schrank schirmt die Steuerung vor Staub, Feuchtigkeit und versehentlichem Zugriff ab.

Die Abmessungen des Schaltschranks sind B x L x H: 600 x 200 x 125 mm.



Abb. 4.4: Schaltschrank, geschlossen. Links unten sind der IEC-Netzanschluss / Schalter / Sicherung und die Kabelanschlüsse zu sehen.

Der Schrank enthält ein 24V / 10A-Netzteil für 220V / 110V-Eingang sowie eine Sicherung (8A). Verteilerblöcke sind im Lieferumfang enthalten. Ein kleiner Lüfter ist auf der linken Seite des Schrankes montiert.

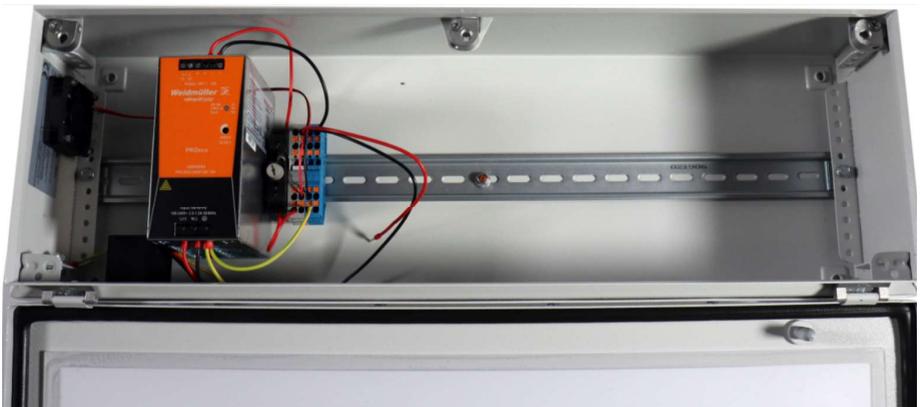


Abb. 4.5: Schaltschrank mit Stromversorgung, ohne modulare Steuerelektronik.

4.7. Option: Integrierter Computer

Ein integrierter Computer kann mit einer modularen Robotersteuerung kombiniert werden, so dass ein externer Computer nur für die Programmierung, nicht aber für den täglichen Betrieb erforderlich ist. Dies ähnelt der „Konfiguration A“ in der Einleitung, Abschnitt 3.1.

Der integrierte Computer ist ein Einplatinencomputer mit Linux-Betriebssystem, auf dem die TinyCtrl-Robotersteuerungssoftware läuft. Er ermöglicht folgendes:

- Bedienen Sie den Roboterarm
- Spielen Sie vorher erstellte Programme ab
- Verbinden Sie sich mit iRC auf einem Windows-Computer über Ethernet zur Programmierung

Der integrierte Computer kommuniziert mit der modularen Steuerelektronik auf den DIN-Schienen.



Abb. 4.6 Integrierter Computer mit Ethernet-Kabel, das an die ETH0-Buchse angeschlossen ist.

4.7.1. Ethernet-Verbindung

ETH0 Primärer Ethernet- Anschluss (neben dem USB-Anschluss)	Verbindung zum PC zur Programmierung über igus@ Robot Control	Standard-Ethernet IP 192.168.3.11
ETH1 Sekundärer Ethernet- Anschluss	Normalerweise ungenutzt. Kann zum Anschluss einer Kamera verwendet werden	Standard-Ethernet IP 192.168.4.11

4.7.2. Anschluss des Bedienfelds

Ein Bedienfeld kann angeschlossen werden, siehe Abschnitt 4.8

RS232-Verbindung	RS232-Verbindung zum Handbediengerät	Leitung 1: UART2-TX Leitung 2: UART-2-RX
------------------	--------------------------------------	---



Das 2-adrige Flachbandkabel des integrierten Computers ist der RS232-Anschluss für das Handbediengerät. Eine Ader ist schwarz markiert (Ader 1).

Er wird über einen Verteilerblock an einen 9-poligen D-Sub-Stecker angeschlossen.

Ader 1 (schwarz) D-Sub-Anschluss Orange

Ader 2 D-Sub-Anschluss Braun

Um das Display mit Strom zu versorgen, werden die schwarze und die rote Ader des D-Sub-Steckers mit GND bzw. 24 V verbunden.

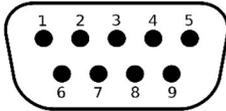


Abb. 4.7: D-Sub-Anschluss für das Handbediengerät

Die Pinbelegung des Steckers ist wie folgt:



Vorsicht! Die Verbindung ist proprietär.
Sie ist nicht für ein Nullmodemkabel oder ähnliches geeignet.



Pin 1: 24V
Pin 3: RS232-TX
Pin 6: GND
Pin 7: RS232-RX
Alle übrigen Pins sind nicht angeschlossen

4.7.3. CAN-Verbindung

CAN-Flachbandkabel (drei Adern)	Stellt die CAN-Verbindung zu den Steppermodulen her. Wird mit dem Support-Modul verbunden.	Ader 1: GND (schwarz) Ader 2: CAN-L Ader 3: CAN-H
------------------------------------	---	---



Das 3-adrige Flachbandkabel des integrierten Computers ist der CAN-Anschluss. Es hat eine schwarz markierte Ader (Ader 1, GND). Die Adern 2 und 3 sind CAN-L und CAN-H.

Die Adern werden wie folgt mit dem Supportmodul verbunden:

Ader 1 (schwarz)	Supportmodul CAN-Anschluss Pin 2
Ader 2	Supportmodul CAN-Anschluss Pin 3
Ader 3	Supportmodul CAN-Anschluss Pin 4

4.7.4. Stromversorgung

Stromversorgung 5V	Versorgt das Linux-Board mit 5V DC	Pin 1: 5V Pin 2: GND
--------------------	------------------------------------	-------------------------



Der Steckverbinder hat eine weiße (5 V) und eine schwarze (GND) Ader. Wenn das Modul auf der DIN-Schiene montiert ist, ist die weiße (5 V) zur Unterseite des Gehäuses hin ausgerichtet.

Das 5-V-Kabel (weiß) wird mit dem 5-V-Anschluss des Support-Moduls (Pin 1 des CAN-Anschlusses) verbunden. Das schwarze Kabel wird an die gemeinsamen GND-Verteilerblöcke der DIN-Schiene angeschlossen.

Überprüfen Sie vor dem Einschalten die Polarität der Verbindung!

4.8. Option: Handbediengerät

Das Handbediengerät wird zur Steuerung des Roboters über den integrierten Computer verwendet. Für die Bedienungsanleitung siehe Abschnitt 9.

Der Anschluss des D-Sub-9-Steckers ist mit der D-Sub-9-Buchse des Schaltschranks Abschnitt 4.7.2 zu verbinden.



Abb. 4.8: Handbediengerät mit Touchscreen und 3-Achsen-Joystick.

5. Sicherheit

5.1. Sicherheitsmerkmale der modularen Robotersteuerung

Die modularen Steuerelektroniken verfügen über Möglichkeiten zum Anschluss externer Sicherheitskomponenten. Es können externe Sicherheitsrelais/-schalter zur Unterbrechung der Motorleistung angeschlossen werden. Der Not-Aus-Schalter ist ein 1-kanaliges Gerät und entspricht keinem SIL-Level.

5.2. CE-Zertifizierung

Die Komponenten dieses Robotersystems sind mit der CE-Kennzeichnung gemäß der europäischen Maschinenrichtlinie versehen. Verantwortlich für die Komponenten sind:

- igus GmbH: Delta-Roboter
- Commonplace Robotics GmbH: Modulare Robotersteuerung

Diese CE-Kennzeichnungen bestätigen die Einhaltung der erforderlichen Normen durch diese Teile der Endmaschine. Die Endmaschine, d.h. die komplette Roboterzelle, ist nicht durch diese CE-Kennzeichnungen abgedeckt.

5.3. Integration von SIL-bewerteten Sicherheitskomponenten

Die modulare Robotersteuerung bietet keine sicherheitsrelevanten Funktionen. Die integrierte Not-Aus-Funktion ist einkanalig.



Um das komplette, kundenspezifische Robotersystem zu betreiben, muss der inbetriebnehmende Ingenieur im Rahmen der CE-Zertifizierung eine Risikobeurteilung durchführen und je nach Ergebnis zusätzliche Sicherheitskomponenten integrieren. Dies sind in der Regel Sicherheitsrelais, Umhausungen und Türschalter.

Sicherheitsrelais können in Reihe mit dem Motorleistungs-Brückenstecker (Abschnitt 4.3) des Supportmoduls (Abb. 5.1) geschaltet werden. Der Steckverbinder ist mit „Ext Rel“ gekennzeichnet. Über diesen Steckverbinder wird die Motorversorgung aus dem Modul heraus und über eine Steckbrücke zurück in das Modul und dann in den Bus geführt. Wenn diese Verbindung unterbrochen wird, ist keine aktive Bewegung der Motoren möglich.

Dies ermöglicht die Implementierung von Sicherheitskomponenten mit dem von der Anwendung geforderten Sicherheitsintegritätslevel (SIL).

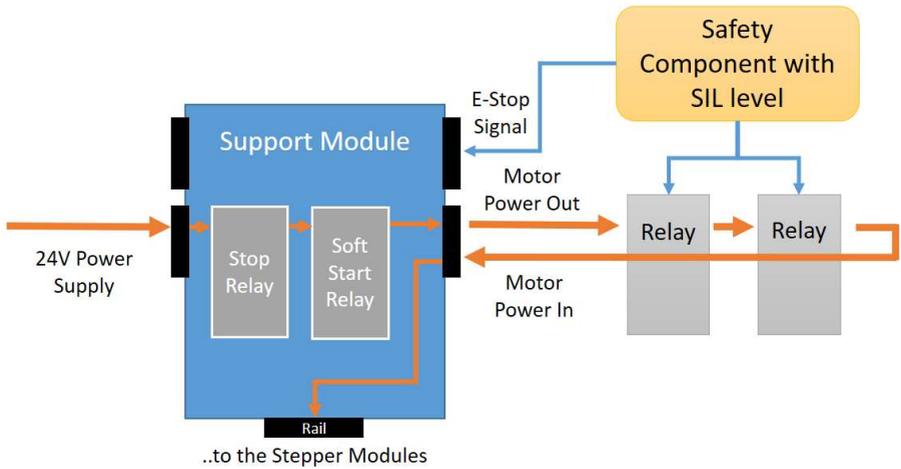


Abb. 5.1 Schematische Verdrahtung der Sicherheitskomponenten mit dem Support-Modul.

6. Software-Installation

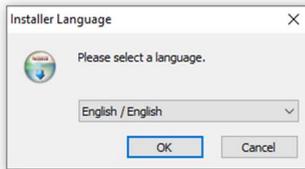


- Sichern Sie wichtige Daten vor der Installation der igus® Robot Control.
- Bevor Sie die igus® Robot Control Software aktualisieren, erstellen Sie ein Backup der aktuellen Version, z.B. durch Umbenennen des Ordners C:\iRC-igusRobotControl\ in C:\iRC-igusRobotControl_BAK\.

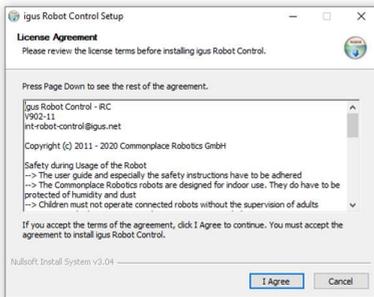
6.1. Installation der iRC - igus® Robot Control



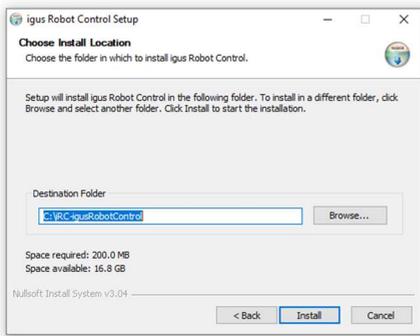
Stecken Sie den igus® Robot Control USB-Speicherstick ein. Starten Sie das Installationsprogramm: z. B. D:\Installer_iRC_V11-nnn.exe
Möglicherweise müssen Sie Änderungen an Ihrem System zulassen.



Nach dem Start der Installation können Sie zwischen deutscher und englischer Sprache wählen.

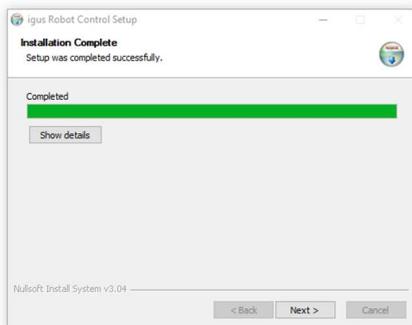


Bestätigen Sie dann die Lizenzvereinbarung.



Im nächsten Schritt können Sie wählen, wo die igus® Robot Control Software installiert werden soll. Das empfohlene Verzeichnis ist C:\IRC-igusRobotControl

Bei der Installation von igus® Robot Control in einem Windows-Programmverzeichnis wie C:\Programme ist es möglich, dass igus® Robot Control nur als Administrator gestartet werden kann.



Die Installation dauert in der Regel nur wenige Sekunden.



Nach Abschluss der Installation können Sie die igus® Robot Control Software starten.

Jetzt können Sie igus® Robot Control über die Verknüpfung auf dem Desktop oder das Startmenü starten.

Installationsfehler:

Der Installationsassistent prüft, ob alle erforderlichen Erweiterungen, insbesondere das .NET-Framework, verfügbar sind. Wenn dies nicht der Fall ist, erscheint eine Fehlermeldung. Das .NET-Framework muss manuell installiert werden:

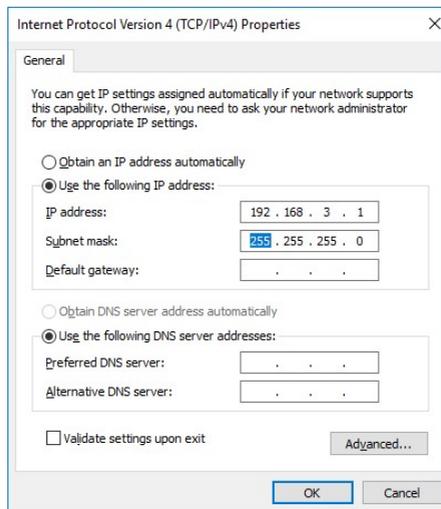
- Suchen Sie im Internet nach „Microsoft .NET Download“ und installieren Sie es.

6.2. Lizenzierung

- Die igus® Robot Control Software erfordert einen Lizenzschlüssel, um zu starten.
- Dieser Schlüssel wird bei der Installation von igus® Robot Control automatisch erstellt.
- Bitte ändern Sie den Inhalt der Lizenzdatei nicht, sonst wird sie ungültig!
- Die mitgelieferte Standardlizenz erlaubt die Installation und Nutzung von igus® Robot Control auf beliebig vielen Rechnern im Unternehmen oder der Organisation des Lizenznehmers.

6.3. Einrichten der Ethernet-Verbindung zum integrierten Computer

- Schließen Sie den Ethernet-Port (ETH0, neben der USB-Buchse) des Roboters mit einem Standard-LAN-Kabel an einen Windows-PC an.
- Setzen Sie die IP des PCs auf 192.168.3.1 (der Roboter hat die IP 192.168.3.11). Weitere Hilfe zum Ändern der IP-Adresse finden Sie online.



Ab diesem Zeitpunkt kann die igus® Robot Control Programmierumgebung für den Betrieb des angeschlossenen Roboters genutzt werden.

6.4. Installieren des CAN-to-USB-Treibers

Wenn kein integrierter Computer bestellt wurde, wird ein CAN-zu-USB-Adapter geliefert, der die modulare Robotersteuerung mit einem Windows-Computer verbindet. Es wird der entsprechende Treiber benötigt. Die Robotersteuerungssoftware wird mit dem PCAN-USB-Treiber von www.peak-system.com geliefert, der sich auch auf dem USB-Speicherstick der igus® Robot Control-Installation (Verzeichnis PCAN-USB-Adapter) oder auf der Installations-CD des Herstellers befindet.

Nach dem Start der Installation müssen Sie:

- die Lizenzvereinbarung akzeptieren,
und
- den Installationsordner angeben.

Prüfen Sie im nächsten Schritt, ob das PCAN-USB-Gerät und der PCAN-View CAN-Bus Monitor wie unten dargestellt zur Installation ausgewählt sind.



Wählen Sie beides für die Installation:

- PCAN-USB
- PCAN-View

Mit dem PCAN-View CAN-Bus Monitor können Sie überprüfen, ob der Adapter korrekt angeschlossen ist. Im täglichen Betrieb wird er jedoch nicht benötigt.

7. Bewegen des Roboters mit 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.

Diese Betriebsanleitung wird durch die jeweilige roboterspezifische Betriebsanleitung ergänzt.

7.1. Die grafische Benutzeroberfläche des iRC

Dieser Abschnitt erklärt die igus® Robot Control Software. Auch ohne einen angeschlossenen Roboter können alle Schritte simuliert werden. In Abschnitt 7.2 wird dann der reale Roboter angeschlossen und bewegt.

Die Programmierumgebung igus® Robot Control ermöglicht die Steuerung und Programmierung des Roboters. igus® Robot Control ist eine Windows-Software. Sie können sowohl online als auch offline arbeiten, d.h. mit dem Roboter oder in der Simulation (bei ausgeschaltetem oder nicht angeschlossenem Roboter).

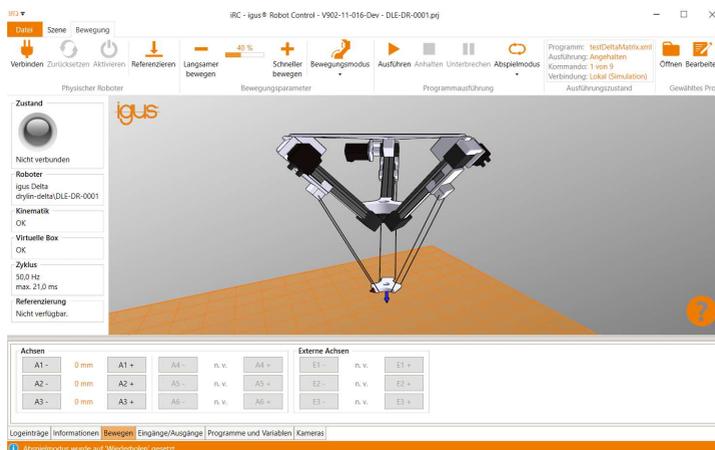


Abb. 7.1 Benutzeroberfläche der igus® Robot Control.

In der linken oberen Ecke bieten die drei Registerkarten „Datei“, „Szene“ und „Bewegung“ 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“ (Abb. 7.1 Benutzeroberfläche der).

Es gibt fünf Registerkarten am unteren Rand des Fensters:

- „Log-Meldungen“: Meldungen des Programms über Status oder Fehler.
- „Infocenter“: zeigt die Achswerte, die kartesische Position und weitere Informationen an.
- „Jogging“: Tasten zum Bewegen des Roboters.
- „Input/Output“: Anzeigen und Einstellen der DIO-Schnittstellen der Robotersteuerung.
- „Programme & Variablen“: zeigt den aktuellen Status von Programmvariablen an.



Das „Hilfe“-Symbol in der rechten unteren Ecke enthält Links zu den Wiki-Seiten (bezogen auf „Online-Dokumentation“, „Software-Updates“, „Beispiele“, „Fehlerbehebung“) und einen Link zum „Support kontaktieren“.

7.1.1. Auswählen des Robotertyps

igus® Robot Control bietet projektbezogene Einstellungen für verschiedene Robotertypen, wie z.B. Delta- oder robolink-Roboter. Klicken Sie auf die Registerkarte „Datei“ in der linken oberen Ecke und wählen Sie „Projekt öffnen“. Wählen Sie die Robotergruppe, z.B. „drylin-delta“. Klicken Sie nun auf das benannte Bild, das zu Ihrem Roboter passt, z.B. DLE-DR-0001.

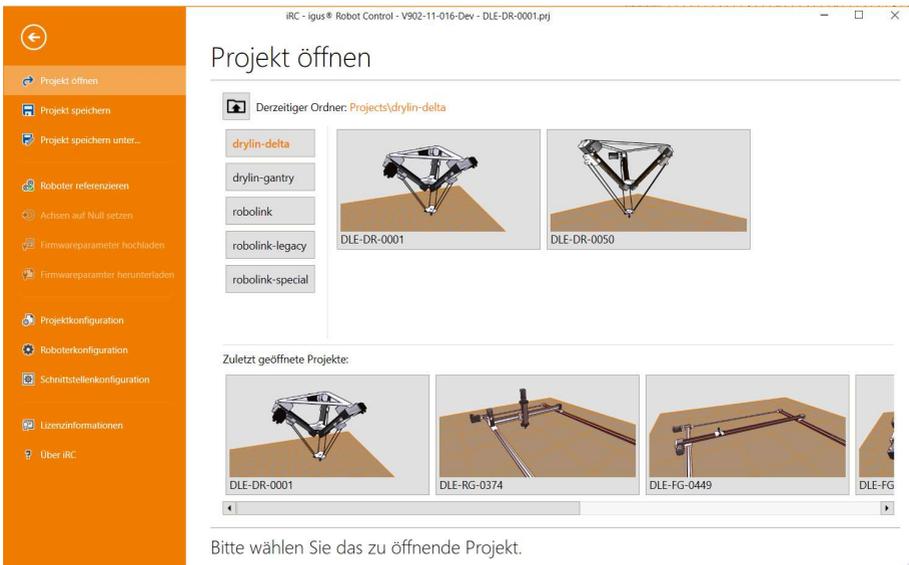


Abb. 7.2 Auswahl des Robotertyps über den Menüpunkt „Datei“ -> „Projekt öffnen“.

7.1.2. Maus: Navigation und Bewegen des Roboters in der Benutzeroberfläche

Zur Navigation in der 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.

7.2. Anschließen des Roboters

7.2.1. 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. 7.3). Danach muss der Roboter referenziert werden.



Abb. 7.3 Symbole für die Verbindung mit der Hardware, das Zurücksetzen von Fehlern und die Aktivierung der Motoren, Referenzierungssymbol und „Status“-Anzeige.

1. **„Verbinden“**: Stellen Sie die Verbindung zur Hardware her.
 - Hierüber wird die USB-CAN-Schnittstelle oder die Ethernet-Verbindung initialisiert.
 - 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 7.3 beschrieben wird.**

 - Die „Status“-Anzeige bleibt rot. Die Fehlermeldungen werden gelöscht, nur „Motoren nicht freigegeben“ 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.

7.2.2. 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 Abschnitte 7 und 7.4.

7.3. Referenzierung 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 igus® Robot Control angezeigt.

Die Schrittmotoren speichern die Motorposition in einem EEPROM. Aufgrund der Schwerkraft oder anderer Kräfte können sich die Gelenke jedoch bei ausgeschalteter Motorleistung bewegen und absacken. In diesem Fall melden die Schrittmotoren nicht mehr die korrekte Position an die Software, da sie ausgeschaltet wurden, während sich der Roboter bewegt hat. Um die Position zwischen Software, Schrittmotormodul und Roboterachse zu synchronisieren, muss eine Referenzierung durchgeführt werden.

7.3.1. Schrittweise Anleitung der Referenzierung

1. Starten Sie die Robotersteuerung und igus® Robot Control.
2. Drücken Sie die Schaltflächen „Verbinden“, „Zurücksetzen“ und „Aktivieren“ (Abb. 7.3).
3. Klicken Sie auf die Schaltfläche „Referenzieren“ (Abb. 7.3) in der Schaltflächen-Gruppe "Physikalischer Roboter" im Register "Bewegung" (oder „Roboter Referenzieren“ im Register „Datei“), um das Referenzierungsfenster zu öffnen.
4. Klicken Sie auf die Schaltflächen, um mit der Referenzierung der Achsen zu beginnen, siehe Abb. 7.4. 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.

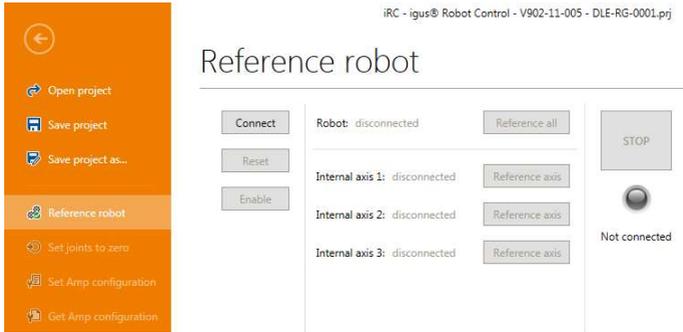


Abb. 7.4 Referenzierungsfenster.

7.4. Bewegen des Roboters mit Gamepad oder Software-Tasten

Der Roboter kann manuell bewegt (oder „gejoggt“) werden, während kein Programm läuft. Die beiden Hauptsteuerungen finden Sie im Register „Bewegung“:

- „Eingabe“ (zum Verbinden eines Gamepads),
- „Bewegungsparameter“ (Auswahl von Bewegungsmodi und Geschwindigkeit für die Software-Schaltflächen).



Abb. 7.5 Bedienfelder zum Bewegen des Roboters (blau markiert).

1. **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. 7.5):

- „Achse“: Ein Klick auf A1 bis A6 bewegt die entsprechende Roboterachse (falls vorhanden). E1 - E3 bewegt die externen Gelenke. Dies könnte eine lineare oder eine Rotationsschse sein (Abb. 7.6).
- „Basis“ (kartesischer Modus) bewegt den Roboter in geraden Linien entlang der X-, Y- und Z-Achse des Basis-Koordinatensystems.
- „Werkzeug“ (kartesischer Modus) bewegt den Roboter in X, Y und Z des aktuellen Werkzeugkoordinatensystems.

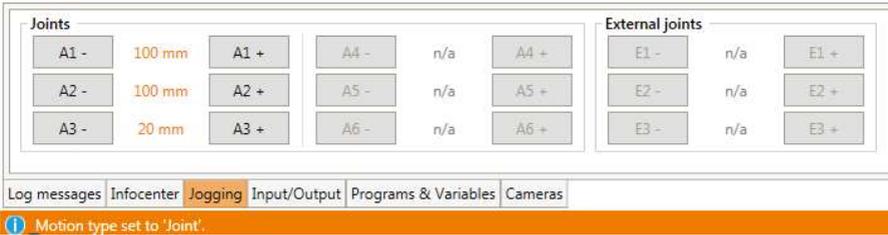


Abb. 7.6 Die Softwareschaltflächen für „Achse“-Bewegungen.
In beiden kartesischen Modi wechseln die Schaltflächen auf X, Y, Z, A, B und C.

2. Ein **Gamepad** ist möglicherweise die intuitivste Art, den Roboter zu bewegen. Die Abb. 7.7 zeigt die Tastenbelegung. Durch Drücken von „Gamepad verbinden“ verbindet sich igus® Robot Control mit einem Gamepad. Bei erfolgreicher Verbindung, wird ein OK-Zeichen unter dem Symbol angezeigt. Das Gerät muss vom Typ "Joystick" oder "Gamepad" sein. Weitere Informationen zur Verbindung finden Sie im Register „LogMessages“ (unten im Bildschirm).



1. Bewegungsart 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 Joint-Bewegungsmodus.

Abb. 7.7 Tastenbelegung des Gamepads

7.5. Starten von Roboterprogrammen

Das Roboterprogramm muss 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. `igus5DOF_TestMotion.xml`

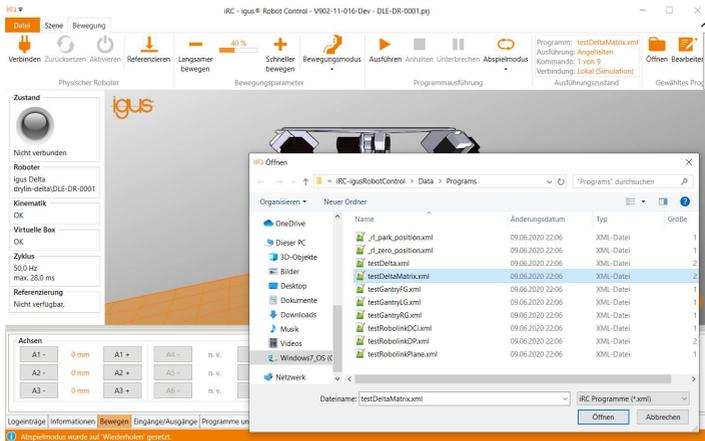


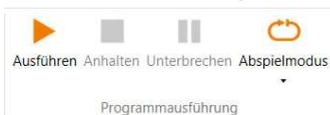
Abb.7.8 Laden eines neuen Programms.

2. 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).

7.6. Digitale Ein- und Ausgänge

Der einfachste Weg, eine speicherprogrammierbare Steuerung (SPS) anzuschließen, ist über digitale Ein- und Ausgänge. Jede Robotersteuerung wird mit einem DIO-Modul geliefert. Dieses bietet 7 Eingänge und 7 Ausgänge (siehe Abschnitt 4.2). Wenn zusätzliche Ein- und Ausgänge erforderlich sind, können bis zu zwei zusätzliche DIO-Module integriert werden, siehe Abschnitt 10.1. Insgesamt können bis zu 3 DIO-Module angesteuert werden.

Der Zustand der Ein- und Ausgänge kann unter „Eingänge/Ausgänge“ überwacht werden.

Sowohl Eingänge als auch Ausgänge können manuell eingestellt 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.

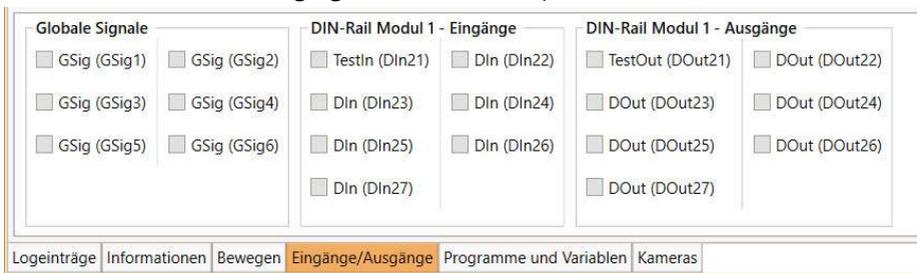


Abb. 7.9: „Eingabe-/Ausgabe“-Abschnitt der iRC-Software.

Beachten Sie bitte Abschnitt 4.5 zur Einrichtung der elektrischen Anschlüsse und Abschnitt 10.1 zur Erweiterung der Anzahl der verfügbaren digitalen Ein-/Ausgänge. Es ist auch möglich, die verschiedenen Ein-/Ausgänge umzubenennen.

Die Programmierung mit digitalen Eingängen / Ausgängen wird in den Abschnitten **Fehler! Verweisquelle konnte nicht gefunden werden.** und 8.6 erläutert.

7.7. Software-Schnittstellen

Die Robotersteuerung stellt verschiedene Schnittstellen zur Verfügung:

- SPS-Schnittstelle zur Steuerung des Roboters über eine SPS, insbesondere zum Starten und Stoppen von Programmen.
- Plug-in-Schnittstelle zur Integration von z.B. Kameras. Das Plug-in überträgt dann die Zielpositionen an die Robotersteuerung.

- Container Runtime Interface (CRI), um weitere Interaktion zu ermöglichen. Diese Schnittstelle kann z.B. dazu verwendet werden, um werkstückspezifische Programme aus einer Datenbank zu generieren.
- ROS-Schnittstelle zum Betrieb des Roboters über RobotOperatingSystem (www.ros.org). Siehe Abschnitt 10 für die Konfiguration dieser Schnittstellen.

7.8. Aktualisieren der Software

Updates der iRC-Software finden Sie auf der [igus LowCostAutomation-Website](http://www.igus.com).

BackUp: Bitte benennen Sie Ihren alten iRC-igus Robot Control-Ordner vor Beginn der Installation in z. B. C:\iRC-igusRobotControl_BAK um. Auf diese Weise können Sie im Zweifelsfall wieder zur alten Version zurückkehren. Folgendes muss von der vorherigen Installation übernommen werden:

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

8. Programmierung des Roboters mit iRC

igus® Robot Control ermöglicht die Erstellung von Roboterprogrammen. Die Art der Programmierung wird als „Teach-In-Programmierung“ bezeichnet, die 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.

8.1. Der Programm-Editor

Im Roboterprogramm besteht jeder Befehl aus einer Zeile, z.B. „Joint“ oder „Wait“. Nur Befehle, die den Programmfluss steuern, werden in mehrere Zeilen aufgeteilt, z.B. „Loop“ und „EndLoop“. Der Programmeditor wird mit der Schaltfläche „Bearbeiten“ in der Registerkarte „Bewegung“ geöffnet.



Abb. 8.1: Öffnen des Programmeditors mit der Schaltfläche „Bearbeiten“.

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

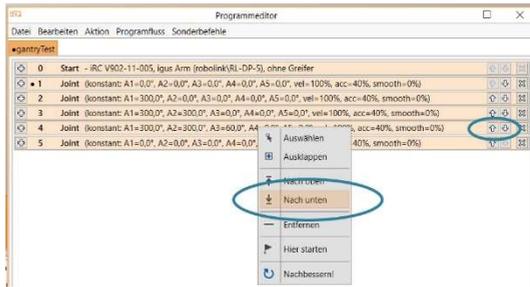
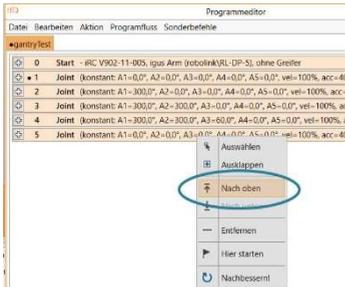


Abb. 8.2: Programmmeditor mit einem kurzen Programm.

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

8.1.1. Ändern der Befehlssequenz

Um einen Befehl zu verschieben, klicken Sie im Kontextmenü auf „Nach unten“ oder auf „Nach oben“. Alternativ können Sie auch die Pfeile auf der rechten Seite der Befehle zu benutzen.

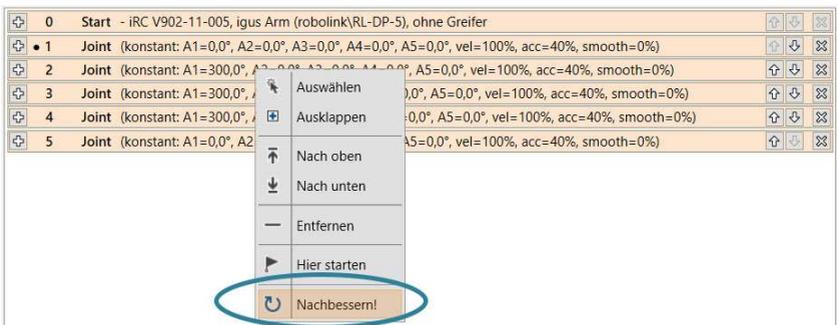


Der Programm-Editor verhindert unerlaubte Befehle, die die Struktur des Programms beschädigen würden. Wenn das Verschieben eines Befehls nach oben oder unten nicht möglich ist, werden die entsprechenden Schaltflächen und Menüpunkte ausgegraut.

8.1.2. 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“:

- Wählen Sie den Befehl aus (siehe Abschnitt 1.3.1) und drücken Sie dann **Strg+T**
- Alternativ Klicken Sie auf „Nachbessern!“ im Kontextmenü, das beim Klicken mit der rechten Maustaste auf die Befehlszeile erscheint:



Diese Funktionalität ist auch im Menü zu finden: „Bearbeiten“ -> „Nachbessern!“

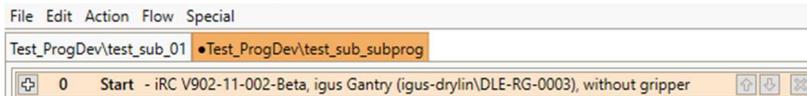
Der Programmierer ersetzt dann die Positionswerte im Befehl durch die aktuelle Position des Roboters.

8.1.3. Startbefehl festlegen

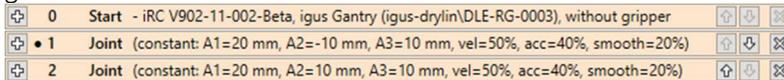
Es ist möglich, Programme befehlsweise auszuführen oder einen bestimmten Befehl als Startpunkt eines Programms zu Debugging-Zwecken auszuwählen.

Der Befehl, der beim nächsten Start des Programms zuerst ausgeführt wird, oder - falls das Programm gerade läuft - der aktuell ausgeführte Befehl wird im Programmeditor angezeigt.

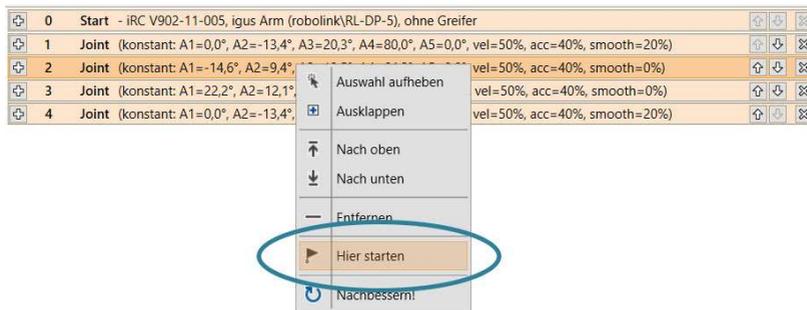
Das Unterprogramm, das diesen Befehl enthält, ist durch einen ausgefüllten schwarzen Kreis vor dem Programmnamen gekennzeichnet:



Der Befehl selbst ist durch einen ausgefüllten schwarzen Kreis in der Kopfleiste des Befehls gekennzeichnet:



Um einen bestimmten Befehl als Startpunkt für die Ausführung zu wählen, klicken Sie auf „Hier starten“ im Kontextmenü, das sich beim Klicken mit der rechten Maustaste auf die Befehlszeile öffnet:

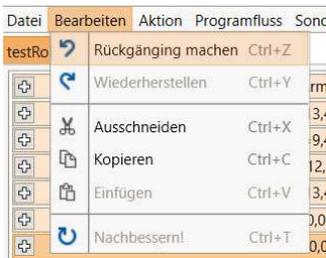


8.1.4. Rückgängig machen und wiederherstellen

Alle im Programmierer vorgenommenen Änderungen können rückgängig gemacht / wiederhergestellt werden.

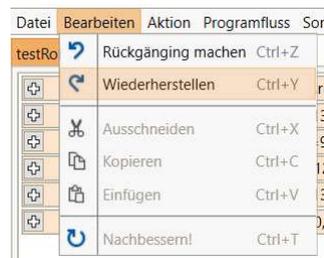
Zum **Rückgängig machen** können Sie

- Strg+Z drücken.
- auf den Menüpunkt „Bearbeiten“ -> „Rückgängig machen“ klicken (er wird ausgegraut, wenn bisher keine Änderungen vorgenommen wurden).



Um eine zuvor rückgängig gemachte Änderung **wiederherzustellen**, können Sie

- Strg+Y drücken
- auf den Menüpunkt „Bearbeiten“ -> „Wiederherstellen“ klicken (er wird ausgegraut, wenn zuvor keine Änderungen rückgängig gemacht wurden)



8.2. Kommentare und Informationen in Programmen

8.2.1. Informationen zum Programm

Der Programmierer führt zu Beginn jedes Programms den Pseudo-Befehl **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.



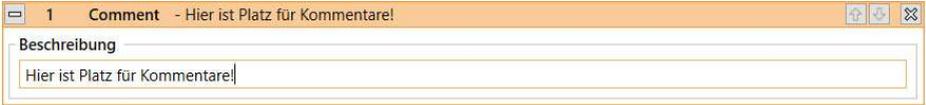
8.2.2. Beschreibungen

Jeder Befehl eines Programms enthält eine Beschreibung. Sie kann mit beliebigem Text gefüllt werden und sollte dazu verwendet werden, anderen Nutzern zu beschreiben, wozu der jeweilige Befehl dient.



8.2.3. 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 iRC-Programmeditor im Menüeintrag „Sonderbefehle“ -> „Kommentar“ zu finden.



8.3. 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.

8.3.1. Benutzer-Variablen

Es ist möglich, Benutzer-Variablen mit dem Befehl **Store** zu definieren, der im Programm-Editor des iRC über die Menüeinträge unter „Sonderbefehle“ -> „Variablendefinition“ zugänglich ist.



Es ist möglich, drei Arten von Speicheroperationen zu wählen:

- „Derzeitige Position“:

The screenshot shows a dialog box titled "1 Store (initialisiere myPosVar mit der derzeitigen Position)". Under the "Parameter" section, the "Variable" is set to "myPosVar" and the "Typ" is set to "Derzeitige Position". There is a "Beschreibung" field below.

Eine Positionsvariable wird mit der kartesischen und Achsposition initialisiert, die der Roboter bei der Ausführung des Befehls hat.

- „Zahlenkonstante“:

The screenshot shows a dialog box titled "1 Store (initialisiere myPosVar mit der Zahl 0)". Under the "Parameter" section, the "Variable" is "myPosVar", the "Typ" is "Zahlenkonstante", and the "Wert" is "0,00". There is a "Beschreibung" field below.

Eine Zahlenvariable wird mit der unter „Wert“ angegebenen Konstante initialisiert.

- „Positionskonstante“:

The screenshot shows a dialog box titled "1 Store (initialisiere myPosVar mit der Position X=327 mm, Y=0 mm,...)". Under the "Parameter" section, the "Variable" is "myPosVar" and the "Typ" is "Positionskonstante". Below this, there are sections for "Kartesische Position" and "Achsisposition".

Kartesische Position

X:	326,7 mm	Y:	0,0 mm	Z:	262,3 mm
A:	180,00 °	B:	0,00 °	C:	180,00 °

Achsisposition

A1:	0,00 °	A2:	10,00 °	A3:	0,00 °
A4:	80,00 °	A5:	0,00 °	A6:	n. v.

Externe Achsen

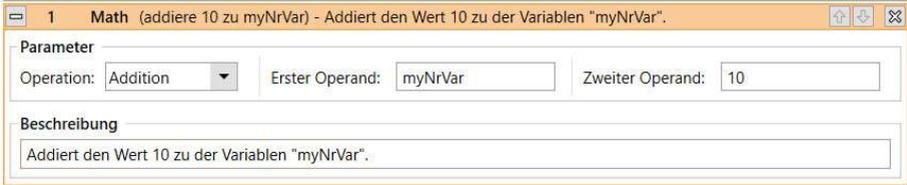
E1:	n. v.	E2:	n. v.	E3:	n. v.
-----	-------	-----	-------	-----	-------

There is also a "Beschreibung" field at the bottom.

Eine Positionsvariable wird mit den unter „Kartesische Position“, „Achsisposition“ und „Externe Achsen“ angegebenen Konstanten initialisiert. Je nach dem kinematischen Modell des aktuellen Roboters kann es sein, dass bestimmte Achsen nicht verfügbar sind.

Es sind nur Namen von benutzerdefinierten Zahlenvariablen oder Komponenten von benutzerdefinierten Positionsvariablen erlaubt.

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

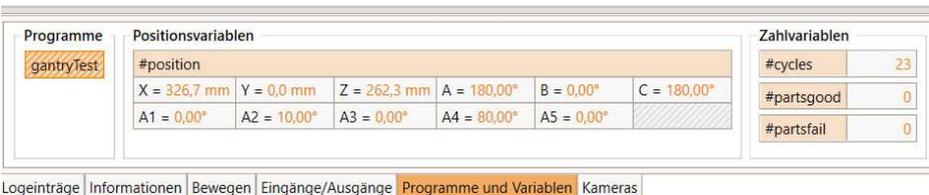


Die folgenden Operationen werden unterstützt und können unter "Operation" ausgewählt werden:

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

8.3.5. Variablen überwachen

Sie können die aktuellen Werte aller definierten Variablen unter iRC in der Registerkarte „Variablen“ überwachen:



8.4. Ablauf der Programmausführung

8.4.1. 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:

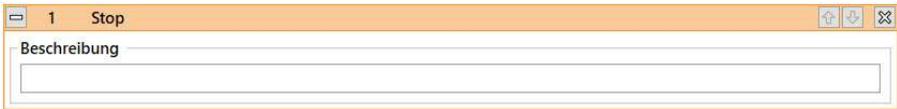
- `DIn23` Wahr, wenn Digitaleingang 23 gesetzt ist.
- `DIn23 AND !DIn27` Wahr, wenn Digitaleingang 23 gesetzt und 27 nicht gesetzt ist.
- `modelclass = 31` Wahr, wenn die Variable „modelclass“ 31 ist
- `mempos.x > 350.0` Wahr, wenn die x-Komponente der Positionsvariablen „mempos“ größer als 350 ist.

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

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

8.4.2. Stop

Der Befehl **Stop** stoppt die Programmausführung.

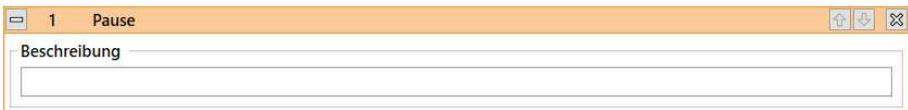


Er ist über den Menüeintrag „Programmfluss“ -> „Anhalten“ verfügbar.



8.4.3. Pause

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



Er ist über den Menüeintrag „Programmfluss“ -> „Pausieren“ verfügbar.



8.4.4. Wait

Der Befehl **Wait** erlaubt es dem Roboter, eine bestimmte Zeitspanne zu warten oder bis eine bedingte Anweisung als wahr bewertet wird. Er ist über die Menüeinträge unter „Programmfluss“->„Warten“ im Programmmeditor des iRC zugänglich.



Die verschiedenen Modi können unter "Typ" ausgewählt werden:

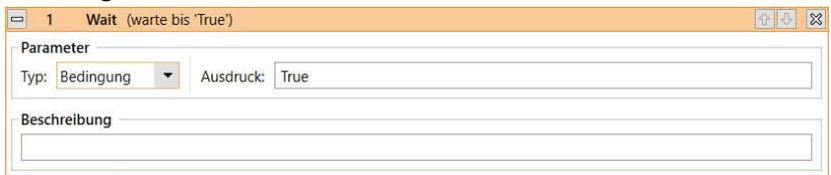
- „Zeitspanne“:

Dies führt dazu, dass der Roboter die unter „Zeitspanne“ angegebene Zeit abwartet.



- „Bedingung“:

In diesem Fall wartet der Roboter, bis die unter „Ausdruck“ angegebene bedingte Anweisung als „wahr“ bewertet wird.

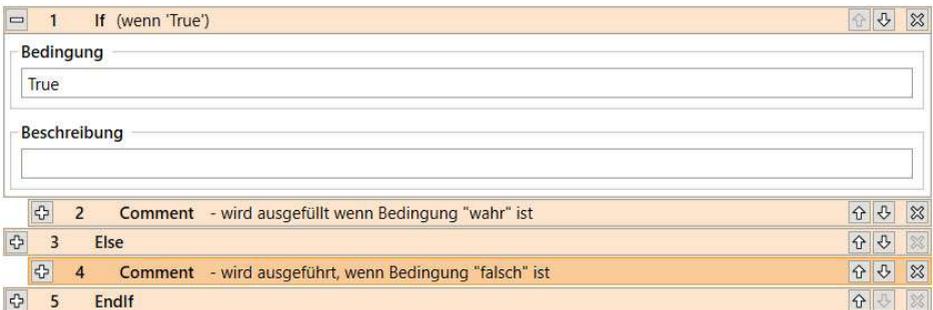


8.4.5. If-then-else

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 Programmeditor des iRC zugänglich.



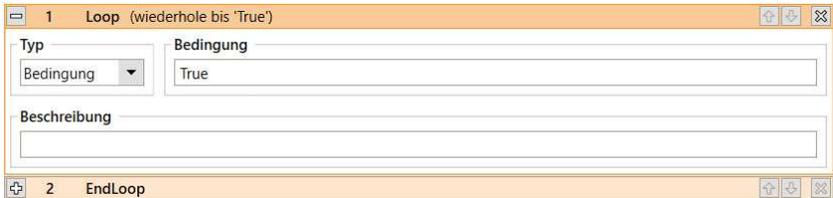
Der unter „Bedingung“ angegebene bedingte Ausdruck muss der in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** beschriebenen Syntax entsprechen. Die Anweisungen zwischen „If“ und „Else“ werden ausgeführt, wenn die Bedingung als wahr ausgewertet wird. Andernfalls werden die Anweisungen zwischen „Else“ und „EndIf“ ausgeführt.



8.4.6. Schleifen

Der Befehl **Loop** ermöglicht die Definition von Ausführungsschleifen. Unter „Typ“ kann zwischen den folgenden Schleifentypen gewählt werden:

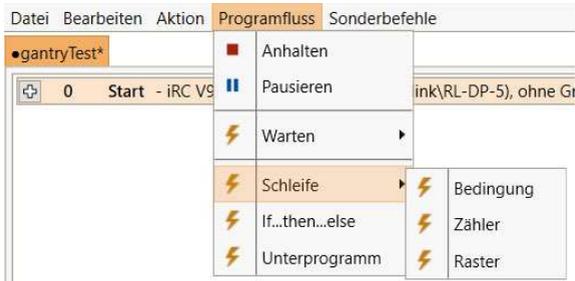
- „Bedingung“.
Die Schleife wird so lange wiederholt, bis der unter „Bedingung“ angegebene bedingte Ausdruck als „wahr“ gewertet wird. Sie muss der in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** beschriebenen Syntax folgen.



- „Zähler“
Die Schleife wird die unter „Wiederholungen“ angegebene Anzahl von Malen wiederholen.

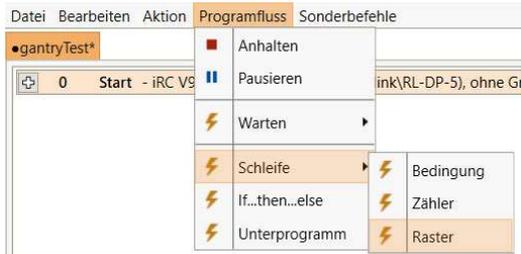


Der Schleifenbefehl ist über die Menüpunkte „Programmfluss“ -> „Schleife“ zugänglich.

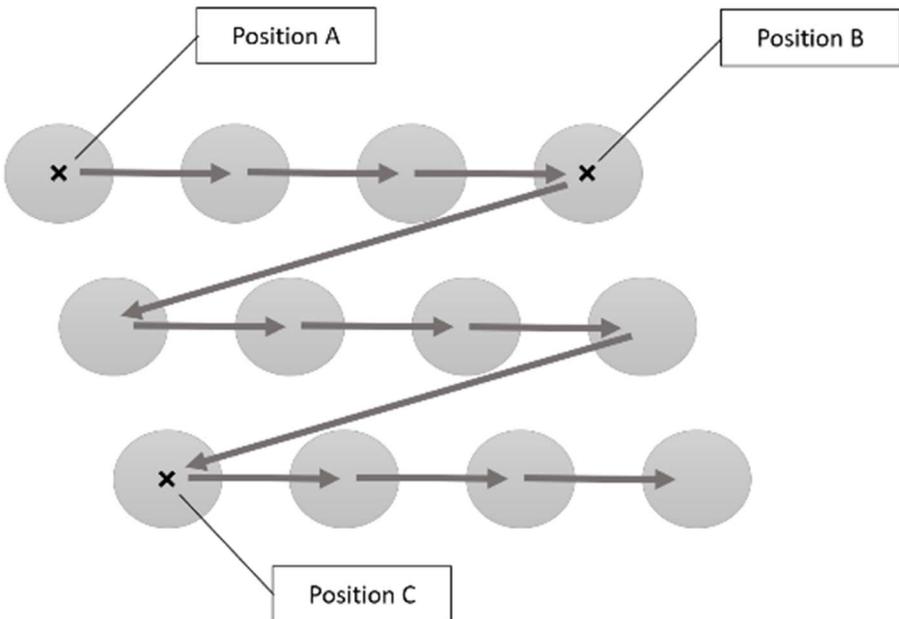


8.4.7. Matrix-Schleifen

Der Befehl **Matrix** ist so konzipiert, dass er Schleifen ausführt, die es dem Roboter ermöglichen, Rasterbewegungen auszuführen, beispielsweise für Palettierungsaufgaben. Er kann über den Menüpunkt „Programmfluss“ -> „Schleife“ -> „Raster“ aufgerufen werden.



Die folgende Abbildung ein Bewegungsmuster, das durch die Verwendung des Matrix-Befehls ausgeführt werden kann.



Die als „Punkt A“, „Punkt B“ und „Punkt C“ angegebenen Positionsvariablen definieren die Ecken des Bereichs, der von der Matrixschleife abgedeckt wird (siehe Abbildung oben). Die Anzahl der durchzuführenden Schritte wird durch „Zähler X“ (von A nach B) und „Zähler Y“ (von A nach C) angegeben. In der obigen Abbildung wäre beispielsweise X 4 und Y 3.

1 **Matrix** (3x3-Raster, A:myPointA, B:myPointB, C:myPointC, Ziel:myMatrixPos, X:myCntX, Y:myCntY)
↑ ↓ ×

Abmessungen

X (von A nach B): Stops Y (von A nach C): Stops

Eckpunkte

Punkt A: Punkt B: Punkt C:

Variablen

Zielposition: Zähler X: Zähler Y:

Beschreibung

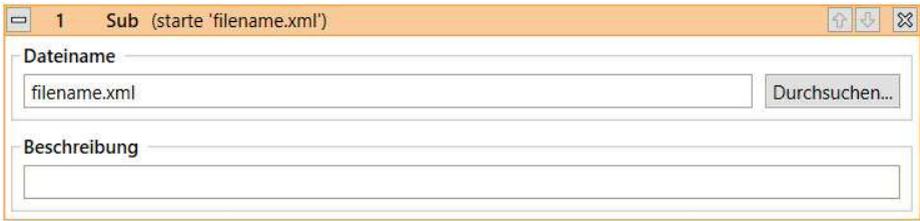
2 **Comment** - Hier stehen die Befehle, die bei jedem Schritt ausgeführt werden.
↑ ↓ ×

3 **Matrix End**
↑ ↓ ×

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.

8.4.8. 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.



8.5. Bewegung

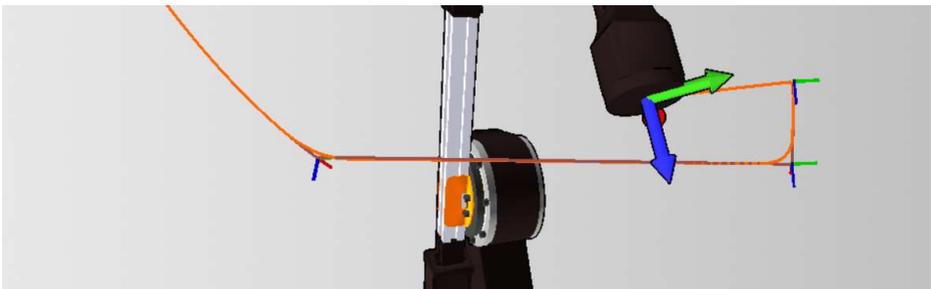
8.5.1. Abbruch-Bedingungen

Jeder Bewegungsbefehl kann mit einer Abbruchbedingung versehen werden. Es handelt sich um einen bedingten Ausdruck, der in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** 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:

Abbruchbedingung
False

8.5.2. Beschleunigung und Glättung

In Bewegungsbefehlen können Achsbeschleunigungen definiert werden, um abrupte Bewegungen des Roboters zu verhindern. Dies kann für einen einzelnen Bewegungsbefehl erfolgen, aber auch für eine Kombination von bis zu maximal 10 Befehlen. Es ist auch möglich, die Ecken innerhalb der durch Blöcke aufeinanderfolgender Bewegungsbefehle definierten Bahn zu glätten. Dies ermöglicht es dem Roboter, durch die Ecken zu fahren, ohne dass er zum Stillstand kommt. Wenn die Eckenglättung auf 0% eingestellt ist, stoppt der Roboter an jeder Ecke für einen kurzen Moment. Dies führt zu einer weniger flüssigen Bewegung.



Werden Bewegungsbefehle von anderen Befehlen unterbrochen, wird der Bewegungssatz aufgeteilt und die Glättung unterbrochen.

Parameter:

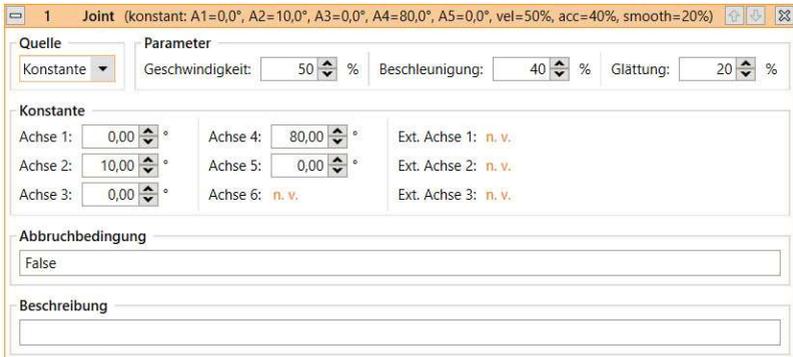
Beschleunigung: % Glättung: %

- „Beschleunigung“:
Prozentualer Anteil der maximal möglichen Beschleunigung.
- „Glätten“:
Umfang der Glättung in Prozent.

8.5.3. Achsbewegung

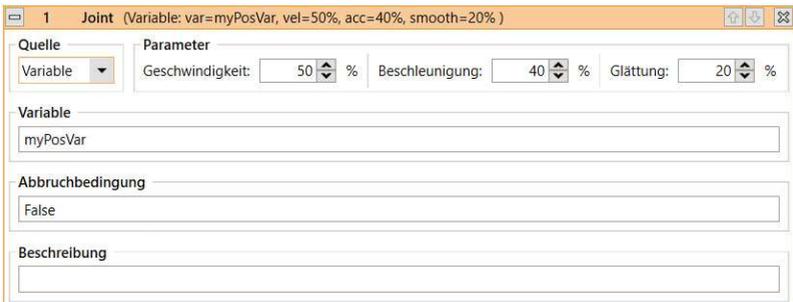
Der Befehl **Joint** bewegt den Roboter zu einer (absoluten) Zielposition, die in Achskoordinaten angegeben ist. Die daraus resultierende Bewegung des TCP ist in der Regel eine Kurve und keine gerade Linie. Die Zielposition kann auf folgende Weise angegeben werden (wählen Sie die entsprechende „Quelle“):

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



The screenshot shows the 'Joint' configuration window with the title '1 Joint (konstant: A1=0,0°, A2=10,0°, A3=0,0°, A4=80,0°, A5=0,0°, vel=50%, acc=40%, smooth=20%)'. The 'Quelle' dropdown is set to 'Konstante'. The 'Parameter' section includes 'Geschwindigkeit: 50 %', 'Beschleunigung: 40 %', and 'Glättung: 20 %'. The 'Konstante' section shows target positions for six axes: Achse 1: 0,00°, Achse 2: 10,00°, Achse 3: 0,00°, Achse 4: 80,00°, Achse 5: 0,00°, and Achse 6: n. v. The 'Abbruchbedingung' is set to 'False' and the 'Beschreibung' field is empty.

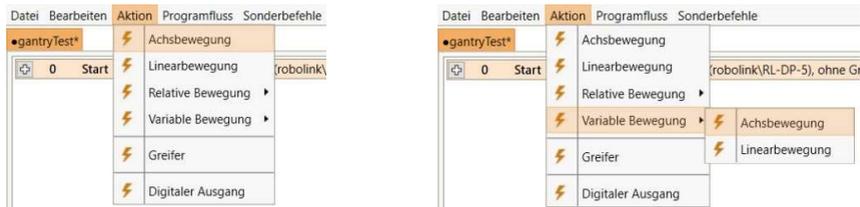
- „Variable“:
Die Zielposition wird aus der Positionsvariablen übernommen, die unter „Variable“ angegeben wird.



The screenshot shows the 'Joint' configuration window with the title '1 Joint (Variable: var=myPosVar, vel=50%, acc=40%, smooth=20%)'. The 'Quelle' dropdown is set to 'Variable'. The 'Parameter' section includes 'Geschwindigkeit: 50 %', 'Beschleunigung: 40 %', and 'Glättung: 20 %'. The 'Variable' section has the text 'myPosVar' entered. The 'Abbruchbedingung' is set to 'False' and the 'Beschreibung' field is empty.

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

Der Joint-Befehl kann im iRC-Programmreditor unter den Menüeinträgen „Aktion“ -> „Achsbewegung“ und „Aktion“ -> „Variable Bewegung“ -> „Achsbewegung“ aufgerufen werden.



8.5.4. 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 und Eulerwinkel A , B , C sowie die Positionen der externen Achsen gegeben ist, wenn sie von der aktuellen Roboterkinematik unterstützt wird.

1 Linear (konstant: X=327 mm, Y=0 mm, Z=262 mm, A=180,0°, B=0,0°, C=180,0°, vel=100 mm/s,...

Quelle: Konstante

Parameter: Geschwindigkeit: 100,0 mm/s, Beschleunigung: 40 %, Glättung: 20 %

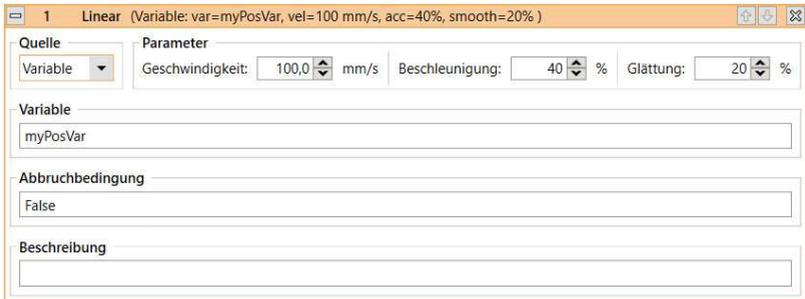
Konstante:

X: 326,7 mm	A: 180,00 °	Ext. Achse 1: n. v.
Y: 0,0 mm	B: 0,00 °	Ext. Achse 2: n. v.
Z: 262,3 mm	C: 180,00 °	Ext. Achse 3: n. v.

Abbruchbedingung: False

Beschreibung:

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



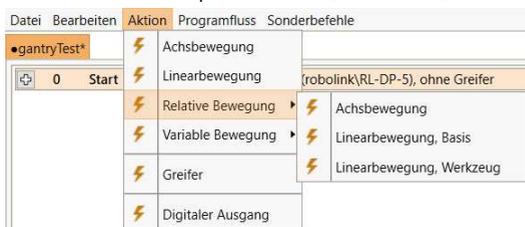
Die Bewegungsgeschwindigkeit wird durch „Geschwindigkeit“ in mm/s angegeben. Wenn sie die maximal zulässige Bewegungsgeschwindigkeit des Roboters überschreitet, führt dies zu einem kinematischen Fehler während der Ausführung.

Der Linear-Befehl kann im iRC-Programmmeditor unter „Aktion“ -> „Linearbewegung“ und „Aktion“ -> „Variable Bewegung“ -> „Linearbewegung“ aufgerufen werden.



8.5.5. 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.

The screenshot shows the 'Relative' dialog box for the 'Achsbewegung' mode. The title bar indicates '1 Relative (Achsen: A1=0,0°, A2=0,0°, A3=0,0°, A4=0,0°, A5=0,0°, vel=20%, acc=40%, smooth=20%)'. The 'Typ' dropdown is set to 'Achsbewegung'. The 'Parameter' section shows 'Geschwindigkeit: 20 %', 'Beschleunigung: 40 %', and 'Glättung: 20 %'. The 'Versatz' section has input fields for Achse 1, 2, 3, 4, 5, and 6, all set to 0,00°. The 'Abbruchbedingung' is set to 'False'. The 'Beschreibung' field is empty.

- „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.

The screenshot shows the 'Relative' dialog box for the 'Linear - Basis' mode. The title bar indicates '1 Relative (Basis: X=0 mm, Y=0 mm, Z=0 mm, vel=50 mm/s, acc=40%, smooth=20%)'. The 'Typ' dropdown is set to 'Linear - Basis'. The 'Parameter' section shows 'Geschwindigkeit: 50,0 mm/s', 'Beschleunigung: 40 %', and 'Glättung: 20 %'. The 'Versatz' section has input fields for X, Y, and Z, all set to 0,0 mm. The 'Abbruchbedingung' is set to 'False'. The 'Beschreibung' field is empty.

- „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.

The screenshot shows a CNC control interface for a linear tool movement. The title bar indicates the mode is 'Relative' with tool coordinates (X=0 mm, Y=0 mm, Z=0 mm), velocity (50 mm/s), acceleration (40%), and smoothness (20%).

Typ		Parameter	
Linear - Werkzeug		Geschwindigkeit: 50,0 mm/s	Beschleunigung: 40 %
		Glättung: 20 %	

Versatz			
X: 0,0 mm	A: n. v.	Ext. Achse 1: n. v.	
Y: 0,0 mm	B: n. v.	Ext. Achse 2: n. v.	
Z: 0,0 mm	C: n. v.	Ext. Achse 3: n. v.	

Abbruchbedingung
False

Beschreibung

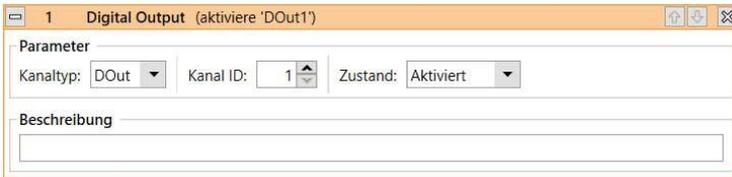
8.6. Greifer und digitale Ein- / Ausgänge

8.6.1. Digitale Eingänge

Digitale Eingänge und globale Signale sind als reservierte Wörter in bedingten Anweisungen verfügbar (siehe Abschnitt 4.1) Bei globalen Signalen sind dies GSig1, GSig2, ..., bei digitalen Eingängen Din1, Din2, ... Je nach Zustand werden sie als „wahr“ oder „falsch“ ausgewertet.

8.6.2. Digitale Ausgänge

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



Unter „Kanaltyp“ wird angegeben, ob ein digitaler Ausgang oder ein globales Signal gesetzt werden soll. Unter „Kanal ID“ wird der Kanal des digitalen Ausgangs oder des globalen Signals, unter „Zustand“ der gewünschte Zustand nach Ausführung des Befehls angegeben. Der Befehl ist im Programmierer des iRC unter „Aktion“ -> „Digitaler Ausgang“ zugänglich.



8.6.3. Öffnen/Schließen des Greifers

Der Befehl **Gripper** ermöglicht die Steuerung des Greifers des Roboters. Er ist im Programmierer des iRC 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%.



The image shows a software window titled "1 Gripper (Greifer 60% öffnen)". The window has a standard OS-style title bar with minimize, maximize, and close buttons. Below the title bar, there are two main sections: "Parameter" and "Beschreibung". In the "Parameter" section, there is a label "Öffnung:" followed by a spin box containing the number "60" and a percentage sign "%". The "Beschreibung" section contains an empty text input field.

8.7. Kamera

Wir arbeiten derzeit an dieser Funktion. Die Befehlsspezifikation wird bald folgen.

9. Stand-alone-Betrieb mit integriertem Computer und dem Handbediengerät

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

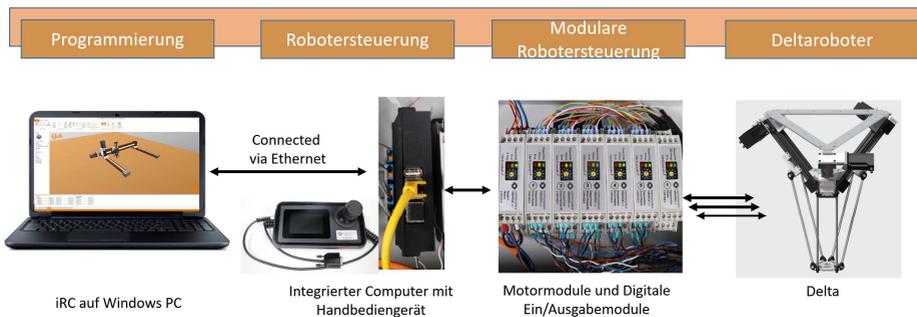


Abb. 9.1: Delta-Roboter mit integriertem Computer als Robotersteuerung

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

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

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

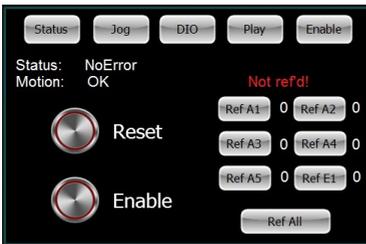
9.1. Fehler zurücksetzen/Roboter aktivieren



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



Drücken Sie jetzt „Reset“:
Der Status wechselt zu „MNE“ (Motor Not Enabled)



Drücken Sie „Enable“:
Der Status wechselt zu „NoError“.
„Not reFd!“ (in rot) bedeutet „nicht referenziert“.

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

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



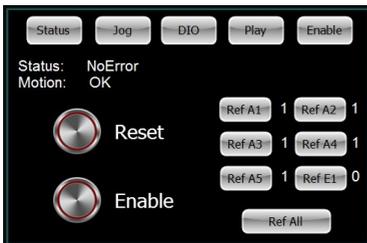
1. Drücken Sie dazu die Taste „Jog“ oben auf dem Display.
2. Drücken Sie „Joints 1 2 3“.
Dann bewegen und drehen Sie den Joystick.
Sie können nun die Achsen 1, 2 und 3 bewegen.

9.3. Referenzierung

Um einen automatischen Programmablauf zu ermöglichen, muss der Roboter referenziert werden. Die Art der Referenzbewegung hängt vom Roboter- oder Encodertyp ab.



1. Drücken Sie dazu den quadratischen „Enable“-Knopf am oberen Rand des Displays.
2. Der Roboter muss sich im Status „NoError“ befinden. Wenn er sich in einem Fehlerzustand befindet, aktivieren Sie ihn, indem Sie die runde Schaltfläche „Reset“ gefolgt von der Schaltfläche „Enable“ drücken.
3. Drücken Sie nun die Schaltfläche „Ref All“, um alle Achsen zu referenzieren. Der Roboter führt nun Suchbewegungen für jede Achse aus.
4. Die Referenzierung ist abgeschlossen, wenn eine 1 neben Ref A1-A1 steht. Ref E1 ist für zusätzliche externe Achsen vorgesehen.
5. Nach der Referenzierung befinden sich die Achsen im Fehlerzustand. Dies ist notwendig, weil sich die Ist-Position während des Referenzierungsvorgangs geändert hat.
6. Drücken Sie nun nochmals den runden Knopf „Reset“, gefolgt von dem runden Knopf „Enable“. Der Roboter ist nun referenziert.



9.4. Starten und Stoppen eines Programms

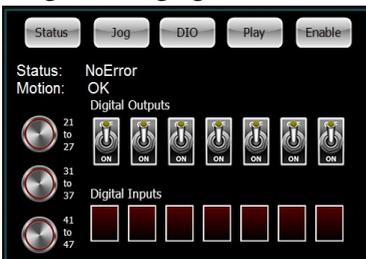


1. Drücken Sie dazu die Taste „Play“ am oberen Rand des Displays und wählen Sie mit den Tasten „prev“ und „next“ das Programm *igus5DOF_TestMotion* aus.
2. Laden Sie das Programm mit der Schaltfläche „Load“.
3. Lassen Sie das Programm mit „Single Play“ einmal abspielen.
4. „Cont. Play“ spielt das geladene Programm kontinuierlich ab.
5. „Stop“ hält die Bewegung an.

Der Schieberegler „Override“ kann nach rechts bewegt werden, um die Geschwindigkeit der Bewegung zu erhöhen, oder nach links, um sie zu verringern.

9.5. Manuelle Einstellung der digitalen Ein-/Ausgänge

Die digitalen Ausgänge können durch Drücken der abgebildeten Schalter gesetzt werden.



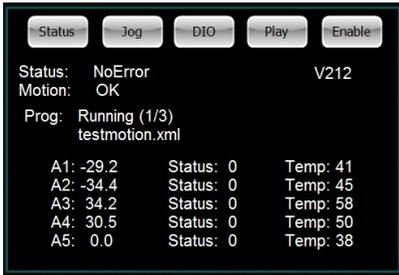
Wenn z.B. ein Greifer an das DIO-Modul angeschlossen ist, kann dieser durch Umschalten der abgebildeten Schalter aktiviert oder deaktiviert werden.

Wenn ein Signal am digitalen Eingang anliegt, wird es im Feld für die digitalen Eingänge am unteren Rand des Displays angezeigt.

Mit den drei Schaltflächen auf der linken Seite können Sie zwischen mehreren DIO-Modulen wechseln.

9.6. Anzeige von Statusinformationen

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



Der Name eines Programms wird angezeigt, z.B. während ein Programm läuft.

Die Achspositionen der Achsen 1-3 werden angezeigt (A1-A3).

Die Temperatur der Steppermodule wird ebenfalls angezeigt. Aufgrund der unterschiedlichen Belastungen der einzelnen Achsen werden unterschiedliche Halteströme angelegt, die zu wechselnden Temperaturen der Schrittmodule führen.

9.7. Programme auf dem integrierten Computer organisieren

Diese iRC-Funktion ist nur in Verbindung mit dem integrierten Computer verfügbar. Beim Schreiben von Programmen auf der Windows-iRC-Software werden diese Programme automatisch auch auf den integrierten Computer heruntergeladen.

Mit der Zeit könnte es notwendig werden, diese Programme neu zu organisieren, z.B. einige davon zu löschen. Dies ist mit der Schaltfläche „Programme verwalten“ in der Registerkarte „integrierter Computer“ möglich.

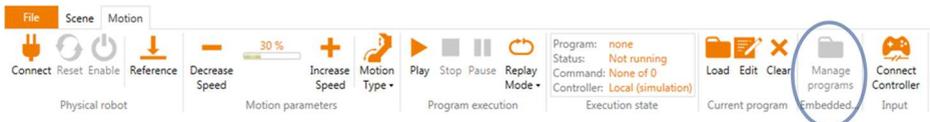


Abb. 9.2 igus® Robot Control Software mit dem Symbol „Programme verwalten“ (blau eingekreist).

10. Projekt-Konfiguration

10.1. Programm

Mit „Projektkonfiguration“ -> „Programm“ kann der iRC-Start definiert werden:

- Datei: welches Programm soll beim Start geladen soll
- Logik-Programmdatei: welches Logikprogramm soll beim Start geladen werden. Die Logikdatei läuft parallel zum Hauptprogramm und ermöglicht z.B. das Setzen von Status-Leuchten.
- Bewegungsgeschwindigkeit: der Override (Geschwindigkeit), der nach dem Start eingestellt wird.
- Abspielmodus: der Abspielmodus, der nach dem Start eingestellt wird

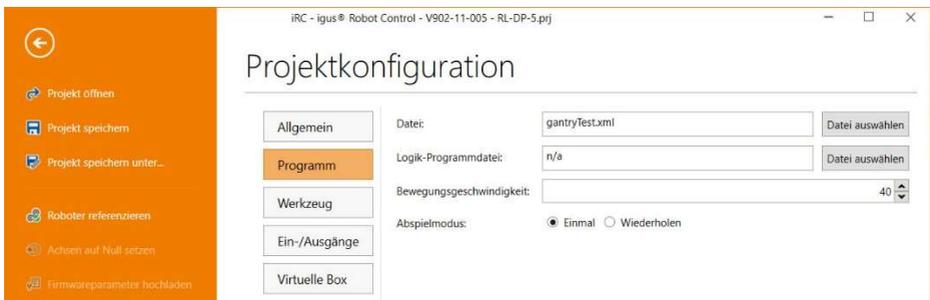


Abb. 10.1: Konfiguration des Startprogramms im Backstage-Bereich.

10.2. Werkzeug

Mit „Projektkonfiguration“ -> „Werkzeug“ wird das am Roboter angebrachte Werkzeug ausgewählt. Diese Einstellung ändert die Roboterkinematik! Programme, die mit einer anderen Kinematik-Einstellung geschrieben wurden, können nicht wiedergegeben werden, da dies zu einer Kollision führen könnte.



Abb. 10.2: Konfiguration des Werkzeugs in der Projektkonfiguration.

10.3. Eingänge / Ausgänge

Mit „Projektkonfiguration“ -> „Ein- / Ausgänge“ kann die Anzahl der verfügbaren Ein- und Ausgänge konfiguriert und umbenannt werden. Die entsprechenden Namen werden im Register „Ein-/Ausgänge“ angezeigt.

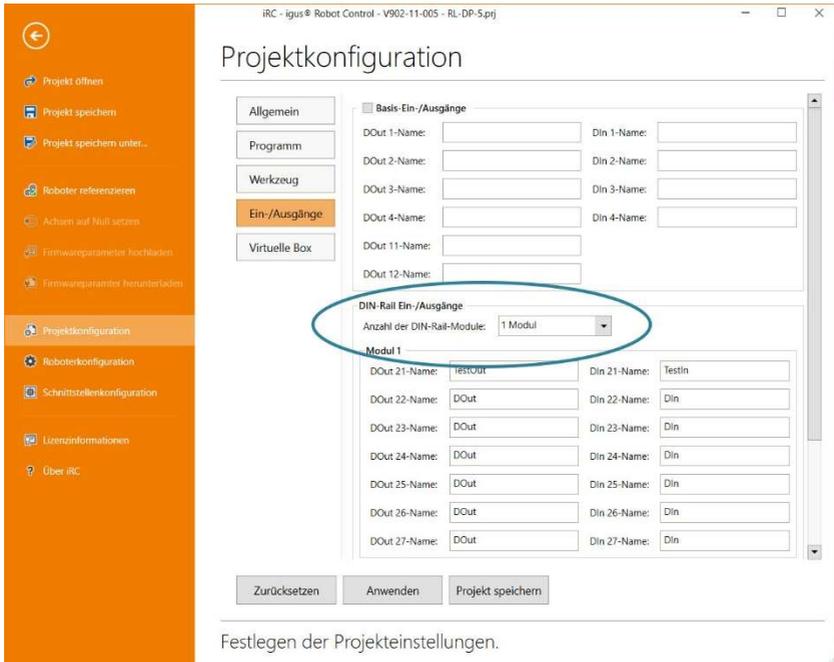


Abb. 10.3: Konfiguration der Ein-/Ausgänge in der Projektkonfiguration.

Für die robolink-Robotersteuerung wird der Bereich „Basis Ein-/Ausgänge“ nicht genutzt. Relevant ist der Bereich „Din-Rail Ein-/Ausgänge“.

Die Standard-Robotersteuerung wird mit einem DIO-Modul mit 7 Ein- und 7 Ausgängen geliefert. Sie können hier Namen für die Ein-/Ausgänge definieren.

Wenn ein weiteres DIO-Modul hinzugefügt wird, dann muss die Einstellung auf „2 Module“ oder „3 Module“ geändert werden.

Die Einstellungen werden automatisch mit dem integrierten Computer synchronisiert.

10.4. Virtuelle Box

Die Einstellungen der virtuellen Box erlauben es, den Bewegungsbereich des Roboterarms einzuschränken. Diese Einschränkungen werden sowohl bei der linearen als auch bei der Achsbewegung überwacht.



Diese Funktionalität hilft, mechanische Schäden zu vermeiden. Es handelt sich nicht um eine Funktionalität, die für die persönliche Sicherheit geeignet ist!

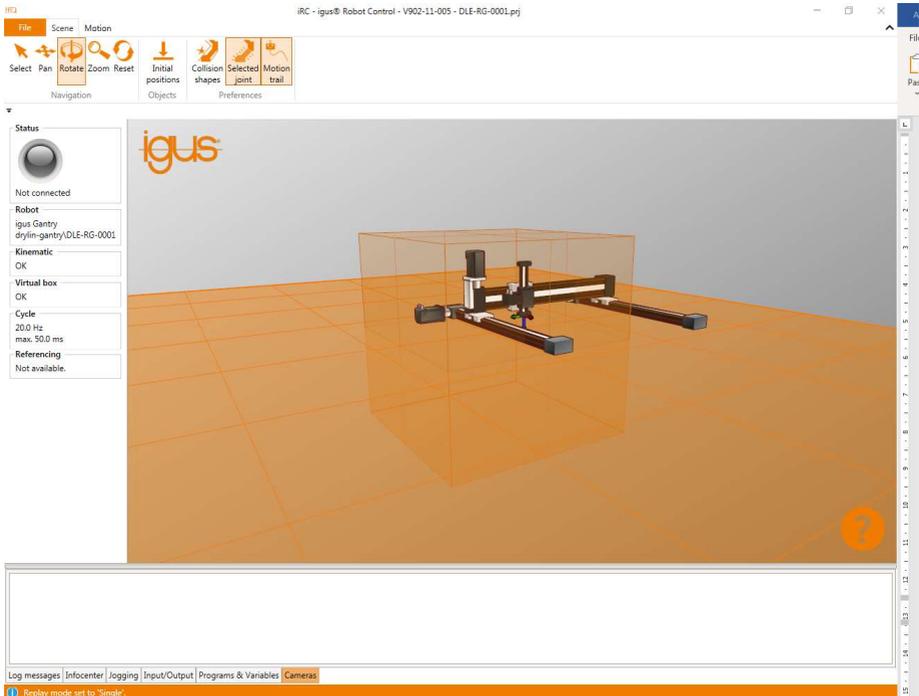


Abb. 10.4: Roboterarm, der die virtuelle Box verletzt.

Im manuellen Jogging-Modus erscheint eine Fehlermeldung auf der linken Seite des iRC-Fensters.

Wenn ein Programm ausgeführt wird, das gegen die virtuelle Box verstößt, wird das Programm mit einer Fehlermeldung gestoppt.

11. Robot Configuration

Die Portalroboter werden mit einem Standardsatz von Parametern für den Linearschienentyp und die Länge der verschiedenen Achsen geliefert.

In der Realität werden die meisten Portale hinsichtlich der Achslänge kundenspezifisch gefertigt. Diese Achslänge kann im Abschnitt Roboterkonfiguration eingestellt werden.

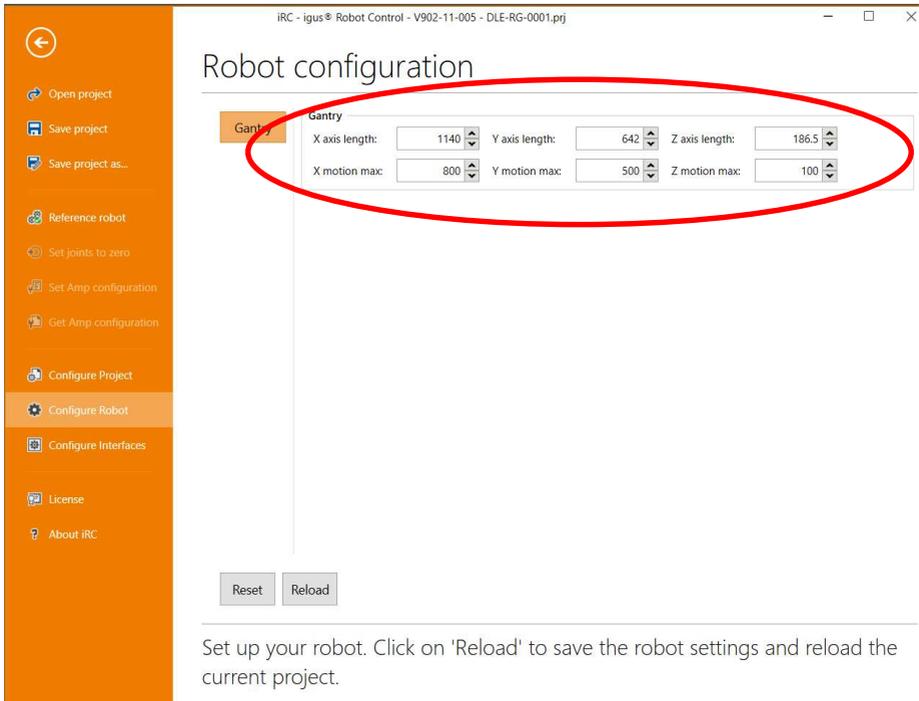


Abb. 11.1: Konfiguration der Achslängen

In der Konfiguration können sowohl die Achslänge als auch der Bewegungsbereich konfiguriert werden:

- Achslänge: hauptsächlich zur Visualisierung verwendet
- Bewegungsbereich: Dient zur Einschränkung der Bewegung um Kollisionen zu vermeiden

Der Bewegungsbereich muss entsprechend dem spezifischen Hardware-Setup eingestellt werden. Er ist abhängig von der Achslänge, aber auch von der Position des Referenz Pins und von zusätzlichen Komponenten wie den Energieketten.

12. Konfiguration der Schnittstellen

12.1. Kameras

Die iRC-Software verfügt über integrierte Unterstützung für den Empfang von Zielpositionen von Kameras. Derzeit wird das IFM O2D unterstützt.

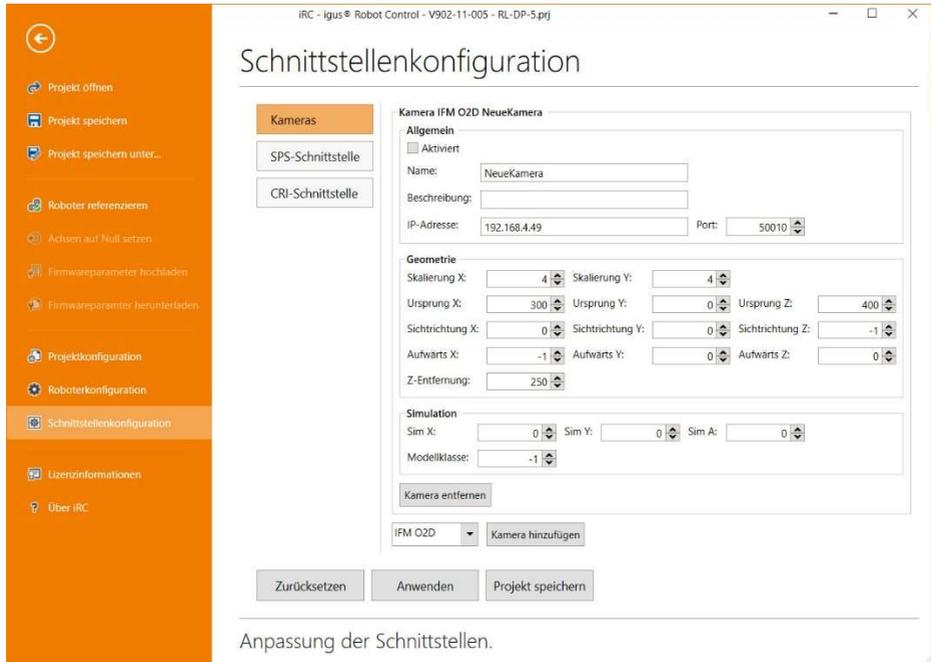


Abb.12.1: Kamera-Konfigurationsdialog.

Auf die von der Kamera gemeldete Werkstückposition kann mit dem Befehl „Kamera“ im Roboterprogramm zugegriffen werden, siehe Abschnitt 8.7.

12.2. PLC-Schnittstelle

Die PLC-Schnittstelle (PLC = Programmable Logic Controller) ermöglicht die Integration des Roboters in ein von einer PLC gesteuertes Produktionssystem. Auf diese Weise kann der Roboter ohne manuelle Interaktion arbeiten. Außerdem ist es möglich, Bedienfelder von Drittanbietern mit Drucktasten zu verbinden, um den Betrieb des Roboters in einer Produktionsumgebung zu vereinfachen oder einzuschränken.

Konkret weist diese Schnittstelle bestimmten digitalen Ein-Ausgangs-Kanälen spezielle Funktionen zu, wie z.B:

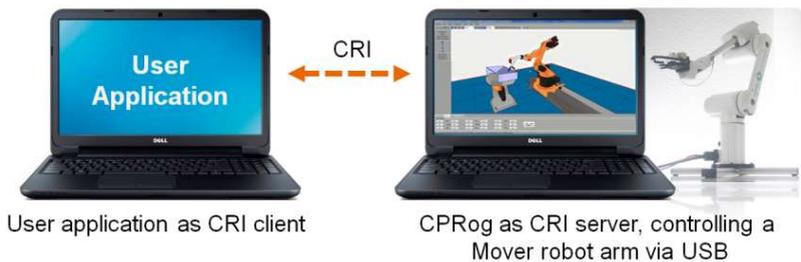
- Eingaben: „Motoren rücksetzen/aktivieren“, „Start-/Stopp-Programm“, „Referenzieren“
- Ausgaben: „noFault“, „aktiviert“ und „Programm läuft“.

Weitere Einzelheiten sind im Wiki verfügbar (Abschnitt „PLC-Schnittstelle“, <http://wiki.cpr-robots.com/>).

12.3. CRI-Ethernet-Schnittstelle

Das Commonplace Robotics Interface (CRI) ermöglicht die Steuerung des Roboters über Ethernet.

Der Roboter kann bewegt werden, Programme können hochgeladen und ausgeführt werden. Er ermöglicht die Entwicklung einer kundenspezifischen Robotersteuerungssoftware. Die Steuerungssoftware (TinyCtrl) auf dem integrierten Computer und die igus® Robot Control auf dem Windows-PC kommunizieren über die CRI-Schnittstelle.



Diese Schnittstelle ermöglicht es, die igus® Robot Control Funktionen mit anwendungsspezifischen Algorithmen, wie z.B. einem Teleoperationssystem oder einer Datenbank, zu kombinieren. Die CRI-Dokumentation und ein C#-Beispielprojekt für einen Kunden können auf Anfrage heruntergeladen werden.

13. Fehlerbehebung

13.1. Support Kontakt

Bei Problemen helfen wir gerne!

- igus Support-Landingpage: <https://www.igus.de/info/robotics-low-cost-robotics>
- Telefon: +49 (0) 2203-96498-255
- E-Mail: ww-robot-control@igus.net

Bitte beschreiben Sie das Problem kurz und senden Sie uns die Datei „logMessages.log“ aus dem Ordner C:\iRC-igusRobotControl\.

13.2. Online-Tool - Fehleridentifizierung und -behebung

Über die igus Support-Landingpage haben Sie Zugang zu Webseiten, die folgendes erlauben:

- ➔ Identifizieren Sie ein Problem auf der Grundlage von Benutzerbeobachtungen
- ➔ Erhalten Sie Lösungsvorschläge für Ihr Problem

13.3. Konfiguration der Stepper-Module

Die Betriebsparameter der Schrittmotoren, insbesondere die Motorströme, können an die Anwendung angepasst werden. Die Schrittmotoren werden mit einer Standardparametrierung für den spezifischen Roboter ausgeliefert. Normalerweise ist keine Änderung der Parameter notwendig. Wenn der Roboter bei hohen Lasten oder Geschwindigkeiten betrieben werden soll, können die Motorströme erhöht werden.

Diese Konfiguration kann über das Software-Tool „CPR ModuleCtrl“ vorgenommen werden. Es kann vom Wiki (<http://wiki.cpr-robots.com/>) heruntergeladen werden.

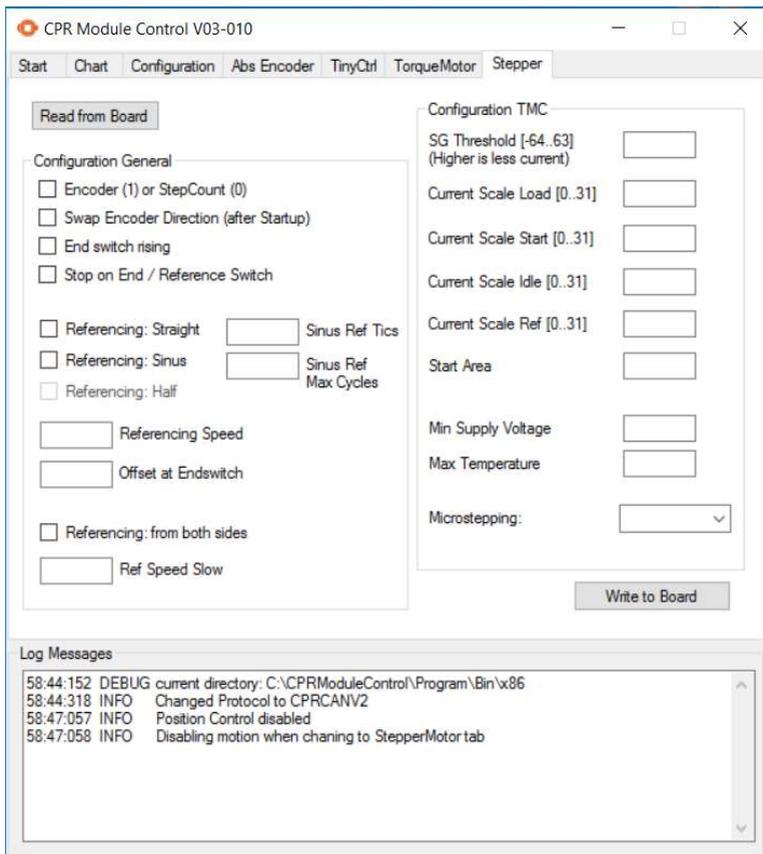


Abb. 13.1. CPR ModuleCtrl-Software, Registerkarte Schrittmotor-Konfiguration.

Abb. 13.1 zeigt die Konfiguration der Schrittmotoren. Das Verhalten während der Referenzierung ist auf der linken Seite definiert. Motorströme und Microstepping sind auf der rechten Seite definiert.

- Die „Current Scale“ Parameter definieren den maximalen Strom im Normalbetrieb (Load), während der Anlaufphase (Start), im Stillstand (Idle) und während der Referenzierung (Ref).
 - Das Schrittmodul verwendet die Trinamic CoolStep-Technologie, um den Motorstrom zwischen diesen Werten und einem unteren Grenzwert, der ein Bruchteil des Current Scale-Wertes ist, einzustellen.
 - Bei niedrigen Werten sind der Gesamtstromverbrauch und die Wärmeerzeugung geringer. Es kann jedoch ein sogenannter "Motorblock" auftreten. In diesem Fall müssen die Werte erhöht werden.
 - Bei höheren Werten gibt es keinen Motorstillstand, sondern das System wird heißer.
- Die „Current Scale“ während der Referenzbewegung kann separat eingestellt werden, um Schäden, z.B. durch Kollisionen, zu vermeiden.
- Je höher das „Microstepping“ eingestellt ist, desto weicher wird die Bewegung. Im Betrieb ohne Encoder kann Microstepping von max. 1:64 verwendet werden.
- Es ist möglich, die Drehrichtung des Encoders umzukehren. Diese Änderung wird nach dem nächsten Kaltstart des Controllers vorgenommen. Wenn die Drehrichtung des Encoders nicht mit der Motorrichtung übereinstimmt, bewegt sich die Achse nicht korrekt, sondern beschleunigt, bis ein Positionsfehler auftritt. In diesem Fall muss die Drehrichtung des Encoders geändert und das System neu gestartet werden. Die Achse sollte dann ordnungsgemäß funktionieren.

13.4. Kalibrierung des Roboters

Die robolink® Roboterarme werden über Referenzschalter referenziert. Aus konstruktiven Gründen liegen diese jedoch nicht genau auf den Nullpositionen der Achsen. Der Offset zwischen Referenzschalter und Nullposition wird im EEPROM der Schrittmodule gespeichert.

Wenn der Roboter mit der Robotersteuerung geliefert wurde, sind diese Offsets bereits eingestellt.

Um die Kalibrierung selbst durchzuführen, finden Sie die notwendigen Schritte im Wiki (Abschnitt „Define the Zero Position Offsets“, <http://wiki.cpr-robots.com/>).

13.5. Fehler-Codes

Die Robotersteuerung stellt verschiedene Statusinformationen zur Verfügung:

- Status-LEDs auf den elektronischen Modulen.
- igus® Robot Control Statusinformationen, empfangen über CAN-Statusbytes.

13.5.1. Status-LEDs der elektronischen Module



Support-Modul:

Grüne LED an: Logik-Stromversorgung an

Orange LED an: Fehler
Rote LED leuchtet: Not-Aus-Schalter gedrückt



Stepper-Modul:

Grüne LED an: Logik-Stromversorgung an
Grüne LED blinkt: CAN-Kommunikation mit dem Steppermodul
Orange-LED an: Referenzschalter ist aktiv
Rote LED an: Schrittmotor ist im Fehlerzustand oder der Motor ist nicht aktiviert



Digitales Ein-/Ausgangsmodul:

Grüne LED an: Logik-Stromversorgung an
Grüne LED blinkt: CAN-Kommunikation mit dem DIO-Modul
Orange LED an: Der Zustand eines Ein- oder Ausgangs ändert sich.
Rote LED leuchtet: Fehler

13.5.2. CAN-Bus und iRC-Statusinformationen

Fehler	Bit im Fehlerbyte	Bedeutung	Maßnahmen
Bus dead		Der CAN-Bus ist nicht verfügbar. Gründe sind fehlende Stromversorgung oder fehlende Steckverbindungen.	Überprüfen Sie die Steckverbindungen der Stromversorgung und der CAN-Leitung. Starten Sie den Steuerrechner neu.
Temp	Bit 1	Die Temperatur der Motormodule ist zu hoch.	Überprüfen Sie, ob die Belüftung installiert ist und funktioniert. Möglicherweise muss der Motorstrom reduziert werden.
E-Stop/ Supply	Bit 2	Not-Aus-Schalter gedrückt oder zu niedrige Spannung.	Prüfen Sie, ob der Not-Aus-Schalter freigegeben ist.
MNE Motor not enabled	Bit 3	Kein Fehler. Die Motoren sind noch nicht freigegeben.	Drücken Sie die Schaltfläche „Aktivieren“.
COM Comm Watch Dog	Bit 4	Die Zeitspanne ohne CAN-Befehl vom Controller war zu lang.	Die Positionsbefehle über den CAN-Bus müssen in kurzen Abständen gesendet werden. Schließen Sie andere Programme und Update-/Virensan-Prozesse
LAG Position Lag	Bit 5	Positionsschleppfehler. Der Roboter kann die Zielposition nicht halten.	Verringern Sie die Geschwindigkeit der Bewegung.
ENC Encoder Fehler	Bit 6	Fehler im Motor-Encoder oder Absolut-Encoder	Überprüfen Sie die Encoderkabel
OC Over Current	Bit 7	Überstrom in den Motoren	Reduzieren Sie den Motorstrom
DRV	Bit 8	Fehler im Motortreiber oder Motor-Algorithmus	Treiberspezifisch

Nach einem „Fehler-Reset“ ist der normale Status der Achsen 0x04 (Motor nicht aktiviert). Nach dem Freigeben der Motoren ist der Status 0x00, jetzt sind die Achsen betriebsbereit.

