RAINBOW ROBOTICS

RB SERIE

Bedienungsanleitung

Deutsch

V3.9.1

Aktualisierung : 2021/09

© RAINBOW ROBOTICS Inc. All rights reserved.

Inhaltverzeichnis

Vorwort **6**

Kapitel 1 Produktbeschreibung **7**
1.1 Kollaboratives Robotersystem 7
1.2 Systemkonfiguration 8
1.3 Roboter-Arm 10
1.4 Steuerbox 17
1.5 Tablet-PC(optional) ..... **오류! 책갈피가 정의되어 있지 않습니다.**
1.6 Reichweite des Roboterarms 19
1.7 Betriebsbereich des Roboterarms 20
1.8 Maximale Nutzlast des Roboterarms 21

Kapitel 2 Sicherheits- und Vorsichtsmaßnahme **22**
2.1 Sicherheitshinweise in der Bedienungsanleitung 22
2.2 Allgemeine Sicherheitshinweise und Vorsichtsmaßnahme 23
2.3 Anwendungszweck 26
2.4 Risikopotenzial 27
2.5 Haftungsbeschränkung 28
2.6 Lieferung und Transport 29
2.7 Not-Aus 30
2.8 Benutzersicherheit 31
2.9 SICHERHEITSKONTROLLSYSTEM 32
2.10 Risiko-Assessment 33

Kapitel 3 Sicherheitsvorrichtungen **34**
3.1 Einleitung 34
3.2 Stopp-Kategorie (Stop Category) 36
3.3 Funktionssicherheit 37
3.4 Montagebereich der Sicherheitsvorrichtung 40
3.5 Not-Aus-Schalter 42
3.6 Betriebsmodus 43
3.7 Betriebsumgebung 45
3.8 Wartung von Sicherheitsfunktion 46
3.9 Angewandte Norm 48

Kapitel 4 Installation **50**
4.1 Hinweise zur Installation 50
4.2 Installationsort 51
4.3 Installationstyp 52
4.4 Befestigung des Roboterarms 53
4.5 Verbindung des Roboters mit dem Werkzeug 54
4.6 Kabelanschluss 59
4.7 Übersicht über Ein/Aus der Robotersteuerbox 61

© RAINBOW ROBOTICS Inc. All rights reserved.

4.8 Konfiguration der Sicherheitseingänge 62
4.9 Universelle Digitale Ein/Aus-Konfiguration 65
4.10 Universelle Analoge Ein/Aus-Konfiguration 67

Kapitel 5 Starten 69
5.1 Ein/Aus der Steuerbox 69
5.2 Tablet-PC einschalten 70

Kapitel 6 Übersicht über Software 71
6.1 Grundkonfiguration des UI-Programms 71
6.2 Startbildschirm 72
6.3 Hauptfenster 73
6.4 Betriebsbildschirm (MAKE) 75
6.5 Wiedergabe-Bildschirm (PLAY) 76
6.6 Einstellung (SETUP) 77

Kapitel 7 Programmieranleitung 78
7.1 Beschreibung von Symbolen und Bedienbildschirm 78
7.2 Teaching-Umgebung schaffen 93
7.3 Teaching (Programmieren) 97
7.4 Detaillierte Beschreibung der wichtigsten Teaching-Symbole 142
7.5 Programmbearbeitung 307
7.6 Programmverwaltung 312
7.7 Betriebliche Dienstprogramme 315

Kapitel 8 Starten von Roboter 335
8.1 Antreiben von Roboter 335
8.2 Prüfung des Roboterstatus 337
8.3 Umgang mit ungewöhnlichen Ereignissen während des Betriebs 338

Kapitel 9 Einstellungen 343
9.1 Set-up (Cobot) 343
9.2 Set-up (Tool) 344
9.3 Set-up (System) 345
9.4 Set-up (Log) 346
9.5 Set-up (Utility) 347
9.6 Set-up (Serial) 349
9.7 Set-up (Ein/Aus 1) 350
9.8 Set-up (Ein/Aus 2) 356
9.9 Set-up (Inbox) 358
9.10 Set-up (Schnittstelle) 359
9.11 Set-up (Coordinate) 360
9.12 Set-up (Devices) 361
9.13 Set-up (Tool List) 362
9.14 Set-up (Program Table) 363

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel10 Wartung 365
10.1 Checkliste und Zyklus 365
10.2 Wartung des Roboterarms 366
10.3 Wartung der Robotersteuerbox 367

Anhang A Technische Spezifikationen des Systems 368
Anhang B Abmessungen der Roboterbefestigungsfläche 370
Anhang C Querschnitt von Werkzeugflansch 372
Anhang D Schaltschema von Steuerbox 374
Anhang D-1 Digitaler Eingang für Steuerbox 375
Anhang D-2 Digitaler Ausgang für Steuerbox 380
Anhang D-3 Digitaler Eingang für Werkzeugflansch 382
Anhang D-4 Digitaler Ausgang für Werkzeugflansch 385
Anhang E Externe Skriptsteuerungs-API 389
Anhang F Einstellung von Koordinatensystem 423
Anhang G Stoppzeit und Stoppdistanz 424
Anhang H Beschriftung 425
Anhang I Modbus-Server 426
Anhang J System Update (Systemaktualisierung) 433
Anhang K Android-Tablet-Einstellungen 437
Anhang L Mechanische Bremssystem (Brake System) 440

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorwort

Lesen Sie diese Bedienungsanleitung vor der Installation des Produkts und befolgen Sie die Anweisungen der Anleitung entsprechend dem Installationsvorgang. Der Inhalt dieser Bedienungsanleitung basiert auf dem Zustand des Produkts zum Zeitpunkt der Erstellung der Bedienungsanleitung, und Produktinformationen können ohne Ankündigung an den Benutzer geändert werden.

Wenn Sie sich bezüglich der in dieser Bedienungsanleitung angegebenen Anforderungen, Empfehlungen oder Sicherheitsverfahren nicht sicher sind, wenden Sie sich bitte an Rainbow Robotics Inc. Ein Teil der Abbildungen in diesem Handbuch dienen zum Verständnis des Konzepts und der Installation des Systems und können vom tatsächlichen Produkt abweichen.

Die Urheberrechte und die Rechte des geistigen Eigentums für alle Inhalte und Designs in dieser Bedienungsanleitung gehören Rainbow Robotics Inc. Daher ist das Verwenden, Kopieren oder Verteilen ohne schriftliche Genehmigung von Rainbow Robotics Inc. strengstens verboten und eine Verletzung der Rechte des geistigen Eigentums von Rainbow Robotics Inc.

Für jeden Missbrauch oder jede Änderung der Patente auf diesem Produkt ist der Benutzer allein verantwortlich. Die in dieser Bedienungsanleitung enthaltenen Informationen sind zuverlässig.

Die in dieser Bedienungsanleitung angegebenen Informationen sind Eigentum von Rainbow Robotics Inc. und dürfen ohne Zustimmung weder ganz noch teilweise reproduziert werden. Der Inhalt kann ohne vorherige Ankündigung geändert werden, und Sie finden detaillierte Informationen zur Überarbeitung der Bedienungsanleitung auf der Webseite (www.rainbow-robotics.com).

ⓒ Rainbow Robotics Inc. All rights reserved

Kapitel 1 Produktbeschreibung

1.1 Kollaboratives Robotersystem

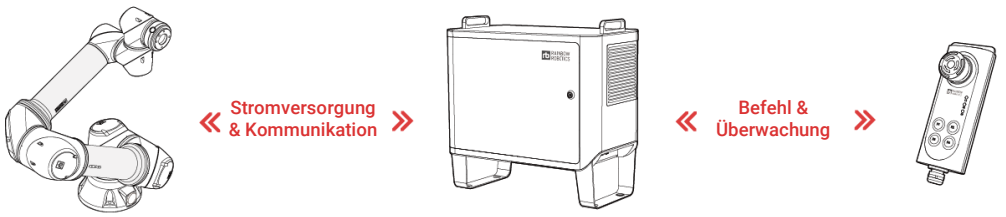
Der kollaborative Roboter der RB-Serie von Rainbow Robotics Inc. ist ein Roboter, den jeder einfach betreiben kann. Der ist die beste Kollaborationslösung, die die Produktionseffizienz maximieren kann, ohne eine separate Sicherheitsvorrichtung zu verwenden, und ist spezialisiert auf die sichere Zusammenarbeit bei sich wiederholenden Aufgaben im engsten Raum in der Nähe von menschlichen Mitarbeitern in verschiedenen Bereichen.

- Anschauliche Bedienbarkeit: RB-Serie besteht aus einer grafikbasierten anschaulichen Benutzeroberfläche (UI, User Interface), sodass nicht nur Experten, sondern auch jeder sie einfach einrichten und bedienen kann.
- Komfort und Sicherheit: Eine sichere Arbeitsumgebung wird geboten, indem Unfälle und Verletzungen während des Betriebs durch externe und eigene Kollisionserkennungssysteme minimiert werden.
- Raumeffizienz: Die RB Serie kann unabhängig von begrenzten Platzverhältnissen auf jede Produktionslinie angewendet werden, da das vielseitige Konfigurationen bietet, die eine Installation in jedem Raum ermöglichen.

© RAINBOW ROBOTICS Inc. All rights reserved.

1.2 Systemkonfiguration

Die Konfiguration des Robotersystems sieht wie folgt aus:



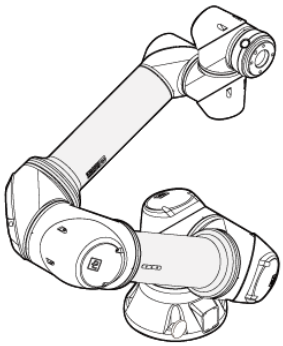
[Konfiguration des Standsteuerbox systems]

- Roboterarm: Es handelt sich um einen kollaborativen Industrieroboter, der für sich wiederholende einfache Aufgaben wie den Transport kleiner Gegenstände oder die Montage von Teilen verwendet werden kann und durch das Anbringen Robotergreifer und verschiedener Werkzeuge verwendet werden kann.
- Steuerbox: Die Bewegung des Roboterarms wird von einem von Benutzer eingestellten Programm gesteuert. Ausgestattet mit digitalen und analogen Ein-/Ausgangsanschlüssen können verschiedene Geräte und Vorrichtungen angeschlossen und verwendet werden.
- Not-Aus/Jog-Schnittstelle(Nur für Standsteuerbox): Mit einem Not-Aus-Schalter kann der Roboter gestoppt werden. Es gibt Tasten für die einfache Programmablaufsteuer wie Ein/Aus-Taste.
- Tablet-PC(optional): Das ist ein externes Gerät zur Systembedienung, das dem Roboter eine bestimmte Haltung beibringen oder verschiedene Steuerungen und Einstellungen des Roboters vornehmen kann.

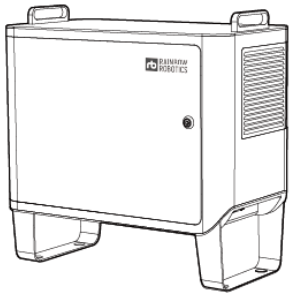
© RAINBOW ROBOTICS Inc. All rights reserved.

Die mitgelieferten Systemkomponenten und die Menge jeweiliger Komponente sind wie folgt:

- Komponenten der Standsteuerbox



[Roboterarm]



[Steuerbox]

[Not-Aus/Jog-Schnittstelle]

[Bedienungsanleitung]

[Tablet-PC]

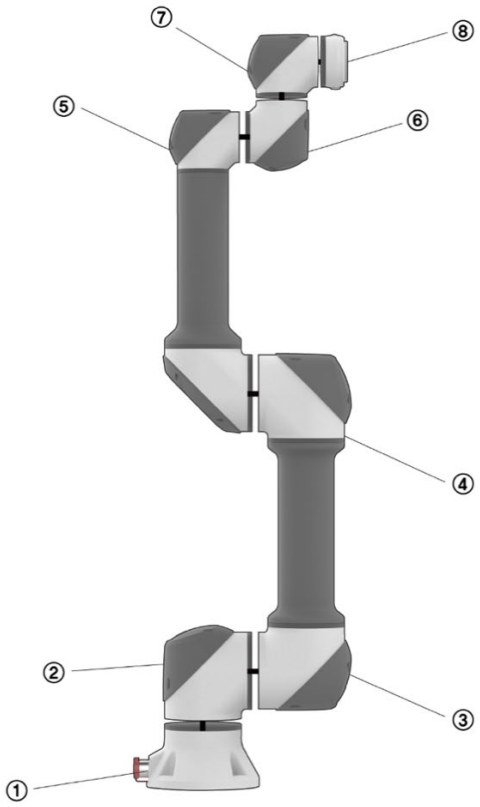
Roboter	Roboterarm	1 EA
	Steuerbox	1 EA
	Not-Aus/Jog-Schnittstelle	1 EA
Systemkomponenten	Netzkabel	1 EA
	Anschlusskabel Roboterarm zu Steuerbox	1 EA
	Optionsartikel-Tablet-PC, Schutzhülle, USB-Kabel	1 EA
Andere Komponenten	Bedienungsanleitung(elektronisches Dokument)	1 EA

※ Es wird empfohlen, das vom Hersteller mitgelieferte 5m-Kabel für das Netzkabel, das Not-Aus/Jog-Schnittstellenkabel und das Anschlusskabel Roboterarm zu Steuerbox zu verwenden. Für LAN-Abschirmkabel, Ein/Aus-Anschlusskabel, USB-Kabel und externe Verlängerungskabel für die Modelle mit elektrischer Leitung(Electic Line Passing Model) wird eine Länge von weniger als 3 Metern empfohlen.

© RAINBOW ROBOTICS Inc. All rights reserved.

1.3 Roboter-Arm

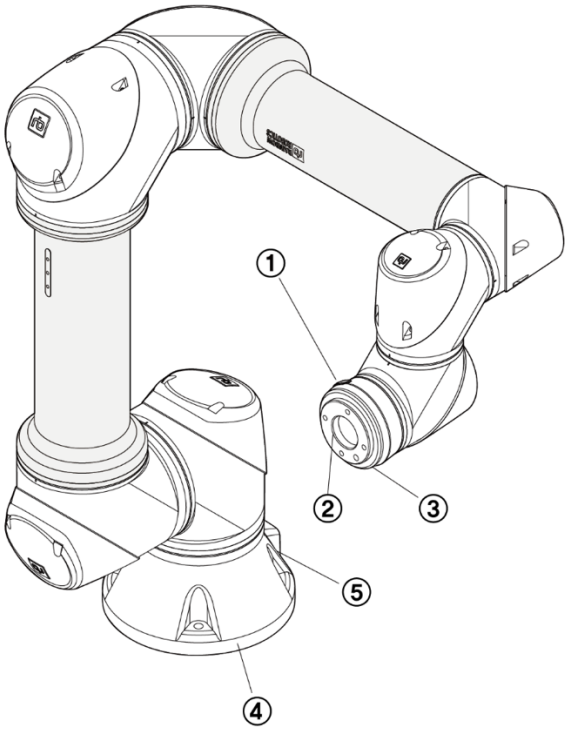
- Beschreibung der Gelenke der RB-Serie



Nr.	Name
①	Basis (Base)
②	Basisgelenk (Base Joint)
③	Schultergelenk (Shoulder Joint)
④	Ellenbogengelenk (Elbow Joint)
⑤	1. Handgelenk (Wrist 1 Joint)
⑥	2. Handgelenk (Wrist 2 Joint)
⑦	3. Handgelenk (Wrist 3 Joint)
⑧	Werkzeugflansch (Tool Flange)

© RAINBOW ROBOTICS Inc. All rights reserved.

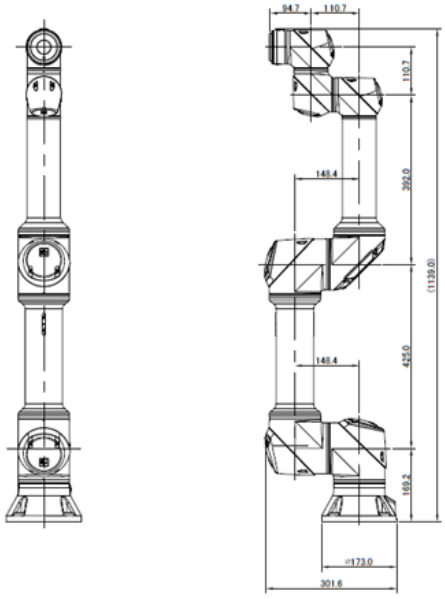
■ Namen der Hauptteile



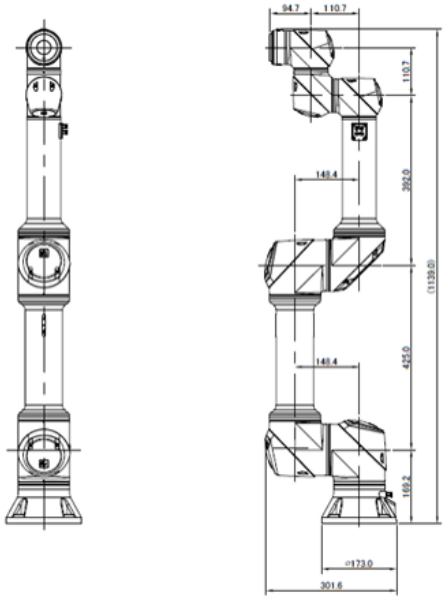
Nr.	Name	Beschreibung
①	Teaching-Taste	Taste für Direkt-Teaching
②	Werkzeugflansch	Teil zur Montage von Greifer oder Werkzeug
③	Werkzeug Ein/Aus	Ein/Aus-Anschluss zur Steuerung von Greifer oder Werkzeug
④	Base	Teil zur Befestigung des Roboterarms
⑤	Anschluss Roboter zu Steuerbox	Kabelanschluss zwischen Roboterarm und Steuerbox

© RAINBOW ROBOTICS Inc. All rights reserved.

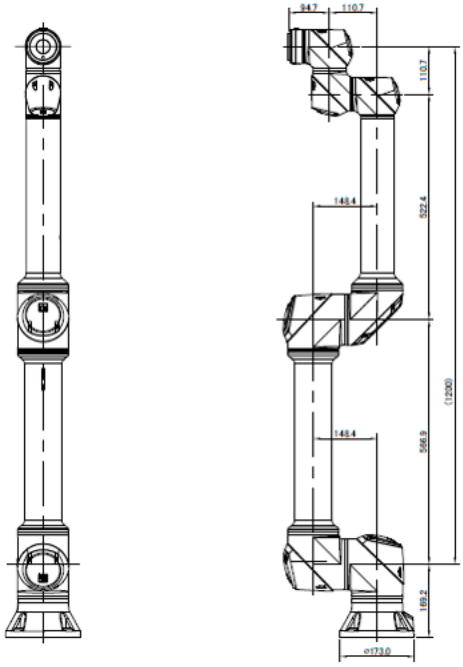
■ Form und Abmessungen des Roboterarms



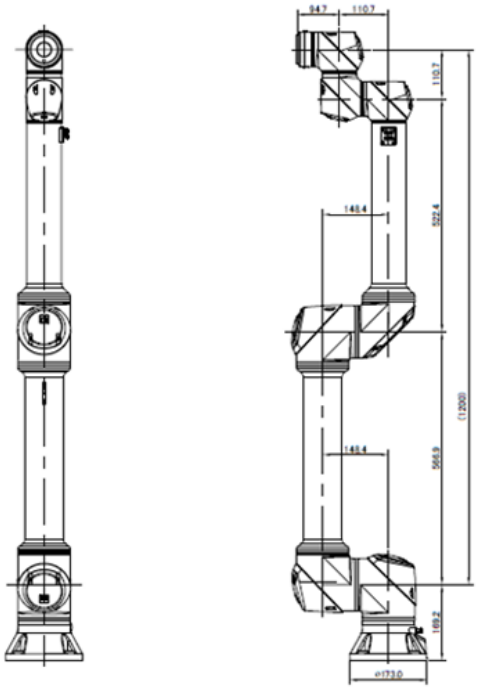
[RB5-850E]



[RB5-850EA1/A2]



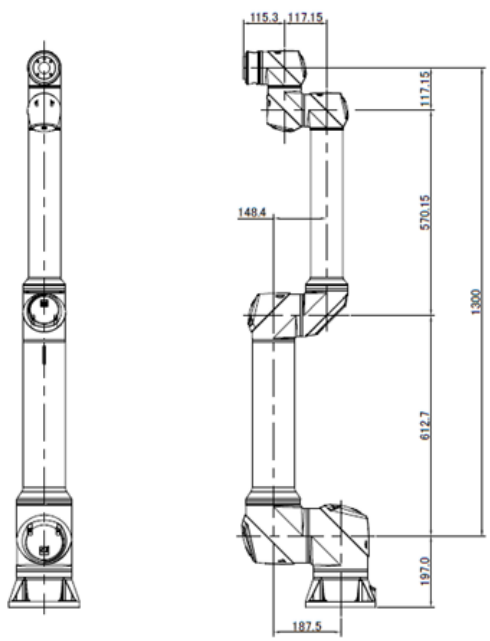
[RB3-1200E]



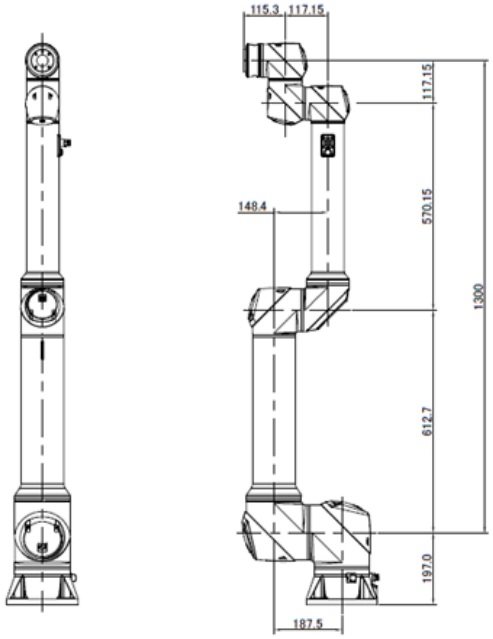
[RB3-1200EA1/A2]

© RAINBOW ROBOTICS Inc. All rights reserved.

- Form und Abmessungen des Roboterarms



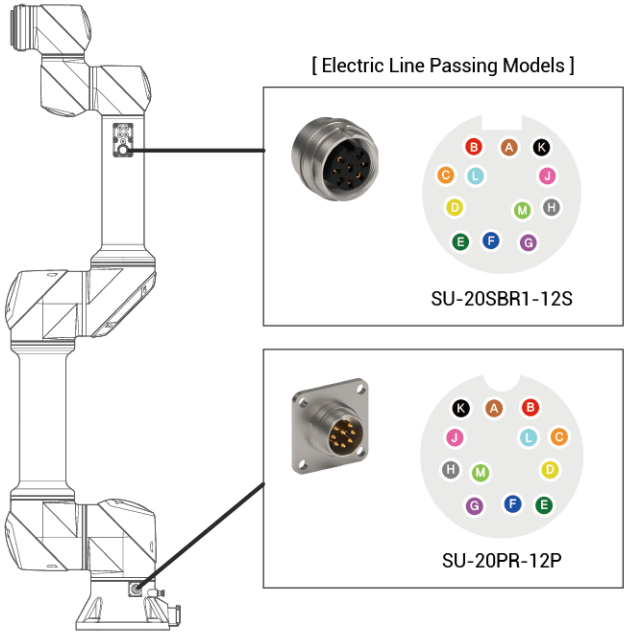
[RB10-1300E]



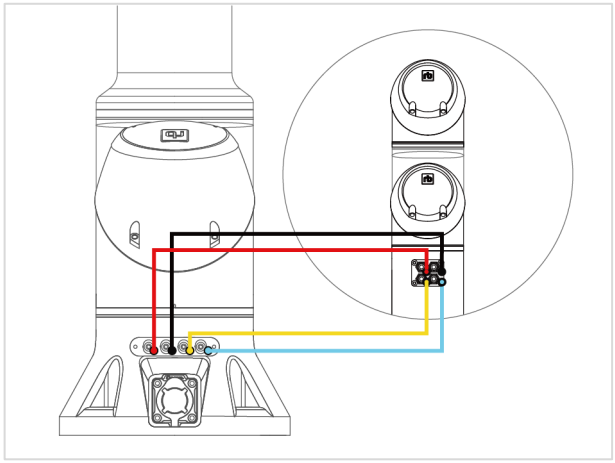
[RB10-1300EA1/A2]

© RAINBOW ROBOTICS Inc. All rights reserved.

- Verwendung von RB5-850EA#, RB3-1200EA#, RB10-1300EA# (Pneumatik-Schlauch/Elektrik-Kabel eingebaute Type)

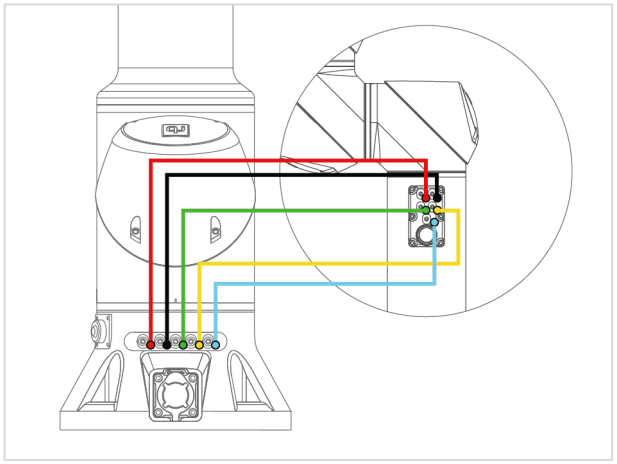


[Anschluss eingebauten pneumatischen Schlauchs]

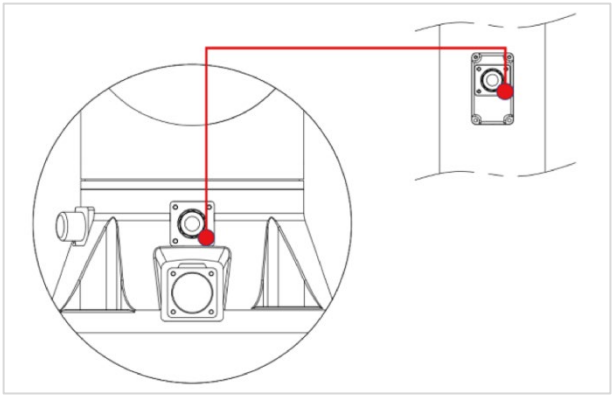


[RB5-850EA1, RB3-1200EA1]

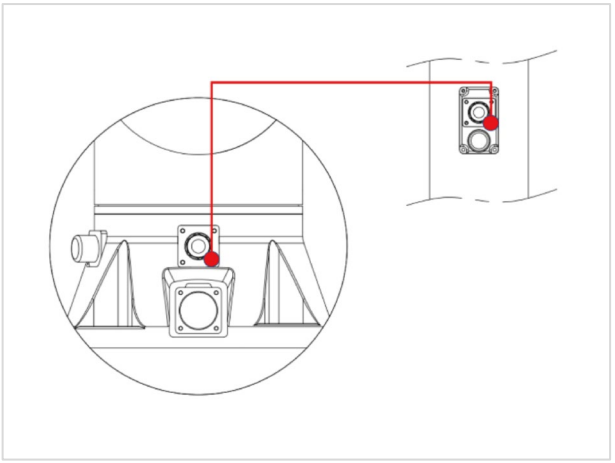
© RAINBOW ROBOTICS Inc. All rights reserved.



[RB5-850EA2, RB3-1200EA2]



[RB10-1300EA1]



[RB10-1300EA2]

© RAINBOW ROBOTICS Inc. All rights reserved.

※ Die pneumatischen Leitungen und elektrischen Leitungen von RB5-850EA#, RB3-1200EA#, RB10-1300EA# werden wie in der Tabelle unten bereitgestellt. Bitte beachten Sie das Bild oben zur Verwendung der Leitungen.

Modellname	Pneumatische Leitung	Elektrische Leitung
RB5-850EA1	Bis zu 4 (4Ø Pneumatikschlauch)*	k. a.
RB5-850EA2	Bis zu 5 (4Ø Pneumatikschlauch)*	12-Pin (AWG28)
RB3-1200EA1	Bis zu 4 (4Ø Pneumatikschlauch)*	k. a.
RB3-1200EA2	Bis zu 5 (4Ø Pneumatikschlauch)*	12-Pin (AWG28)
RB10-1300EA1	1 (8Ø Pneumatikschlauch)	k. a.
RB10-1300EA2	1 (8Ø Pneumatikschlauch)	12-Pin (AWG28)

* Die Anzahl der pneumatischen Leitungen muss nach Reichweite angepasst werden.

Vorsicht:

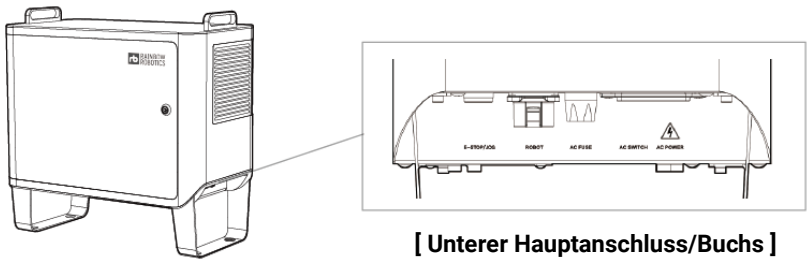
1) Bei Modellen mit eingebauten pneumatischen/elektrischen Leitungen kann die Hardware beschädigt werden, wenn die Pneumatik- oder Stromversorgung die definierten Spezifikationen überschreitet.

© RAINBOW ROBOTICS Inc. All rights reserved.

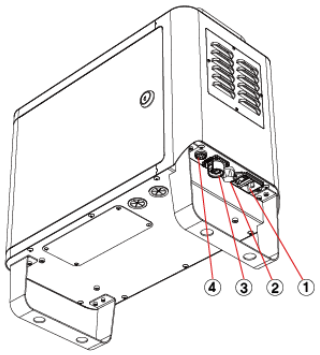
1.4 Steuerbox

Die Vorderseite, die Unterseite und das Innere von Steuerbox sind wie folgt:

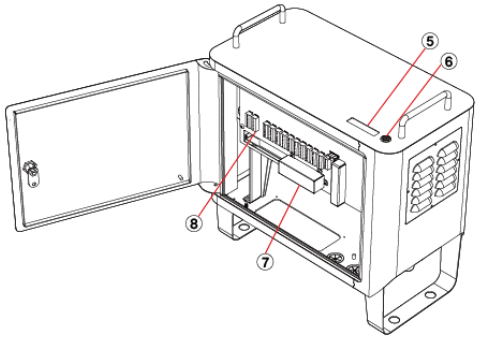
- Standsteuerbox



[Unterer Hauptanschluss/Buchs]



[Untersicht]



[Ansicht von oben]

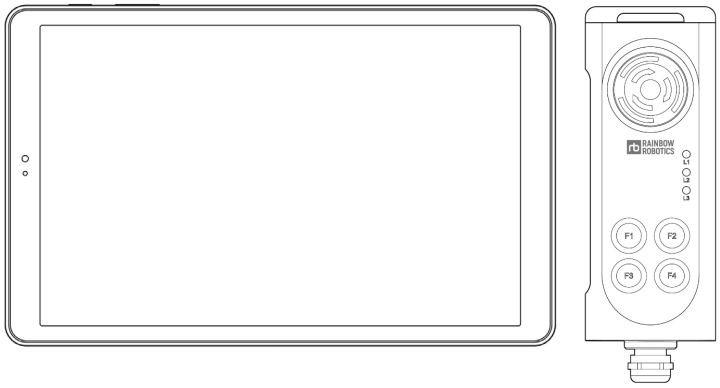
Nr.	Name
①	Wechselstromsteckdose (AC POWER)
②	Wechselstromschalter (AC SWITCH)
③	Kabelanschluss des Roboterarm (ROBOT)
④	Anschluss von Handprogrammiergerät (EMO/TEACH PENDANT)
⑤	LCD
⑥	Hauptschalter
⑦	USB/LAN-Anschluss
⑧	Ein/Aus-Anschluss

© RAINBOW ROBOTICS Inc. All rights reserved.

1.5 Tablet-PC(optional)

Das Tablet-PC ist ein optionales Zubehör. Das muss separat gekauft werden.

- Für Standsteuerbox



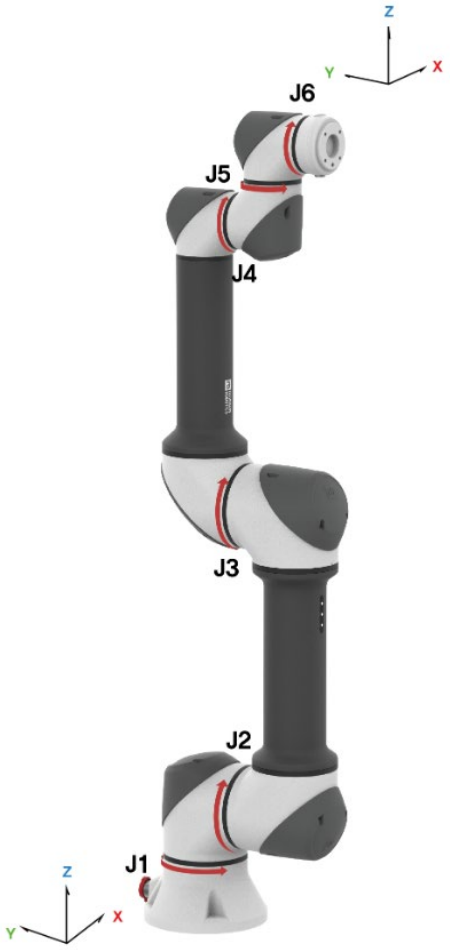
※ Wenn Sie Tablet-PC als Option nicht wählen, wird eine spezielle Anwendung zum Antreiben, Bedienen und Programmieren des Roboterarms angeboten (Die App kann von der Webseite von Rainbow Robotics herunterladen werden.).

※ Einstellungen von Tablet-PC für die Verwendung mit unseren Produkten sind erforderlich. Siehe Anhang.

© RAINBOW ROBOTICS Inc. All rights reserved.

1.6 Reichweite des Roboterarms

Der RB-Roboterarm besteht aus 6 Achsen und die Reichweite jeder Gelenkachse ist wie folgt.

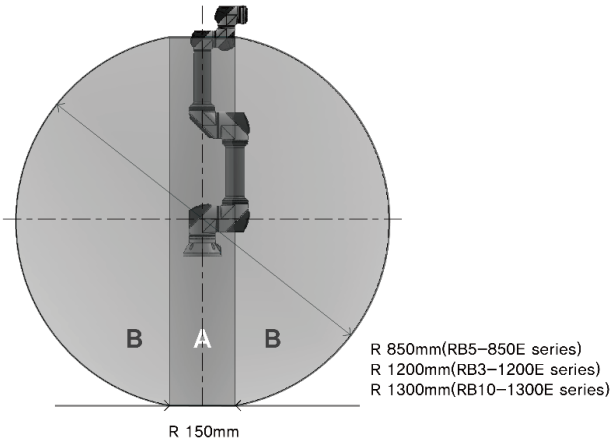


Gelenk (Joint)	Reichweite
J 1	± 360 °
J 2	± 360 °
J 3	± 165 °
J 4	± 360 °
J 5	± 360 °
J 6	± 360 °

© RAINBOW ROBOTICS Inc. All rights reserved.

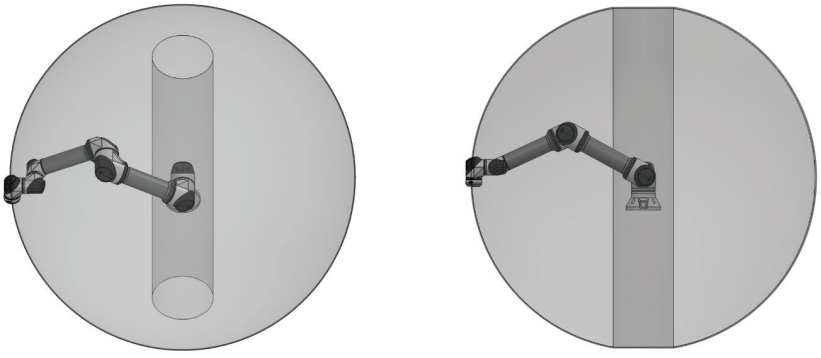
1.7 Betriebsbereich des Roboterarms

Der maximale Wenderadius des Roboterarms beträgt 850 mm für die RB5-850E Serie, 1.300 mm für die RB10-1300E Serie und 1.200 mm für RB3-1200E Serie. Direkt unter oder direkt über der Roboterbasis (Bereich A in der Abbildung) befindet sich der eingeschränkte Betriebsbereich. Anders als im gemeinsamen Koordinatensystem (z. B. Move J) ist der Bereich bei Bewegungen im inertialen Koordinatensystem (z. B. Move L) eingeschränkt. Er ist ein mechanisch singulärer Bereich. Wenn der Roboterarm sich unter dem inertialen Koordinatensystem bewegt, kann eine schnelle Gelenkgeschwindigkeit erzeugt werden oder kann er stoppen.



A. Singulärer Punktbereich (in dem die Bewegung im kartesischen Koordinatensystem eingeschränkt ist)

B. Betriebsbereich des kollaborativen Roboters



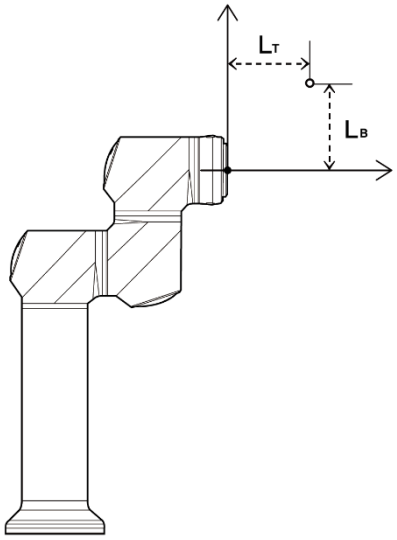
Tilt View

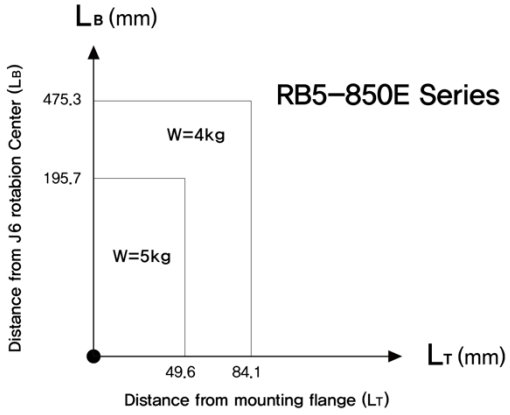
Front View

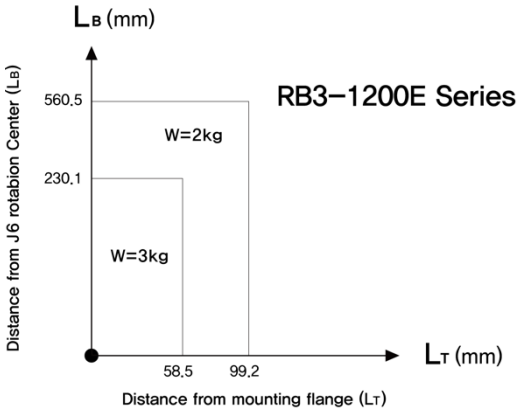
© RAINBOW ROBOTICS Inc. All rights reserved.

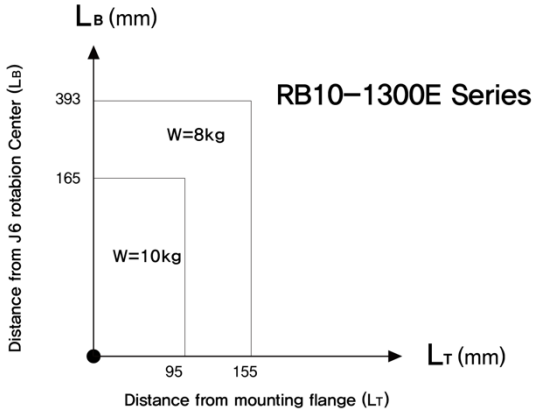
1.8 Maximale Nutzlast des Roboterarms

Die Nutzlast des Roboterarms hängt vom Abstand zwischen dem Werkzeugflansch und dem Schwerpunkt der Nutzlast ab. Die Nutzlast nach Abstand ist wie folgt.









© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 2 Sicherheits- und Vorsichtsmaßnahme

2.1 Sicherheitshinweise in der Bedienungsanleitung

Die folgenden Sicherheitszeichen werden in dieser Bedienungsanleitung verwendet.

Gefahr:

Die Nichtbeachtung der mit diesem Zeichen gekennzeichneten Anweisungen kann zu schweren Unfällen führen, die zum Tod oder zu schweren Verletzungen des Bedieners oder Benutzers führen können.

Warnung:

Die Nichtbeachtung der mit diesem Zeichen gekennzeichneten Anweisungen kann zu einem Unfall führen, der zu schweren Verletzungen des Bedieners oder Benutzers führen kann.

Vorsicht:

Die Nichtbeachtung der mit diesem Zeichen gekennzeichneten Anweisungen kann zu Schäden am Produkt oder zu Verletzungen des Bedieners oder Benutzers führen.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.2 Allgemeine Sicherheitshinweise und Vorsichtsmaßnahme

Dieser Abschnitt enthält allgemeine Gefahren, Warnungen und Vorsichtshinweise, die möglicherweise in anderen Teilen dieser Bedienungsanleitung wiederholt erläutert werden. Es werden auch andere Warnungen und Vorsichthinweise dargestellt.

Gefahr:

1) Gemäß den Anweisungen in Kapitel 4 Installation müssen der Roboter und die elektrische Ausrüstung installiert werden.

Warnung

Warnung:

1) Der Hersteller des Anwendungssystem für den Roboter oder der Benutzer muss den Inhalt dieser Bedienungsanleitung vollständig verstehen und Schulung für Betrieb absolvieren.
2) Es muss genügend Raum vorhanden sein, damit sich der Roboterarm frei bewegen kann.
3) Tragen Sie Keine weite Kleidung oder Schmuck, wenn Sie den Roboter verwenden. Binden Sie lange Haare zusammen und achten Sie darauf, dass die Haare sich nicht in den Gelenken des Roboterarms verfangen.
4) Verwenden Sie keinen beschädigten Roboter.
5) Wenn ein schwerwiegender Fehler in der Software auftritt, halten Sie sofort den Roboter mit Not-Aus und wenden Sie sich an den Produktlieferanten oder an Rainbow Robotics Inc.
6) Überprüfen Sie, ob der Installationswinkel des Roboters, die Werkzeugeinstellung, Sicherheitseinstellung usw. korrekt eingegeben sind.
7) Schließen Sie keine Sicherheitsausrüstung an allgemeinen Ein/Aus-Anschluss an. Sicherheitsausrüstungen sollten nur mit sicherheitsrelevanten Ein/Aus-Anschlüssen verwendet werden.
8) Achten Sie auf die Bewegung des Roboterarms, während Sie den Tablet-PC zu Teaching verwenden.
9) Betreten Sie nicht den Betriebsbereich des Roboters und berühren Sie den Roboter nicht, während Sie den Roboter bedienen.

© RAINBOW ROBOTICS Inc. All rights reserved.

10) Modifizieren Sie nie den Roboter. Rainbow Robotics Inc. (nachfolgend ‘Hersteller‘ genannt) ist nicht verantwortlich für Probleme, die durch willkürliche Änderungen oder Modifizierung von dem Benutzer verursacht werden.
11) Bei längerem Gebrauch erzeugen der Roboterarm und die Steuerung Wärme. Berühren Sie den Roboter nach längerem Gebrauch nicht mit den Händen. Schalten Sie die Steuerung aus und lassen Sie den Roboter ausreichend abkühlen, wenn Sie den Roboter berühren müssen.
12) Wenn der Roboter mit einem externen Objekt kollidiert, wird eine beträchtliche kinetische Energie erzeugt, die proportional zu seiner Geschwindigkeit und Nutzlast ist.
13) Stellen Sie sicher, dass die empfohlenen Installationseinstellungen verwendet werden. Wenn die Installationsrichtung des Roboterarms, das Gewicht und der Schwerpunkt des Werkzeugs, die Länge und die Sicherheitskonfiguration nicht korrekt eingegeben werden, funktioniert das Direkt-Teaching oder die Kollisionserkennung möglicherweise nicht richtig.
14) Die Direkt-Teaching-Funktion muss in einer sicheren Umgebung verwendet werden. Verwenden Sie die Funktion nicht, wenn scharfe oder einklemmenden Stellen an Werkzeug oder umgebenden Vorrichtungen vorhanden sind.
15) Bevor Sie die Direkt-Teaching-Funktion verwenden, geben Sie die Eingabedaten in Bezug auf das Werkzeug (Länge, Gewicht, Schwerpunkt des Werkzeugs) genau ein. Die eingegebenen Informationen, die von den Spezifikationen abweichen, führen zu Fehlfunktion bei der Direkt-Teaching-Funktion.
16) Bei der Verwendung der Direkt-Teaching-Funktion kann das Gelenk zur Sicherheit des Benutzers einen Not-Aus ausführen, wenn es über einer bestimmten Geschwindigkeit betrieben wird.

Warnung

Warnung:

1) Der Roboterarm und die Steuerbox erzeugen während des Betriebs Wärme. Berühren Sie den Roboterarm nicht während oder unmittelbar nach dem Betrieb, da andauernder Kontakt mit dem Roboterarm zu Fehlfunktionen führen kann. Bevor Sie den Roboterarm bedienen oder berühren, überprüfen Sie die Temperatur auf dem UI-Bildschirm oder schalten Sie den Roboterarm aus und lassen Sie ihn mindestens 1 Stunde ausreichend abkühlen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

Vorsicht:

1) Bei Verwendung mit einer Maschine oder einem anderen Roboter, der den Roboterarm beschädigen könnte, wird empfohlen, alle Funktionen vor der Verwendung einzeln zu testen. Der Hersteller haftet nicht für den Schäden an anderen Maschinen oder Robotern, oder am Roboterarm aufgrund von Programmierfehlern oder Roboterausfall.
2) Der Roboterarm sollte keinen starken Magnetfeldern ausgesetzt werden. Der Roboter kann dadurch beschädigt werden.

Warnung

Warnung:

1) Ein Warmschild wird angebracht, wo die Gefahr eines Stromschlags durch elektrische Geräte vorhanden ist.
2) Reißen, beschädigen oder entfernen Sie Warnschilder nicht. Seien Sie vorsichtig beim Umgang mit gekennzeichneten Teilen oder Geräten.
3) Um einen Stromschlag zu vermeiden, berühren Sie nicht die elektrischen Teile im Inneren.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.3 Anwendungszweck

Dieser Roboterarm kann zum Transportieren und Montieren von Objekten mit Werkzeugen verwendet werden. Der darf nur in der in den Spezifikationen angegebenen Umgebung verwendet werden. Ausgestattet mit einer Sicherheitsfunktion ist kollaboratives Arbeiten ohne physischen Schutzzaun möglich, aber das endgültige System, an dem der Roboter angebracht ist, muss nach einer Risiko-Assessment des gesamten Systems verwendet werden. Die Verwendung in den folgenden Verwendungszwecken und Umgebungen gilt als unsachgemäß. Der Hersteller ist nicht verantwortlich für Schäden oder indirekte Schäden am Roboter oder zufällige oder besondere Folgeschäden, die sich daraus ergeben.

- Einsatz in explosionsgefährdeten Umgebungen
- Medizinische und humanbezogene Verwendungen
- Zum Transport von Menschen und Tieren
- Einsatz ohne Risiko-Assessment
- Einsatz an Orten, die die Sicherheitsfunktion nicht erfüllen
- Verwendung außerhalb der Leistungs-/Umgebungsspezifikationen

※ Die unsachgemäße Verwendung ist nicht auf die oben genannten Punkte beschränkt.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.4 Risikopotenzial

Falls festgestellt wird, dass das endgültige System mit einem Roboterarm das Risiko mit den in den Roboterarm eingebauten Sicherheitsfunktionen nicht ausreichend reduzieren kann, müssen zusätzliche Schutzmaßnahmen festgelegt werden.

- Verletzungen (Stenose), die durch Einklemmen der Finger im Gelenkbereich des Roboters entstehen können
- Verletzungen (Stechen, tief Schneiden) durch scharfe Kanten von Werkzeugen
- Verletzungen (Stechen, tief Schneiden, Sturz) durch die sich in der Nähe des Roboters befindenden Gegenstände
- Verletzungen (Hautschäden, Atemnot) durch Arbeiten mit Gift- und Gefahrstoffen
- Verletzungen (Prellungen, Brüche) durch Kollisionen mit Robotern
- Verletzungen durch die nicht vollständig gesicherten Befestigungselemente
- Verletzungen durch vom Werkzeug gelöste Gegenstände

※ Je nach endgültigen Systemen können die Risikopotenziale variieren.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.5 Haftungsbeschränkung

Diese Bedienungsanleitung deckt nicht alle sicherheitsrelevanten Peripheriegerät ab. Das Installationspersonal des Systems muss die Sicherheitsanforderungen gemäß den Sicherheitsvorschriften und Gesetzen des Landes erfüllen, in dem der Roboterarm installiert wird. Der Roboterarm besteht aus einem endgültigen System, das mit Peripheriegeräten verbunden ist. Diese Bedienungsanleitung behandelt nicht alle Peripheriegeräte, einschließlich Design, Installation, Betrieb und Sicherheitsvorrichtungen des endgültigen Systems. Das endgültige System, an dem der Roboterarm angebracht wird, muss gemäß den Sicherheitsanforderungen konstruiert und installiert werden, um den Vorschriften und Gesetzen des Landes zu entsprechen.

Der Bediener oder der Benutzer des endgültigen Systems mit dem Roboterarm ist verantwortlich für:

- Risiko-Assessment für das endgültige System
- Hinzufügen von Sicherheitsvorrichtungen gemäß den Ergebnissen des Risiko-Assessments
- Sicherstellung von ordnungsgemäßem Design, Einstellungen und Installation
- Definition der Nutzung für das System
- Klärung wichtiger Kennzeichen oder Kontaktinformationen bezüglich Gebrauch und Sicherheit
- Bereitstellung von technischer Dokumentation wie Bedienungsanleitung

※ Peripheriegeräte sind nicht auf die oben genannten Punkte beschränkt. Die Einhaltung der Sicherheitshinweise in dieser Bedienungsanleitung bedeutet nicht, dass alle Risikopotenziale vollständig vermeidbar sind.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.6 Lieferung und Transport

Für Lieferung und Transport sind mindestens zwei Personen erforderlich- Achten Sie dabei darauf, das Produkt nicht durch Herunterfallen zu beschädigen. Schäden am Produkt während Lieferung und Transport sind von der Gewährleistung ausgeschlossen.

Warnung:

1) Achten Sie darauf, das Produkt während Lieferung und Transport nicht durch das Herunterfallen zu beschädigen. Schäden am Produkt während Lieferung und Transport sind von der Gewährleistung ausgeschlossen.
2) Beim Transport des Roboterarms können starke Vibrationen oder Stöße das System beschädigen. Der Roboter muss in der vom Hersteller bereitgestellten Verpackung transportiert werden.

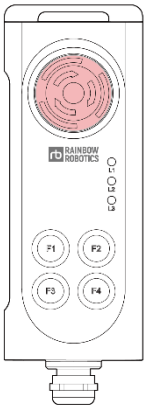
© RAINBOW ROBOTICS Inc. All rights reserved.

2.7 Not-Aus

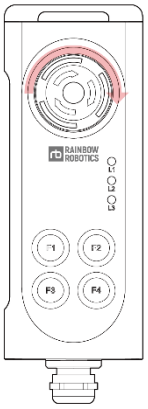
Im Notfall kann der Roboter gestoppt werden, indem der Not-Aus-Schalter verwendet wird, der an die Steuerbox angeschlossen ist, um den Roboterarm dringend anzuhalten.

Im Folgenden wird die Bedienweise des Not-Aus-Schalters der Standsteuerbox beschrieben.

- Not-Aus
  Der Roboterarm kann sofort durch Drücken des Not-Aus-Schalters gestoppt werden.



- Not-Aus-Entriegelung
  Drehen Sie den Not-Aus-Schalter im Uhrzeigersinn, um den Not-Aus zu entriegeln.



© RAINBOW ROBOTICS Inc. All rights reserved.

2.8 Benutzersicherheit

Zu Ihrer Sicherheit bitte beachten Sie Folgendes.

- Betrieb des Roboters in einem nicht motorisierten Zustand

Wenn im Notfall der Roboterarm nicht mit Strom versorgt werden kann oder soll, kann Kraft auf das Gelenk ausgeübt werden, um den Roboterarm zu bewegen (Forced Backdrive). Um einen erzwungenen Rückwärtsantrieb zum Bewegen des Gelenks durchzuführen, muss der Roboterarm so stark wie möglich gedrückt oder gezogen werden.

Warnung

Warnung:

1) Wenn Sie in einem nicht motorisierten Zustand übermäßige Kraft anwenden, um das Gelenk zu bewegen, kann das Antriebsteil beschädigt werden. Der Hersteller haftet nicht für Schäden, die durch übermäßige Krafteinwirkung verursacht werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.9 SICHERHEITSKONTROLLSYSTEM

Das Sicherheitskontrollsystem (Safety Control System) der Rainbow Robotics-Robotermodelle wie RB5-850E Serie, RB3-1200E Serie und RB10-1300E Serie entspricht ISO 13849-1 Cat3. PLd.

© RAINBOW ROBOTICS Inc. All rights reserved.

2.10 Risiko-Assessment

Risiko-Assessment ist ein wichtiger Faktor bei der Konstruktion eines Systems mit einem Roboterarm. Ein Risiko-Assessment kann nicht mit einem Roboter allen durchgeführt werden, weil die Sicherheit einer Roboteranlage davon abhängt, wie der Roboterarm ins Gesamtsystem integriert ist. Um Risiko-Assessment durchzuführen, müssen Benutzer, die Systeme mit Robotern konfigurieren, Roboter gemäß den Richtlinien von ISO 12100 und ISO 10218-2 suchen und betreiben. Siehe zusätzlich die technische Spezifikation ISO/TS 15066.

Risiko-Assessment muss unmittelbar nach der Installation des Roboterarms durchgeführt werden. Hauptinhalte vom Risiko-Assessment liegt darauf, die Notwendigkeit von Sicherheitskonfiguration, zusätzlichen Not-Aus-Schaltern und Schutzmaßnahmen zu bestimmen, die für die Umgebung und die Roboterkonfiguration geeignet sind.

Die sicherheitsbezogene Funktion des kollaborativen Roboters können im Einstellungsmenü für Sicherheitskonfiguration konfiguriert werden und bietet die folgenden Funktionen.

- I/O-Einstellung: Die Steuerbox kann so eingestellt werden, dass wichtige sicherheitsrelevante Informationen an die Ausgangsanschluss ausgegeben werden.
- Geschwindigkeitssteuerung: Der Benutzer kann Bewegungsgeschwindigkeit des Roboterarms steuern.
- Anpassung der Empfindlichkeit der Kollisionserkennung: Wenn der Roboter mit einem Objekt in der Nähe kollidiert, hält er automatisch an. Der Benutzer kann die Empfindlichkeit anpassen, mit der der Roboterarm die Kollision erkennt.

Wenn die oben genannten sicherheitsbezogenen Funktion die Gefahr nicht ausreichend verringern oder beseitigen, sollte eine zusätzliche Schutzvorrichtung hinzugefügt, um die Gefahr zu beseitigen. Der Hersteller ist nicht verantwortlich für Unfälle, die durch Nichteinhaltung der einschlägigen Anforderungen internationaler Normen und nationaler Gesetze in Bezug auf die Installation oder Nichtbeachtung vom obigen Risiko-Assessment verursacht werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 3 Sicherheitsvorrichtungen

3.1 Einleitung

Dieses Kapitel enthält wichtige Sicherheitsinformationen, die der Integrator der kollaborativen Roboter der RB-Serie lesen und verstehen muss, bevor der Roboter zum ersten Mal eingeschaltet wird.

Die RB-Serie bietet eine Vielzahl von Sicherheitsfunktionen und Schnittstellen für Sicherheitsgeräte zum Schutz von Benutzern und Geräten. Sicherheitsfunktionen und Schnittstellen erfüllen Kategorie 3, Performance Level d (PL d) in ISO 13849-1 und Hardware Fault Tolerance 1, Safety Integrity Level 2(SIL 2) in IEC 62061.

Warnung:

1) Je nachdem, wo der Roboter installiert ist, muss der Integrator des Systems ein Risiko-Assessment durchführen und den Betriebsbereich mit Sicherheitsüberwachung und Schnittstellen entsprechend konfigurieren.
2) Wenn ein Fehler bei der Sicherheitsfunktion, Schnittstelle des Roboters erkannt wird, wird ‚Stoppkategorie(Stop Category) 0' eingeleitet.
3) Beispiele für Fehler sind Kabelbruch im Not-Aus-Kreis, falsche Verdrahtung zusätzlicher Sicherheitseinrichtungen und nicht überlappende Verdrahtung weiterer Sicherheitseinrichtungen (siehe Abschnitt 5 dieses Kapitels).
4) Integrator und Bediener des Systems sollten sich der Tatsache bewusst sein, dass es eine Sicherheitsüberwachung gibt, die vom Roboter intern ausgeführt wird. Die Sicherheitserkennungsfunktion umfasst nicht nur die Bestätigung des Not-Aus-Schalters und der Schutzstopp-Vorrichtung, Der Roboter kann auch bei Bewegungen oberhalb der physikalischen Grenze wie Position, Geschwindigkeit und Impuls des Roboterarms im vorgesehenen Stoppmodus angehalten werden (Siehe Abschnitt 3 dieses Kapitels für die Sicherheitsüberwachung).
5) Integrator und Bediener des Systems müssen die Dauer und Anhalteweg bis zum Anhalten des Roboters aufgrund des Betriebs der oben beschriebenen Fehler- und Sicherheitsüberwachungsfunktionen berücksichtigen. Sie müssen ein Risiko-Assessment unter Berücksichtigung von Anhalteweg und -dauer durchführen (siehe Abschnitt 4 dieses Kapitels).
6) Integrator und Bediener des Systems sollten Betriebsbereich des Roboters auswählen und dabei auf Sicherheitsüberwachungsfunktion berücksichtigen, die die Gelenke des Roboters und die Roboter-/TCP-Bewegung begrenzen. TCP bezeichnet die Position, an der der Offset zum Mittelpunkt des Endes vom Roboterarm hinzugefügt wird.

© RAINBOW ROBOTICS Inc. All rights reserved.

Gefahr:

1) Der Systemintegrator muss vor dem Einschalten des Roboters ein Risiko-Assessment durchführen. Wenn der Roboter anders verwendet wird oder die Parameter eingestellt werden als im Risiko-Assessment bestimmt, können nicht ausreichend reduzierte Risiken auftreten.
2) Beim Verdrahten zusätzlicher Sicherheitsvorrichtungen muss die Stromversorgung sowohl des Roboters als auch des Bedienfelds unterbrochen sein.
3) Bei der Installation zusätzlicher Sicherheitseinrichtungen sind Maßnahmen zu treffen, damit ein mechanischer Einsatz problemlos ist. Beispielsweise bei der Verwendung eines Lichtvorhangs muss der Roboter fest mit dem Boden/der Vorrichtung verbunden sein und darf sich während des Roboterbetriebs nicht bewegen oder vibrieren.
4) Alle Schnittstellen für die Sicherheitsfunktion sind auf 24 V eingestellt. Wenn Geräte mit unterschiedlichen Spannungen angeschlossen werden, kann dies zu Schäden am Gerät oder Brand führen.
5) Signale von Geräten, die am Werkzeugflansch montiert sind, sind nicht in der Sicherheitsfunktion enthalten. Schließen Sie die Sicherheitsvorrichtung nicht an das Werkzeugflanschkabel an.

© RAINBOW ROBOTICS Inc. All rights reserved.

3.2 Stopp-Kategorie (Stop Category)

Die Sicherheitsfunktion ermöglicht dem Roboter, drei Arten von Stopp-Kategorien zu starten, die in IEC 60204-1 definiert sind.

Stop Category	Beschreibung
0 [STO]	Der Roboter wird angehalten, indem die Stromversorgung sofort unterbrochen wird. *Die Gelenkbremsen können verschleißen und die Lebensdauer des Roboters verkürzen. Verwenden Sie es nur, wenn es absolut notwendig ist. **Da die Stromversorgung des Roboters unterbrochen wird, ist eine Aktivierung erforderlich, wenn er nach Risikobeseitigung wieder verwendet wird.
1 [SS1]	Nach dem Abbremsen auf das Maximum und Stoppen aller Gelenke des Roboters wird die Stromversorgung unterbrochen und der Roboter gestoppt. * Da die Stromversorgung des Roboters unterbrochen wird, ist eine Aktivierung erforderlich, wenn er nach Risikobeseitigung wieder verwendet wird.
2 [SS2]	Nach dem Abbremsen auf das Maximum und Stoppen aller Gelenke des Roboters geht der Roboter in den SOS-Zustand über. *SOS: Die Stromversorgung wird angelegt und die aktuelle Position wird in einem aktivierten Zustand gehalten. Bei der Erkennung einer Positionsänderung wird die Stopp-Kategorie 0 gestartet. ** Das keine Stromunterbrechung vorhanden ist, kann der Roboter nach Risikobeseitigung sofort verwendet werden.

Vorsicht:

1) Gemäß ISO10218-1 5.5.2 und 5.5.3 sollte eine geeignete Stopp-Kategorie für Not-Aus und Schutzstopp gewählt werden.
2) Im Falle eines Not-Aus muss eine der Stopp-Kategorien 0 und 1 ausgewählt werden.
3) Bei Not-Aus ist Aktivierung erforderlich.
4) Bei Schutzstopp muss mindestens eine der Stopp-Kategorien 0 und 1 gewählt werden.
5) Zusätzliche Schutzstopps können die Stopp-Kategorie 2 verwenden.

© RAINBOW ROBOTICS Inc. All rights reserved.

3.3 Funktionssicherheit

Der Hersteller empfiehlt, dass für den Installationsort folgende Bedingungen erfüllt sind. Die Sicherheitsfunktionen des kollaborativen Roboters RB-Serie dienen dazu, das durch das Risiko-Assessment ermittelte Risiko des Robotersystems zu reduzieren.

Die Parameter der Sicherheitsfunktion werden zum Zeitpunkt der Auslieferung anfänglich eingestellt. Integrator des Systems kann einige Elemente gemäß des Risiko-Assessments ändern. Elemente für Position und Geschwindigkeit basieren auf der Basis des Roboters.

Nachfolgend finden Sie die Sicherheitsfunktionsspezifikationen für die RB-Serie.

	Sicherheitsfunktion	PL & Kategorie
Sicherheits-stopp-funktionen	SF.1 STO (Safe Torque Off)	PL d, Kategorie 3
	SF.2 SS1 (Safe Stop 1)	PL d, Kategorie 3
	SF.3 SS2 (Safe Stop 2)	PL d, Kategorie 3
Sicherheits-überwachungs-funktion	SF.4 SOS (Safe Operating Stop)	PL d, Kategorie 3
	SF.5 SLP (Safely-Limited Position)	PL d, Kategorie 3
	SF.6 SLS (Safely-Limited Speed)	PL d, Kategorie 3
	SF.7 SLA (Safely-Limited Acceleration)	PL d, Kategorie 3
	SF.8 SLI (Safely-Limited Increment)	PL d, Kategorie 3
	SF.9 SLT (Safely-Limited Torque)	PL d, Kategorie 3
	SF.10 RPL (Robot Position Limit)	PL d, Kategorie 3
	SF.11 TSL (TCP Speed Limit)	PL d, Kategorie 3
	SF.12 CBPL (Control Box Power Limit)	PL d, Kategorie 3
Not-Aus	SF.13 EMS1 (Emergency Stop1)	PL d, Kategorie 3
	SF.14 EMS2 (Emergency Stop2)	PL d, Kategorie 3
Schutzstopp	SF.15 PRS (Protective Stop)	PL d, Kategorie 3
	SF.16 HSS (Hard Safeguard Stop)	PL d, Kategorie 3
	SF.17 SSS (Soft Safeguard Stop)	PL d, Kategorie 3

- **STO(Safe Torque Off)**: Diese Funktion verhindert, dass dem Motor kraftzeugende Leistung zugeführt wird. Dem Motor wird keine Energie zugeführt, die eine Drehung verursachen kann. Diese Teilfunktion bezüglich der Sicherheit entspricht dem ungesteuerten Stillsetzen nach Stopp-Kategorie in IEC 60204-1.

- **SS1(Safe Stop 1)**: Diese Funktion wird als eines der folgenden angegeben. a) SS1-d (Safe-Stop 1 Deceleration Control) startet und steuert die Motorverzögerungsgeschwindigkeit innerhalb der ausgewählten Grenze, um den Motor zu stoppen, und führt die STO-Funktion aus, wenn die Motorgeschwindigkeit unter der angegebenen Grenze liegt. ODER b) SS1-r (Safe-Stop 1 Ramp Monitoring) startet und überwacht die Motorverzögerungsgeschwindigkeit innerhalb der ausgewählten Grenze, um den Motor zu stoppen, und führt die STO-Funktion aus, wenn die Motorgeschwindigkeit unter der angegebenen Grenze liegt. ODER c) SS1-t (Safe Stop 1 Time Control) startet die Motorverzögerung und führt die STO-Funktion nach einer anwendungsspezifischen Zeitverzögerung aus. Diese sicherheitsbezogene Teilfunktion

© RAINBOW ROBOTICS Inc. All rights reserved.

entspricht einem gesteuerten Stopp nach Stopp-Kategorie 1 der IEC 60204-1. Unser System verwendet SS1-t.

- SS2(Safe Stop 2): Diese Funktion wird als eines der folgenden angegeben. a) SS2-d (Safe-Stop 2 Deceleration Control) startet und steuert die Motorverzögerungsgeschwindigkeit innerhalb der ausgewählten Grenze, um den Motor zu stoppen und den Schutzstopp auszuführen, wenn die Motorgeschwindigkeit unter der angegebenen Grenze liegt. ODER b) SS2-r (Safe-Stop 2 Ramp Monitoring) startet und überwacht die Motorverzögerungsgeschwindigkeit innerhalb der ausgewählten Grenze, um den Motor zu stoppen, und führt die STO-Funktion aus, wenn die Motorgeschwindigkeit unter der angegebenen Grenze liegt. ODER c) SS2-t (Safe Stop 2 Time Control) startet die Motorverzögerung und führt den Schutzstopp nach einer anwendungsspezifischen Zeitverzögerung aus. Diese sicherheitsbezogene Teilfunktion SS2 entspricht einem gesteuerten Stopp nach Stopp-Kategorie 2 der IEC 60204-1. Unser System verwendet SS2-t.

- SOS(Safe Operating Stop): Diese Funktion verhindert, dass der Motor um mehr als einen definierten Betrag von der gestoppten Position abweicht. Das PDS (SR) versorgt den Motor mit Energie, damit er äußeren Kräften widerstehen kann. Diese Beschreibung der Stoppfunktion basiert auf einer Implementierung mittels PDS (SR) ohne externe (mechanische) Bremsen.

- SLP(Safely-Limited Position): Diese Funktion verhindert, dass die Motorwelle(shaft) (oder der Mover bei Verwendung eines Linearmotors) die angegebene Positionsgrenze überschreitet.

- **SLS(Safely-Limited Speed)**: Diese Funktion verhindert, dass der Motor die angegebene Drehzahlgrenze überschreitet.

- **SLA(Safely-Limited Acceleration)**: Diese Funktion verhindert, dass der Motor die angegebenen Beschleunigungs- oder Verzögerungsgrenzen überschreitet.

- **SLI(Safely-Limited Increment)**: Diese Funktion verhindert, dass die Motorachse die angegebene Positionserhöhungsgrenze innerhalb der angegebenen Zeit überschreitet.

- **SLT(Safely-Limited Torque)**: Diese Funktion verhindert, dass der Motor seine spezifizierte Drehmomentgrenze (oder Kraft bei Verwendung eines Linearmotors) überschreitet.

- **RPL(Robot Position Limit)**: Diese Funktion verhindert, dass der TCP (Tool Center Point) oder Körperrahmen des Roboterarms den angegebenen räumlichen Bereich überschreitet.

- **TSL(TCP Speed Limit)**: Diese Funktion verhindert, dass die TCP-Geschwindigkeit des Roboterarms die angegebene Geschwindigkeit überschreitet.

© RAINBOW ROBOTICS Inc. All rights reserved.

- CBPL(Control Box Power Limit): Diese Funktion verhindert, dass der Stromverbrauch der Steuerbox die angegebene Grenze überschreitet.

- **EMS1(Emergency Stop1)**: Diese Funktion aktiviert den Stoppmodus, wenn der Not-Aus-Schalter des Programmierhandgeräts aktiviert wird. Der Stoppmodus ist SF.2.

- **EMS2(Emergency Stop2)**: Diese Funktion aktiviert den Stoppmodus, wenn der spezielle Ein/Aus-Anschluss der Steuerbox aktiviert wird. Diese Anschlüsse werden bereitgestellt, damit Benutzer ihre eigenen Schalter-Geräte anschließen können. Der Stoppmodus ist SF.2.

- **PRS(Protective Stop)**: Diese Funktion aktiviert den Stoppmodus, wenn der spezielle Ein/Aus-Anschluss der Steuerbox aktiviert wird. Diese Anschlüsse werden bereitgestellt, damit Benutzer ihre eigenen Schutzvorrichtungen anschließen können. Der Stoppmodus ist SF.2.

- **HSS(Hard Safeguard Stop)**: Diese Funktion aktiviert den Stoppmodus, wenn der spezielle Ein/Aus-Anschluss der Steuerbox aktiviert wird. Diese Anschlüsse werden bereitgestellt, damit Benutzer ihre eigenen Schutzvorrichtungen anschließen können. Der Stoppmodus ist SF.1.

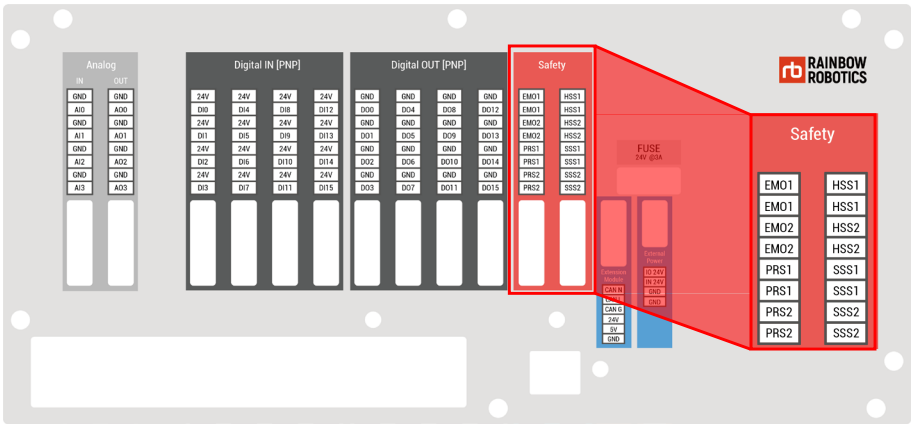
- **SSS(Soft Safeguard Stop)**: Diese Funktion aktiviert den Stoppmodus, wenn der spezielle Ein/Aus-Anschluss der Steuerbox aktiviert wird. Diese Anschlüsse werden bereitgestellt, damit Benutzer ihre eigenen Schutzvorrichtungen anschließen können. Der Stoppmodus ist SF.3.

© RAINBOW ROBOTICS Inc. All rights reserved.

3.4 Montagebereich der Sicherheitsvorrichtung

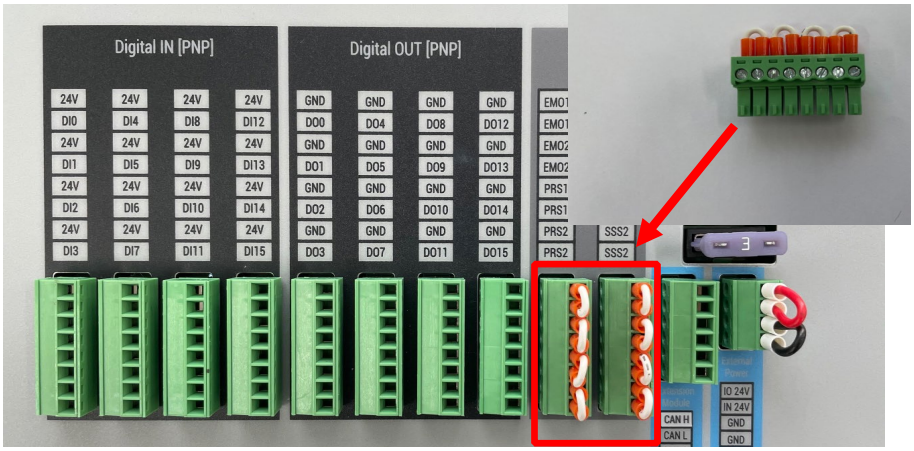
Zusätzlich zum Not-Aus-Schalter kann die RB-Serie mit zusätzlichen Sicherheitsvorrichtungen vom Integrator des Systems ausgestattet werden, die durch Risiko-Assessment gefordert werden.

Die sicherheitsbezogene Kontaktklemme besteht aus 16 Anschlüssen. Dieser Anschluss ist ein Redundanz-dedizierter Kontakteingangsanschluss.



Die zusätzlichen Anschlüsse können mit 4 Geräten bestückt werden. Der Roboter wird mit einer Standardkonfiguration geliefert, die einen Betrieb ohne zusätzliche Sicherheitsvorrichtung ermöglicht.

Wenn Sie das Gerät ohne Anschluss an ein externes Sicherheitsgerät verwenden, achten Sie darauf, den Basiskontakteingang wie unten gezeigt anzuschließen und zu verwenden.



Die Spezifikationen der Anschlüsse für Sicherheitsvorrichtungen lauten wie folgt.

© RAINBOW ROBOTICS Inc. All rights reserved.

- **EMO**

Dieser Anschluss wird verwendet, wenn aufgrund des Risiko-Assessments ein zusätzlicher Not-Aus-Schalter installiert werden muss.

Für den Not-Aus-Schalter muss ein Produkt gemäß IEC 60947-5-5 verwendet werden.

Not-Aus durch EMO wird als Stopp-Kategorie 1 bezeichnet.

- **PRS**

Dieser Anschluss wird verwendet, wenn aufgrund des Risiko-Assessments eine oder mehrere Schutzstopp-Vorrichtungen angeschlossen werden sollen.

Für die Schutzstopp-Vorrichtungen müssen Produkte gemäß 5.3.8.3 in ISO 10218-2 verwendet werden.

Schutzstopp durch PRS wird als Stopp-Kategorie 1 bezeichnet.

- **HSS**

Dieser Anschluss wird verwendet, wenn aufgrund des Risiko-Assessments eine oder mehrere Schutzstopp-Vorrichtungen angeschlossen werden sollen.

Für die Schutzstopp-Vorrichtungen müssen Produkte gemäß 5.3.8.3 in ISO 10218-2 verwendet werden.

Schutzstopp durch HSS wird als Stopp-Kategorie 0 bezeichnet.

- **SSS**

Dieser Anschluss wird verwendet, wenn aufgrund des Risiko-Assessments eine oder mehrere Schutzstopp-Vorrichtungen angeschlossen werden sollen .

Für die Schutzstopp-Vorrichtungen müssen Produkte gemäß 5.3.8.3 in ISO 10218-2 verwendet werden .

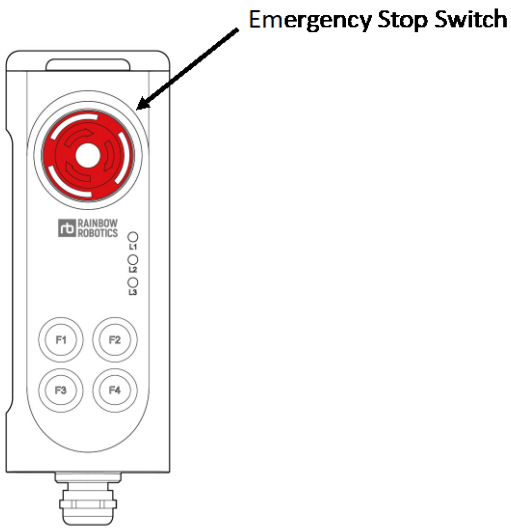
Schutzstopp durch SSS wird als Stopp-Kategorie 2 bezeichnet.

© RAINBOW ROBOTICS Inc. All rights reserved.

3.5 Not-Aus-Schalter

Bei den kollaborativen Robotern RB-Serie kann der Bediener den Not-Aus-Schalter verwenden, um den Roboterarm im Notfall anzuhalten.

Im Notfall muss der Roboter sofort durchs Drücken des Not-Aus-Schalters oben auf dem Handprogrammiergerät gestoppt werden.



Vorsicht

Vorsicht:

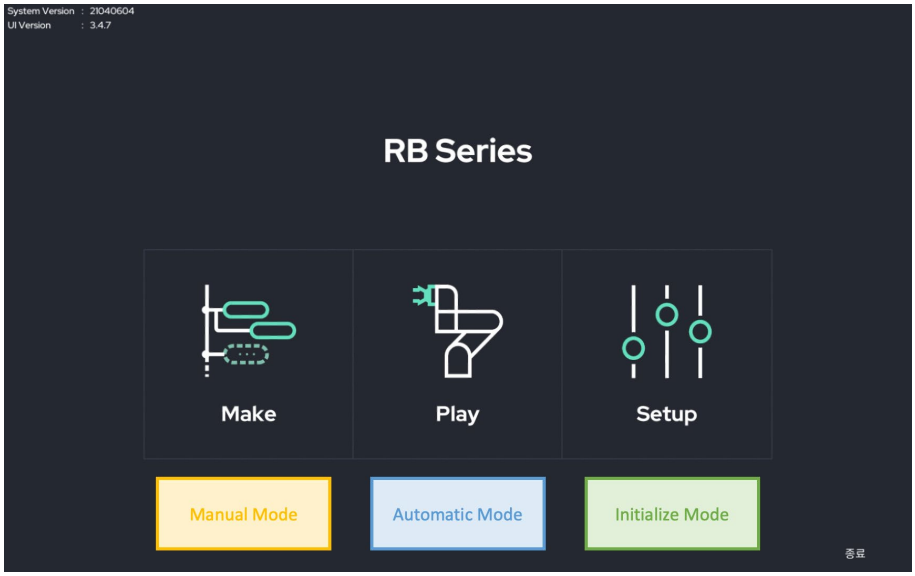
1) Not-Aus-Schalter wird als Stopp-Kategorie 1 bezeichnet.
2) Drehen Sie den Not-Aus-Schalter im Uhrzeigersinn, wird die Not-Aus-Funktion freigegeben.
3) Wird zusätzlicher Not-Aus-Schalter benötigt, kann dieser über die Steuerbox verwendet werden.
4) Not-Aus-Funktion sollte nicht als Risikominderung verwendet werden, sondern als sekundärer Schutz verwendet werden.

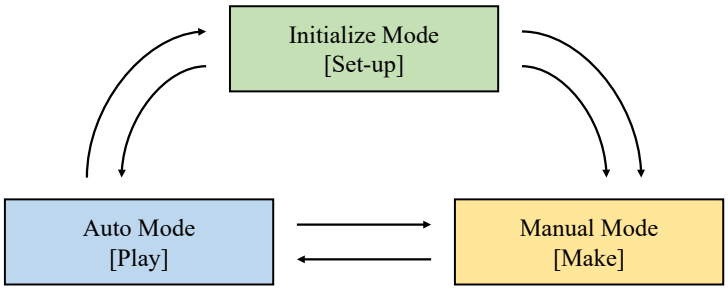
© RAINBOW ROBOTICS Inc. All rights reserved.

3.6 Betriebsmodus

Der Betriebsmodus des kollaborativen Roboters RB-Serie ist wie folgt.

Beim Aufruf des Automatikmodus (Automatic Mode) müssen Sie über ein Passwort darauf zugreifen.





Vorsicht

Vorsicht:

1) Bei Auslieferung ist das Passwort zum Aufrufen des Auto-Modus nicht eingestellt. Sie müssen vor der Verwendung ein Passwort einstellen, um zu verhindern, dass jemand anderes darauf zugreift.
2) Vor dem Aufrufen des Auto-Modus muss der Benutzer gefährliche Situationen beseitigen und den Status des Not-Aus-Schalters und der Schutzstopp-Vorrichtung überprüfen.

© RAINBOW ROBOTICS Inc. All rights reserved.

3) In jedem Fall muss der Benutzer vor der Bedienung des Roboters den Installationsstatus des Roboters richtig verstehen und die Einstellung abschließen.

- **Initialize Mode [Set-up]**

Vor der Bewegung des Roboters können die Peripherieeinstellungen oder der Status des Roboters festgelegt werden.

Allgemeine Einstellungen für die Roboterbewegung, wie Arbeitsbereich, TCP-Offset und Nutzlast können vorgenommen werden.

Zu diesem Zeitpunkt wird dem Motor kein Strom zugeführt, und durch eine Aktivierungsaktion kann dem Motor Strom zugeführt werden.

- **Auto Mode [Play]**

Ein Zustand, in dem der Roboter nur vordefinierte Aufgaben ohne Benutzereingriff ausführt. Zu diesem Zeitpunkt wird dem Motor Strom zugeführt.

Durch die Simulationsfunktion kann der programmierte Roboterbetrieb verifiziert werden.
Der Roboter kann mit dem verifizierten Programm betrieben werden, indem es in einen ‚Real-Zustand' umgewandelt wird.

Zu diesem Zeitpunkt führt der Roboter im Voraus festgelegte Sicherheitsfunktionen aus. Der Benutzer kann den Status des Roboters und der Peripheriegeräte über das Wiedergabefenster(Play-Fenster) überwachen.

- **Manual Mode [Make]**

Ein Zustand, in dem der Roboter durch die direkte Steuerung des Benutzers betrieben werden. Zu diesem Zeitpunkt wird dem Motor Strom zugeführt.

Direkt-Teaching, Programmerstellung und -änderung sowie manueller Betrieb von Peripheriegeräten können durchgeführt werden. Zu diesem Zeitpunkt kann der Roboter vom Benutzer über Schieber für sichere Geschwindigkeit auf dem Tablet bedient werden.

Wenn Sie Ihre Hand vom Sicherheitsschieber lösen, stoppt der Roboter sofort seine Bewegung.

Vorsicht

Vorsicht:

1) Für Direkt-Teaching muss die Sicherheitsschieber eingestellt werden.
2) Bei der Auslieferung ist die Sicherheitsschieber deaktiviert.
3) Für den 3-Stellungs-Zustimmschalter müssen die Produkte gemäß 5.8.3 in ISO 10218-1 verwendet werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

3.7 Betriebsumgebung

Um den Roboter langfristig in einem sicheren Zustand zu halten, sollte er in den folgenden Umgebungen verwendet werden.

Maximal zulässige Betriebstemperatur	50˚C
Maximal zulässige Lagertemperatur	60˚C
Minimal zulässige Betriebstemperatur	0˚C
Minimal zulässige Lagertemperatur	-5˚C
Maximal zulässige Luftfeuchtigkeit	80%
Minimal zulässige Luftfeuchtigkeit	20%

© RAINBOW ROBOTICS Inc. All rights reserved.

3.8 Wartung von Sicherheitsfunktion

Um den Roboter langfristig in einem sicheren Zustand zu halten, ist die kontinuierliche Wartung von Sicherheitsfunktion notwendig.

Eine regelmäßige Inspektion der folgenden Punkte von Sicherheitsinspektionspersonal ist erforderlich. Sollten Sie bei der Inspektion ein Problem feststellen, das Sie nicht selbst lösen können, wenden Sie sich bitte an den Hersteller.

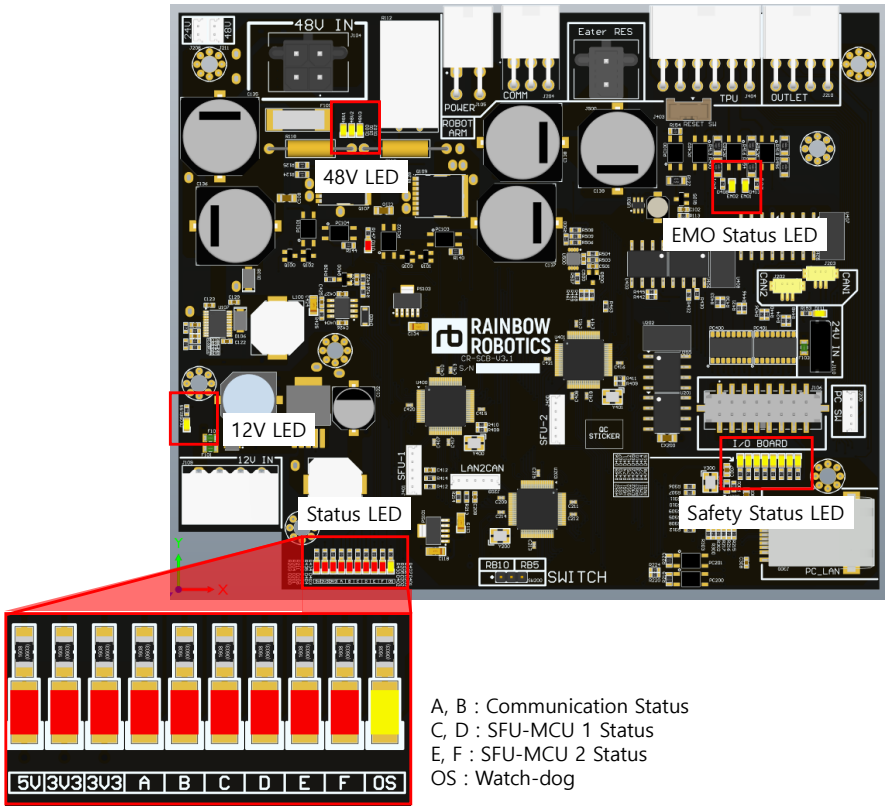
Inspektionsziel		Checkliste	Zyklus
Hand-programmier-Gerät	Sicherheits-funktion	Prüfen Sie, ob der am Handprogrammiergerät montierte Not-Aus-Schalter problemlos funktioniert.	monatlich
	Kabel	Überprüfen Sie den Zustand des Verbindungskabels zwischen dem Handprogrammiergerät und der Steuerbox.	monatlich
Steuerbox	Schnittstelle	Überprüfen Sie, ob der EMO-Anschluss, der die Sicherheitsvorrichtung verbindet, ordnungsgemäß funktioniert.	monatlich
		Überprüfen Sie, ob der PRS-Anschluss, der die Sicherheitsvorrichtung verbindet, ordnungsgemäß funktioniert.	monatlich
		Überprüfen Sie, ob der HSS-Anschluss, der die Sicherheitsvorrichtung verbindet, ordnungsgemäß funktioniert.	monatlich
		Überprüfen Sie, ob der SSS-Anschluss, der die Sicherheitsvorrichtung verbindet, ordnungsgemäß funktioniert.	monatlich
	Strom-versorgung	Überprüfen Sie die normale Ausgabe der 24V-Spannung, die die Sicherheitsvorrichtung anschließt.	monatlich
		Überprüfen Sie, ob die 24V-Sicherung richtig eingesetzt ist.	monatlich
	Kabel	Überprüfen Sie den Zustand des Verbindungskabel zwischen der Sicherheitsvorrichtung und der Steuerbox.	monatlich

© RAINBOW ROBOTICS Inc. All rights reserved.

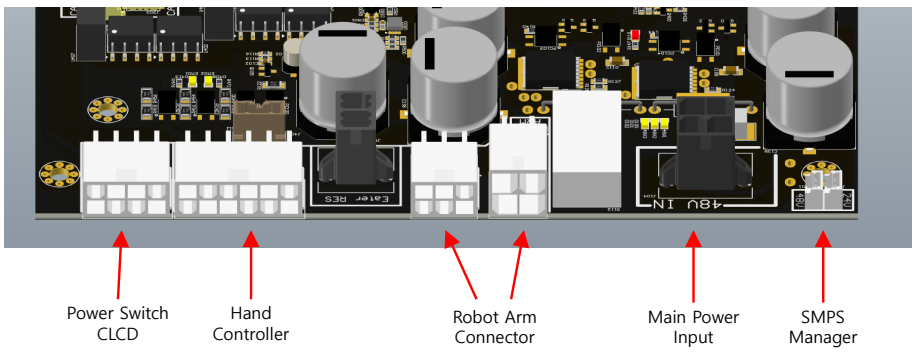
■ Spezifikation der Sicherheitsfunktionsplatine

In der Steuerbox ist eine Sicherheitsfunktionsplatine zum Antrieb des kollaborativen Roboters RB-Serie eingebaut.

Die Informationen der LED, die den Betriebsstatus der Platine zeigt, sind wie folgt.



Informationen über Anschlüsse, die mit der Platine verbunden sind, lauten wie folgt.



© RAINBOW ROBOTICS Inc. All rights reserved.

3.9 Angewandte Norm

Standard	Überschrift
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety related systems
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements
IEC 61508-4:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations
IEC 61508-5:2010	Functional safety of electrical/electronic/programmable electronic safety-related system – Part 5: Examples of methods for the determination of safety integrity levels
IEC 61508-6:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3
IEC 61508-7:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures
IEC 60204-1:2016	Safety of machinery – Electrical equipment of machines – Part 1: General requirements
IEC 61000-6-1: 2016	Electromagnetic compatibility (EMC) – Part 6-1: Generic standards – Immunity standard for residential, commercial, and light-industrial environments
IEC 61000-6-2: 2016	Electromagnetic compatibility (EMC) – Part 6-2: Generic standards – Immunity standard for industrial environments

© RAINBOW ROBOTICS Inc. All rights reserved.

IEC 61000-6-7: 2014	Electromagnetic compatibility (EMC) – Part 6-7: Generic standards – Immunity requirements for equipment intended to perform functions in a safety-related system (functional safety) in industrial locations
IEC 61326-3-1: 2017	Electrical equipment for measurement, control, and laboratory use – EMC requirements – Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) – General industrial applications
IEC 61800-5-1: 2007	Adjustable speed electrical power drive systems – Part 5-1: Safety requirements –Electrical, thermal and energy
IEC 61800-5-2: 2016	Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional
IEC 62061:2005	Safety of machinery – Functional safety of safety-related electrical, electronic, and programmable electronic control systems
ISO/TS 15066: 2016	Robots and robotic devices — Collaborative robots
ISO 10218-1: 2011	Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots
ISO 10218-2: 2011	Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration
ISO 12100:2010	Safety of machinery — General principles for design — Risk assessment and risk reduction
ISO 13849-1: 2015	Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design
ISO 13849-2: 2012	Safety of machinery — Safety-related parts of control systems — Part 2: Validation

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 4 Installation

4.1 Hinweise zur Installation

Der Roboterinstallateur muss den Roboter gemäß den Richtlinien von ISO 12100 und ISO 10218-2 installieren und betreiben, wobei internationale Standards wie ISO/TS 15066 und die relevanten Anforderungen der nationalen Gesetzgebung eingehalten werden müssen. Der Hersteller ist nicht verantwortlich für Unfälle, die durch Risiken verursacht werden, die den Anforderungen auf der Grundlage internationaler Normen oder auf der Grundlage nationaler Gesetze und Vorschriften entsprechen oder durch Nichtbeachtung des im vorherigen Kapitel erwähnten Risiko-Assessments verursacht werden.

Unsere Roboter müssen unter Berücksichtigung der Inspektionsstandards der “Regeln zu Arbeitsschutznormen” und “Sicherheitsprüfung” installiert werden. Einzelheiten finden Sie in Anhang M.

© RAINBOW ROBOTICS Inc. All rights reserved.

4.2 Installationsort

Es wird empfohlen, das System an einem Ort zu installieren, der die folgenden Bedingungen erfüllt.

- Gebäude mit erdbebensicherer Auslegung
- ein leckagefreier Ort
- Ort, der frei von brennbaren und explosiven Stoffen ist
- Ort, an dem Temperatur und Luftfeuchtigkeit konstant gehalten werden können
- Ort, an dem nicht viel Staub eindringt

Vorsicht

Vorsicht:

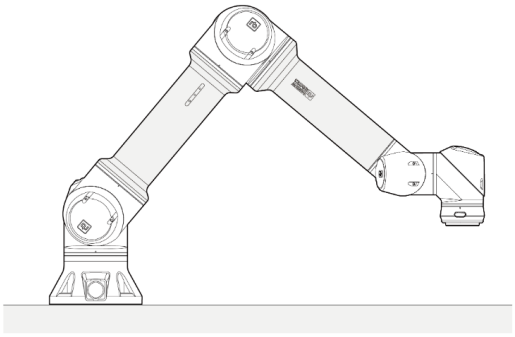
1) Wenn das System nicht an einem empfohlenen Ort installiert wird, kann die Leistung und Lebensdauer des Roboters reduziert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

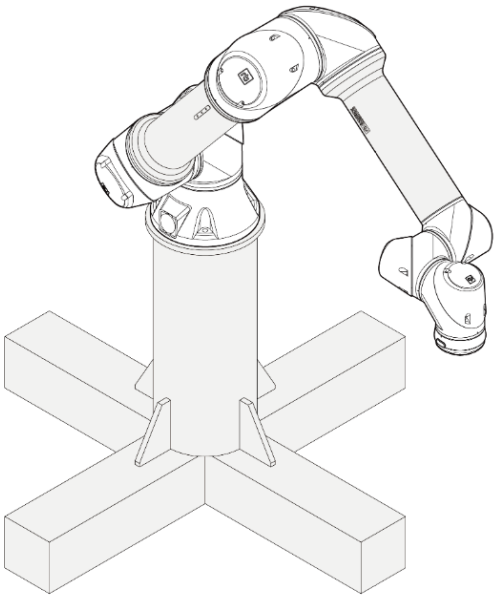
4.3 Installationstyp

Der Roboterarm kann beliebig installiert werden, z.B. auf dem Boden, an der Wand oder an einem festen Pfosten. Der Befestigungspfosten ist nicht im Lieferumfang des Produkts enthalten.

Darüber hinaus kann er an der Wand, der Decke in einem beliebigen Winkel installiert werden. Bei anderen Installationen als der flachen Installation muss der Installationswinkel über die Systemeinstellung eingegeben werden.



[Bodeninstallation]

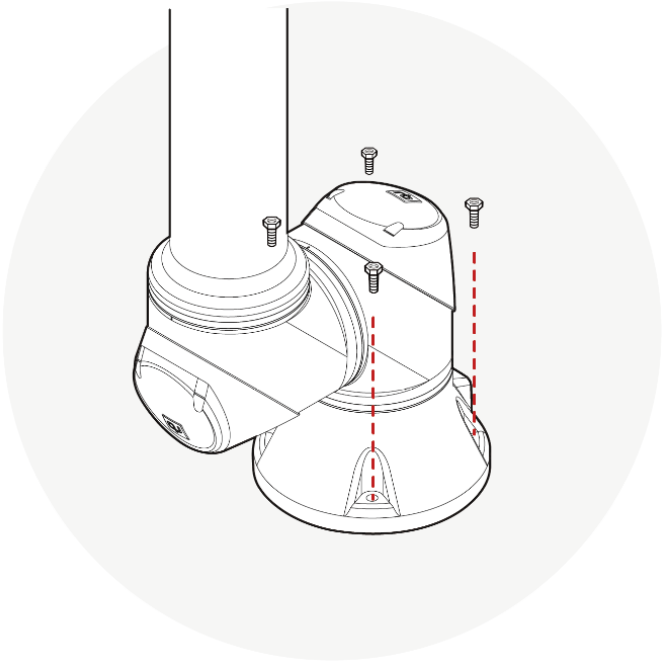


[Installation am festen Pfosten]

© RAINBOW ROBOTICS Inc. All rights reserved.

4.4 Befestigung des Roboterarms

Es wird empfohlen, den Roboterarm mit vier M8 30 mm Schrauben zu montieren.



Warnung

Warnung:

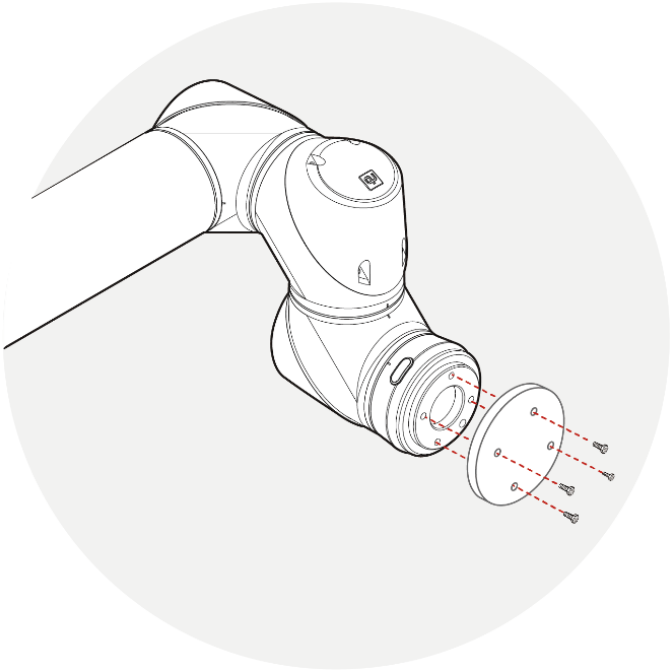
1) Ziehen Sie die Schrauben bei der Befestigung des Roboters fest, damit sie nicht gelöst werden.
2) Installieren Sie den Roboter auf einer stabilen Oberfläche, die dem Gewicht des Roboters und der Belastung beim Betrieb standhalten kann.
3) Überprüfen Sie, ob die Montagefläche des Roboters dicht an der Basisfläche steht.
4) Demontieren Sie nicht willkürlich die im Roboter montierten Bolzen. Stellen Sie vor dem Betrieb des Roboterarms sicher, dass alle Schrauben fest angezogen sind.
5) Wenn das Produkt mit demontierten Bolzen verwendet wird oder wenn Halterungen mit diesem Schraubenbauteil installiert werden, kann das Produkt beschädigt oder die Sicherheit beeinträchtigt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

4.5 Verbindung des Roboters mit dem Werkzeug

Mit vier M6-Schrauben wird das Werkzeug am Werkzeugflansch befestigt.

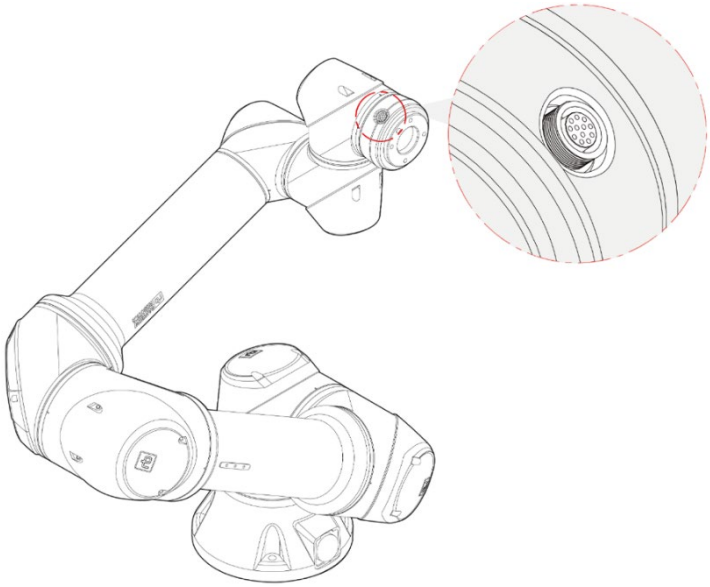
- Werkzeug und M6-Schrauben zur Befestigung sind nicht im Lieferumfang enthalten.
- Die Verbindungsmethoden können je nach Werkzeug variieren. Weitere Einzelheiten finden Sie in der von Hersteller bereitgestellten Bedienungsanleitung.



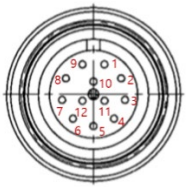
- Verbinden Sie nach der Befestigung des Werkzeugs am Werkzeugflansch die erforderlichen Kabel mit den Ein/Aus-Anschlüssen entweder am Werkzeug oder an der Steuerbox.
- Als Werkzeuganschluss wird SAMWOO SW-10W-12(P) verwendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Der Werkzeug-Ein/Aus hat einen 12-Pin-Stecker.



- Die Anschlussspezifikationen im Werkzeugflansch lauten wie folgt.

Anschluss	Layout	Pin-Nr.	Signal (Non-E Version)	Signal (E Version)	Farbe (Dicke)
Werkzeug Ein/Aus		1	Digital Output A	Digital Output A	**Braun (AWG22)**
		2	Digital Output B	Digital Output B	**Blau (AWG22)**
		3	0/12/24 VCC	0/12/24 VCC	**Rot (AWG22)**
		4	Ground	Ground	**Schwarz (AWG22)**
		5	Digital Input A	Digital Input A	Weiß (AWG26)
		6	Digital Input B	Digital Input B	Blau (AWG26)
		7	Analog Input A	Digital Input C	Weiß (AWG26)
		8	Analog Input B	Digital Input D	Gelb (AWG26)
		9	RS485+	RS485+	Grau (AWG26)
		10	RS485-	RS485-	Violett (AWG26)
		11	Common Ground	Digital Input E	Schwarz (AWG26)
		12	Common Ground	Digital Input F	Grün (AWG26)

- Die interne Stromversorgung kann auf dem Ein/Aus-Tab der GUI auf 0V, 12V oder 24V eingestellt werden.

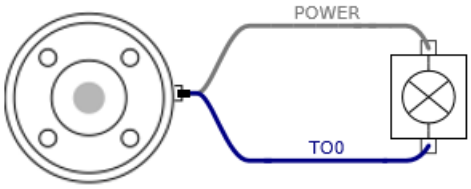
-	min.	Nominal	max.	Einheit
24V-Modus	-	24	-	V
12V-Modus	-	12	-	V
Beide Modi	-	-	2.000	mA

© RAINBOW ROBOTICS Inc. All rights reserved.

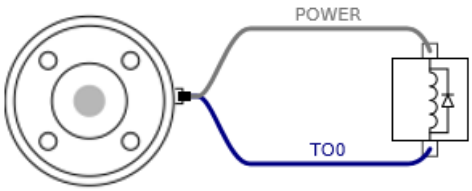
- Der Werkzeuganschluss verwendet NPN für den digitalen Ausgang. Wenn digitaler Ausgang aktiviert wird, wird der entsprechende Anschluss mit GND (ground) verbunden. Wenn digitaler Ausgang deaktiviert wird, wird der entsprechende Ausgangsanschluss geöffnet (open-collector / open-drain). Die elektrischen Spezifikationen sind wie folgt.

-	min.	Nominal	max.	Einheit
Spannung beim Öffnen	0	-	24	V
Strom durch GND	0	-	2.000	mA

- Das unten gegebene Bild zeigt, wie eine Last mit 12V oder 24V eingeschaltet wird, um den digitalen Ausgang zu verwenden. Der Spannungspegel kann im Tool Out-Block angegeben werden.



※ Es wird dringend empfohlen, eine Schutzdiode für induktive Lasten zu verwenden, wie in der folgenden Abbildung gezeigt.

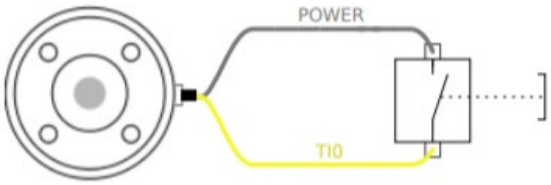


- Bei digitalen Eingängen des Werkzeugs werden PNP- und Pulldown-Widerstände verwendet. Wenn der Eingangsanschluss nicht verbunden ist (floating), wird daher der entsprechende Eingangsanschluss als Low (0) gelesen. Die elektrischen Spezifikationen sind wie folgt.

-	min.	Nominal	max.	Einheit
Eingangsspannung	0	-	24	V
Logische Niederspannung	-	-	9	V
Logische Hochspannung	10	-	-	V

© RAINBOW ROBOTICS Inc. All rights reserved.

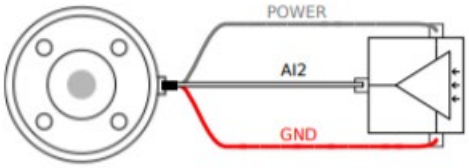
- Die folgende Abbildung zeigt, wie der einfache Schalter für den digitalen Eingang angeschlossen wird.



- Der Analogeingang des Werkzeugs misst die Spannung auf nicht differenzielle (non-differential) Weise. Die Messkategorien sind wie folgt:

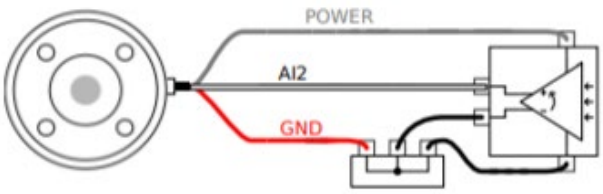
-	min.	Nominal	max.	Einheit
Eingangsspannung	0	-	10	V
Auflösung	-	12	-	bit

- Die folgende Abbildung zeigt, wie ein analoger Sensor mit nicht differenziellen Spannungsausgang an den Werkzeugflansch angeschlossen wird (nicht differenzielle Weise)



© RAINBOW ROBOTICS Inc. All rights reserved.

- Die folgende Abbildung zeigt, wie ein analoger Sensor mit differenziellen Spannungsausgang an den Werkzeugflansch angeschlossen wird. Wenn der negative Ausgang des Sensors mit GND (ground) verbunden wird, funktioniert dies genauso wie beim nicht differenziellen Sensor.



Vorsicht

Vorsicht:

1) Weitere Einzelheiten zu technischen Daten und Kabelanschlüssen finden Sie in Anhang D.
2) Die Querschnittsansicht des Werkzeugflansches ist in Anhang C dargestellt.

- Der Werkzeugflansch unterstützt die serielle RS485-Kommunikation und den folgenden seriellen Kommunikationsstandard.

Baudrate	**9.600, 19.200, 38.400, 57.600, 115.200, 1M, 2M.**
Stop-Bit	1, 2
Parität	k.a., Gerade, Ungerade

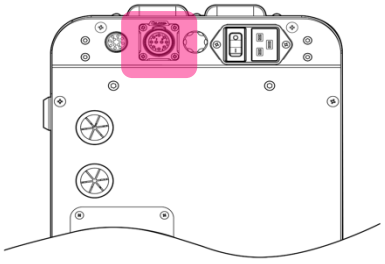
© RAINBOW ROBOTICS Inc. All rights reserved.

4.6 Kabelanschluss

Der Kabelanschluss für die Standsteuerbox wird wie folgt dargestellt.

- Robotersteuerbox mit dem Roboterarmkabel verbinden

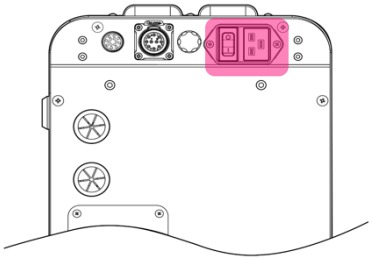
  Verbinden Sie die Buchse mit dem Roboterarm und den Stecker mit der Steuerbox. Achten Sie darauf, dass die Stifte (Pin) im Stecker nicht verbogen werden sollten.



[Anschlussteil für Roboteramkabel]

- Netzkabel an die Robotersteuerbox anschließen

  Schließen Sie das Netzkabel an den Netzanschluss an.



[Anschlussteil für Wechselstromkabel]

Die Spezifikationen des Stromversorgungssystems sind wie folgt.

Eingangsspannung	**100 - 240 VAC**
Eingangsfrequenz	50 - 60 Hz

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

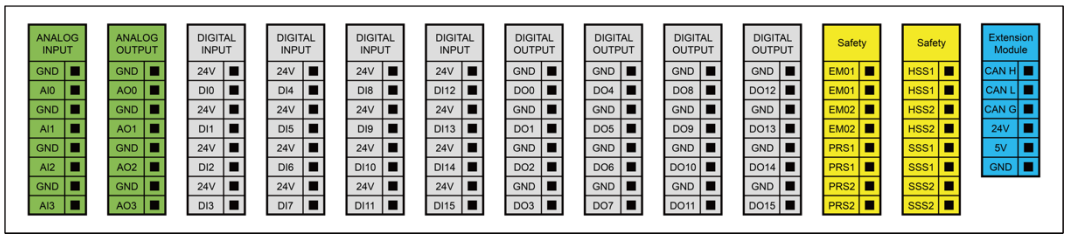
Vorsicht:

1) Ziehen Sie das Roboterkabel, das Netzkabel oder das Kabel des Programmierhandgeräts nicht ab, während der Roboter eingeschaltet ist.
2) Für die Verwendung von Wechselstrom sollten die Peripheriegeräte eine gemeinsame Erdung haben.

© RAINBOW ROBOTICS Inc. All rights reserved.

4.7 Übersicht über Ein/Aus der Robotersteuerbox

Um andere externe Geräte an die Robotersteuerbox anzuschließen, verbinden Sie die Ein/Aus der Steuerbox mit dem entsprechenden Gerät. Ein/Aus der Steuerbox sind so flexibel, dass sie für den Anschluss verschiedener Geräte wie Relais, SPS und Not-Aus-Schalter verwendet werden. Das Layout der elektrischen Schnittstellen in der Robotersteuerbox ist wie folgt.



Die Spezifikationen der Stromversorgung und der digitalen Ein/Aus der Steuerbox lauten wie folgt. Alle digitalen Ein/Aus entsprechen der Norm IEC 61131-2.

※IEC 61131-2: IEC-Standard für programmierbare Steuerungen

Terminal	Parameter	min.	Nominal	max.	Einheit
Digitaler Ausgang					
[Dox]	Strom	0	-	1	A
[Dox]	Spannungsabfall	0	-	0,5	V
[Dox]	Leckstrom	0	-	0,1	mA
[Dox]	Typ	-	PNP		Type
[Dox]	IEC 61131-2	-	1A	-	Type
Digitaler Eingang					
[Dix]	Spannung	-3	-	30	V
[Dix]	AUS-Bereich	-3	-	5	V
[Dix]	EIN-Bereich	11	-	30	V
[Dix]	Strom (11-30V)	2	-	15	mA
[Dix]	Typ		PNP+		Type
[Dix]	IEC 61131-2	-	1	-	Type

Vorsicht

Vorsicht:

1) Schalten Sie vor dem Anschließen der Ein/Aus-Verkabelung die Stromversorgung zur Steuerbox aus. Produktschäden aufgrund von Fahrlässigkeit des Benutzers (Kurzschluss der 24 V-Stromversorgung und falsche Verdrahtung usw.) werden nicht von der Produktgarantie abgedeckt.

© RAINBOW ROBOTICS Inc. All rights reserved.

4.8 Konfiguration der Sicherheitseingänge

Zur Sicherheit der Benutzer müssen alle sicherheitsrelevanten Ein/Aus mit mehreren Sicherungen konfiguriert werden. Sicherheitsvorrichtungen und -ausrüstungen müssen gemäß den Anweisungen in Kapitel 2 Sicherheit und Kapitel 3 Installation installiert werden. Schutzstopp gehört zu Sicherheitseingängen. Not-Aus-Eingang dient zur Not-Aus des Roboters, und der Schutzstoppeingang dient allen Arten des Sicherheitsschutzes des Roboters.

Gefahr

Gefahr:

1) Schließen Sie niemals ein Sicherheitssignal an eine andere PLC als eine Sicherheits-PLC. Die Nichtachtung dieser Warnungen kann dazu führen, dass die Funktion, Sicherheitsstopp' nicht ordnungsgemäß funktioniert, was zu schweren oder tödlichen Verletzungen führen kann. Das Sicherheitssignal und das allgemeine Ein/Aus-Signal müssen getrennt werden.

Warnung

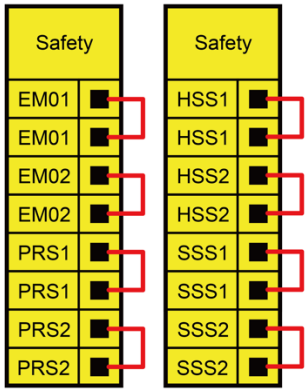
Warnung:

1) Ein- und Ausgänge aller Arten von Sicherheitsfunktionen weisen Redundanzen auf. Es ist erforderlich, den Kanal zu isolieren, damit die Sicherheitsfunktion nicht aufgrund eines Signalausfalls aktiviert wird. Die Sicherheitsfunktion muss vor der Installation des Roboters bestätigt werden. Auch die Sicherheitsfunktion sollte regelmäßig auf Auffälligkeiten überprüft werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

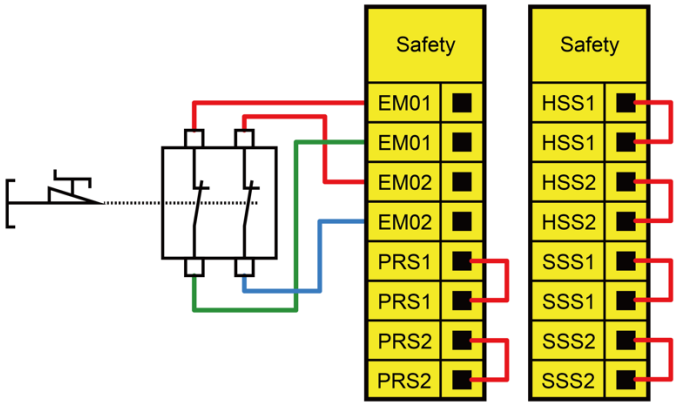
- Anfängliche Sicherheitskonfiguration

  Der Roboter wird mit standardmäßigen Sicherheitskonfiguration ausgeliefert, sodass er ohne zusätzliche Konfiguration verwendet werden kann. Die anfängliche Sicherheitskonfiguration ist wie folgt.



- Schutzstopp und automatischer Neustart

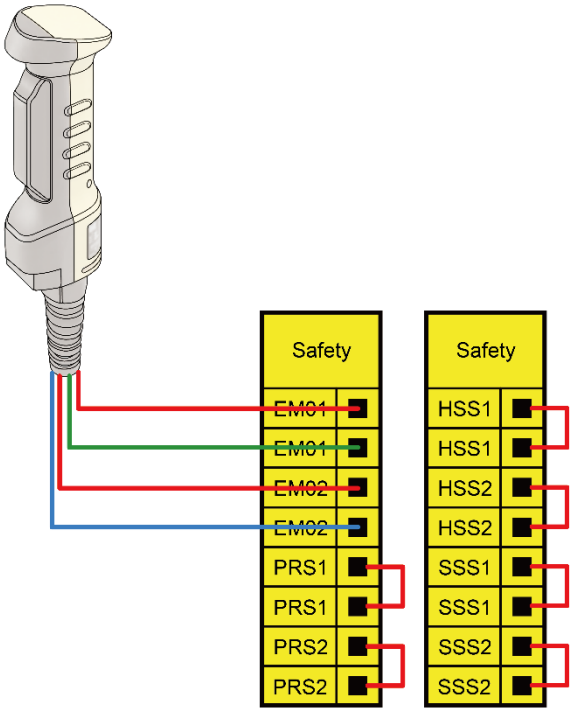
  Ein Beispiel für eine Sicherheitsschutzvorrichtung wäre ein Türschalter, der den Roboter stoppt, wenn die Tür geöffnet wird. Die folgende Abbildung zeigt, wie diese Funktionen konfiguriert werden:



© RAINBOW ROBOTICS Inc. All rights reserved.

- Enabling Device Input (Optional)

  Benutzer kann einen 3-Position-Zustimmungsschalter verwenden, um die Eingangsschnittstelle des Aktivierungsgeräts anzuschließen. Wenn sich der 3-Position-Zustimmungsschalter in der Betriebsposition (Mittlere Position) befindet, beginnt sich der Roboter zu bewegen. Wenn der Benutzer den 3-Position-Zusimmungsschalter loslässt oder drückt, befindet sich der 3-Position-Zustimmungsschalter in der nicht betreibenden Position und der Roboterarm bewegt sich nicht. Der Hersteller stellt die Zustimmungsvorrichtung nicht zur Verfügung und kann optional verwendet werden, wenn der Benutzer die Zustimmungsvorrichtung verwenden möchten. So konfigurieren Sie diese Funktion.



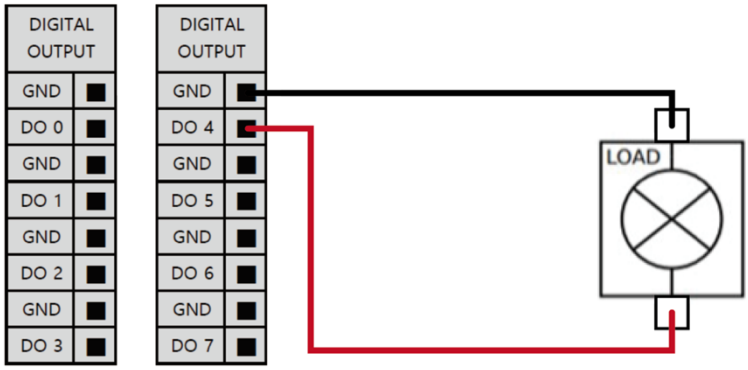
© RAINBOW ROBOTICS Inc. All rights reserved.

4.9 Universelle Digitale Ein/Aus-Konfiguration

Alle Digital-Ein/Aus können als universelle Digital-Ein/Aus verwendet werden. Wenn Sie andere externe Geräte mit dem Roboter verwenden möchten, müssen Sie die Ein/Aus in der Steuerbox mit dem Gerät verbinden. Universelle digitale Ein/Aus können zur Konfiguration von Geräten wie Relais oder PLC-System verwendet werden. In dieser Konfiguration sollte der Ausgang immer 'Low' bleiben, es sei denn, das Programm läuft. Die folgenden Unterpunkte zeigen Beispiele.

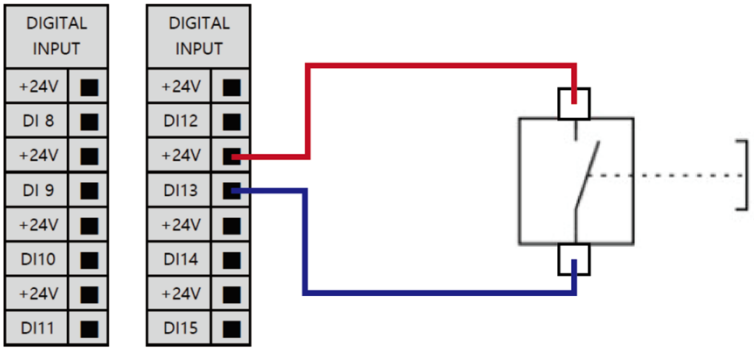
- **Elektrische Laststeuerung mit digitalem Ausgang**

  So steuern Sie die Last mit einem digitalen Ausgang. So konfigurieren Sie es:



- **Digitale Eingangssteuerung über Tasten**

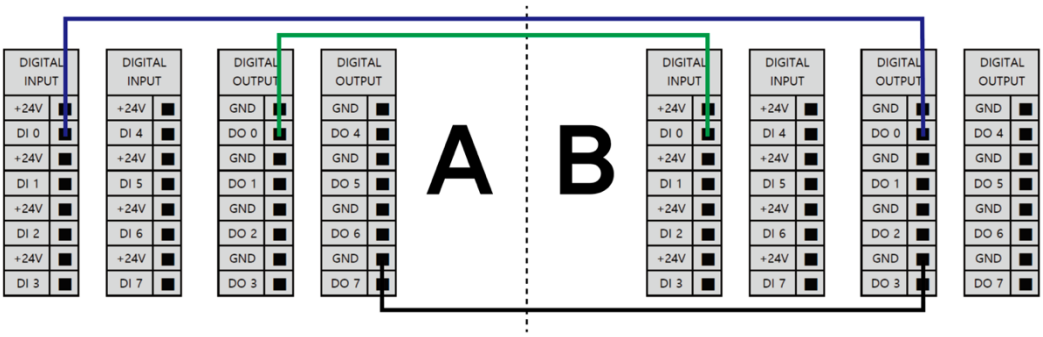
  So verbinden Sie eine einfache Taste mit einem digitalen Eingang. So konfigurieren Sie es:



© RAINBOW ROBOTICS Inc. All rights reserved.

- Kommunikation mit anderem System oder PLC

  Wenn Sie einen gemeinsamen GND verwenden und Ihr Gerät PNP unterstützt, können Sie Ihre digitalen Ein/Aus für die Kommunikation mit anderen Geräten konfigurieren. So konfigurieren Sie es:



Vorsicht:

Detaillierte technische Spezifikationen und Verdrahtungsbeispiele finden Sie in Anhang D.

© RAINBOW ROBOTICS Inc. All rights reserved.

4.10 Universelle Analoge Ein/Aus-Konfiguration

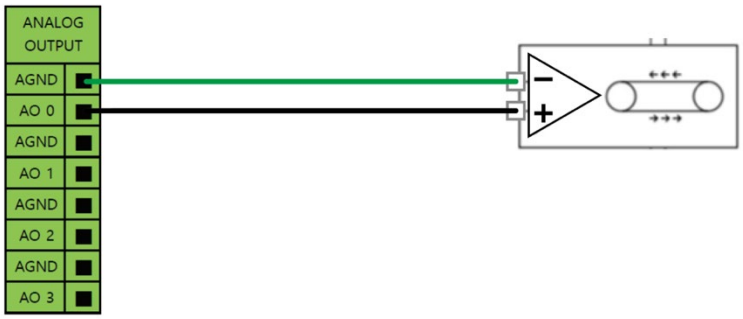
Für eine hohe Zuverlässigkeit werden folgende Methode empfohlen.

- Verwenden Sie den analogen GND, der dem Ein/Aus am nächsten liegt.
- Maschinen- und Robotersteuerbox verwenden denselben GND. Analoge Ein/Aus sind nicht von der Robotersteuerbox isoliert.
- Verwenden Sie geschirmte oder paarweise verdrillte Kabel. Schließen Sie an die GND-Abschirmung der Stromversorgungsklemme (J12) an.

Terminal	Parameter	min.	Nominal	max.	Einheit
Eingang im Spannungsmodus					
AIx - AG	Spannung	0	-	10	V
AIx - AG	Auflösung	-	16	-	bit
Ausgang im Spannungsmodus					
AOx – AG	Spannung	0	-	10	V
AOx – AG	Auflösung	-	16	-	bit

■ Analogausgang

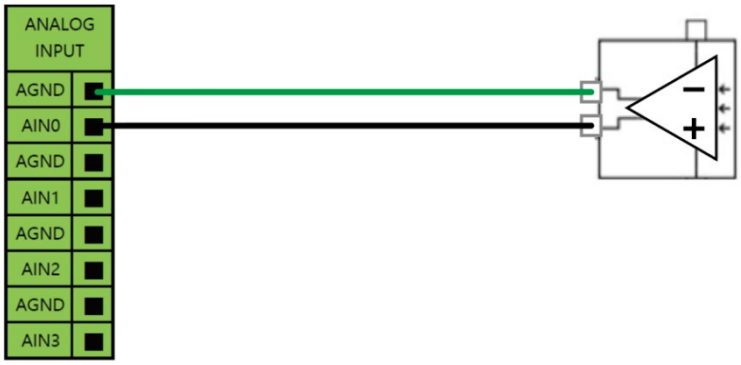
Analogausgänge können als Geschwindigkeitssteuereingänge für Förderbänder verwendet werden. So konfigurieren Sie es:



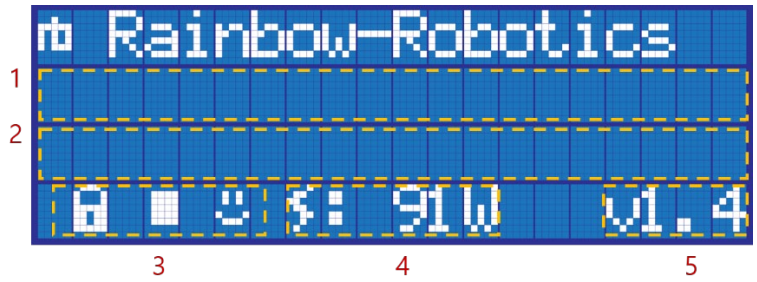
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Analogeingang

Der Ausgangswert eines analogen Sensors kann als analoger Eingang verwendet werden. So konfigurieren Sie es:



■ Beschreibung der LCD-Statusanzeige



1. Benachrichtigungsfeld(1): Ein Benachrichtigungsfeld über den Systemstatus.
   : Please Wait (Es zeigt, dass der PC in der Steuerbox nicht bereit ist.)
   : Normal Operation (Es zeigt, dass der PC in der Steuerbox bereit ist.)
2. Benachrichtigungsfeld(2): Ein Benachrichtigungsfeld über die Bewegung und den Status des Roboters.

4. Stromverbrauch: Gesamter Stromverbrauch in Watt (W).
5. Infos zur Systemversion: Informationen zur Systemversion.

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 5 Starten

5.1 Ein/Aus der Steuerbox

So schalten Sie die Robotersteuerbox ein / aus.

- Ein-/Ausschalten der Standsteuerbox

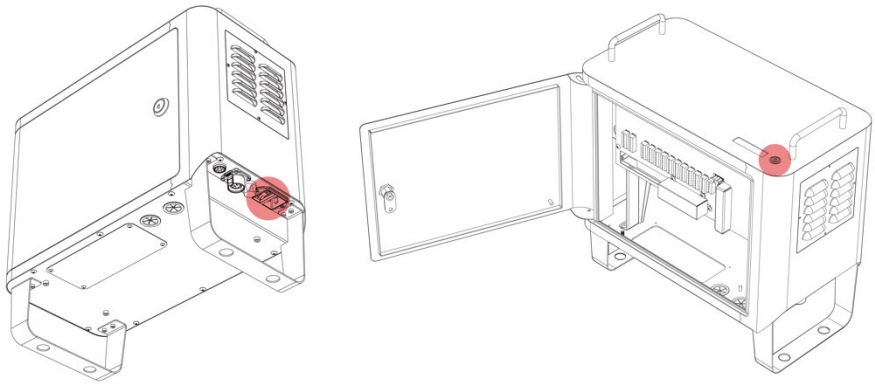
  Wechselstrom wird durch Drücken des Wechselstromschalters an der Unterseite der Steuerbox angelegt.

  Drücken Sie den Hauptnetzschalter auf der Oberseite der Steuerbox, um die Hauptstromversorgung einzuschalten.

  "Please Wait" wird auf dem LCD-Fenster der Robotersteuerbox angezeigt.

  Dies zeigt, dass sich die Steuerbox im anfänglichen Zustand befindet.

  Die LCD-Meldung ändert sich zu "Normal Operation", wenn die Steuerbox in den aktivierten Zustand versetzt wird.



  Halten Sie den Hauptschalter auf der Oberseite der Steuerbox gedrückt, um die Stromversorgung auszuhalten.

Vorsicht:

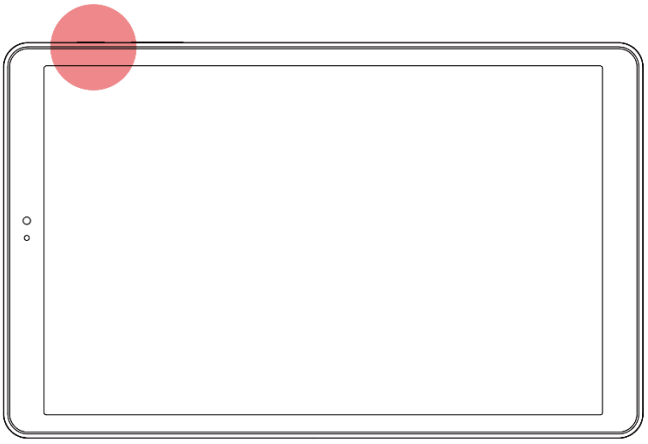
1) Unsere Steuerbox verwendet AC 100 - 240V einphasig (50~60Hz).

© RAINBOW ROBOTICS Inc. All rights reserved.

5.2 Tablet-PC einschalten

Wenn Sie den optionalen Tablet-PC, der vom Hersteller für die Steuerbox bereitgestellt wird, auswählen, werden der unten gezeigte Tablet-PC und die Hülle mitgeliefert. Drücken Sie den Netzschalter oben links am Tablet-PC, um den Tablet-PC einzuschalten.

- Für Standsteuerbox



Vorsicht:

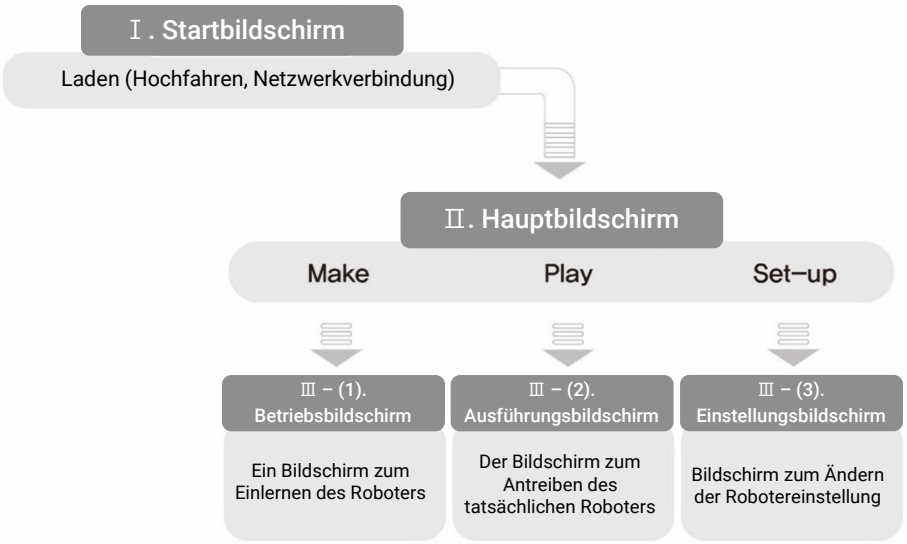
1) Überprüfen Sie, ob der Tablet-PC mit der Steuerbox verbunden ist. Und führen Sie die vom Hersteller bereitgestellten Anwendung aus. Vermeiden Sie unnötige Bedienung am Bildschirm während des Systemstarts. Es kann zu Problemen mit dem System führen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 6 Übersicht über Software

6.1 Grundkonfiguration des UI-Programms

Das UI-Programm (Benutzerschnittstelle) ist wie folgt in drei Bildschirme unterteilt, und durch jeden Bereich kann der Benutzer die erforderliche Stufe betreten.

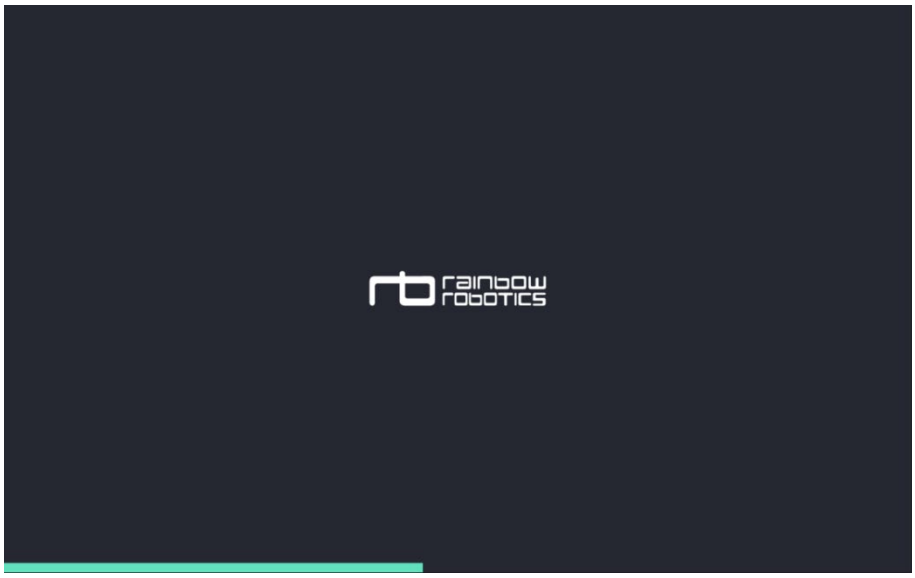


© RAINBOW ROBOTICS Inc. All rights reserved.

6.2 Startbildschirm

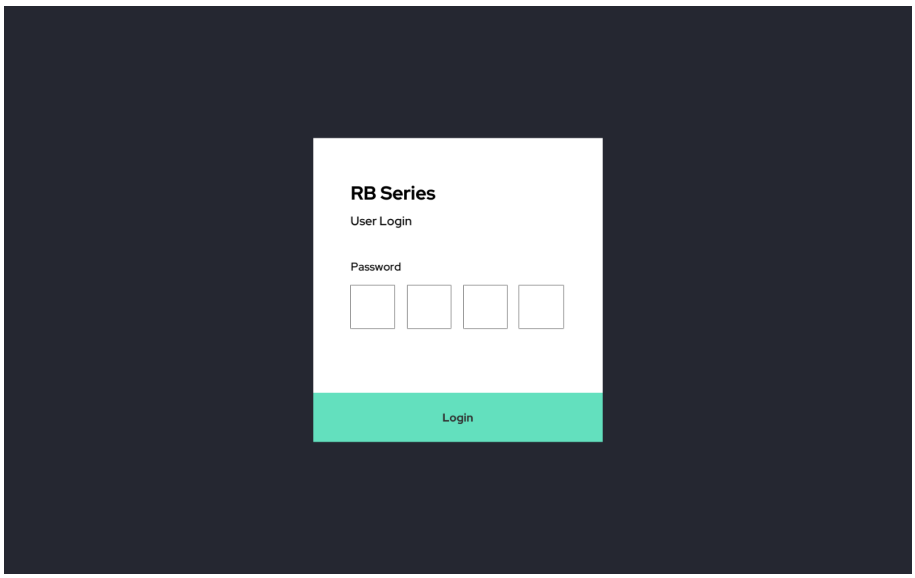
- Einführung (Intro)

  Der Ladevorgang erfolgt über den Intro-Screen auf dem Startbildschirm.



- Einloggen (Bei Auslieferung ist das Passwort: 0000 eingestellt)

  Login-Bestätigen beim Starten. Passwort und automatische Login-Einstellungen werden in **“Setup-System-Password”** vorgenommen.



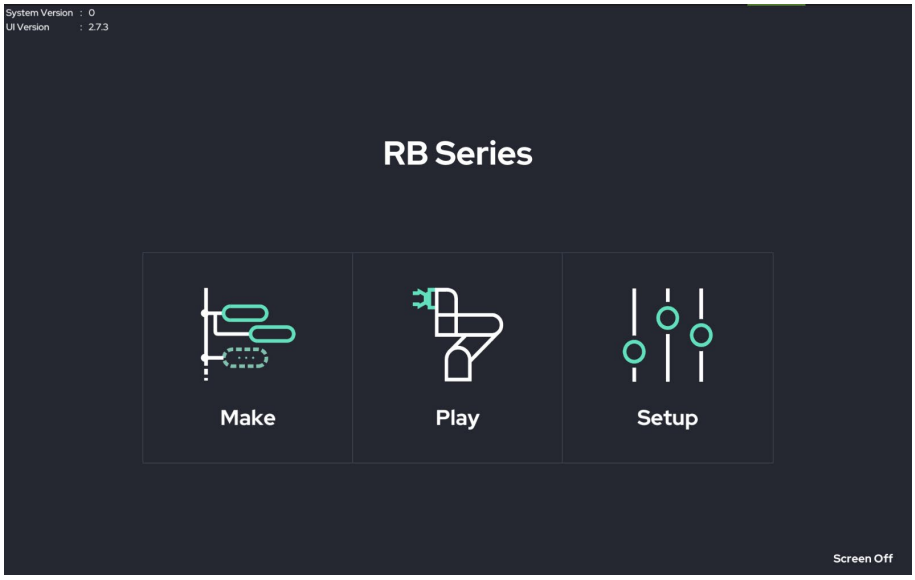
© RAINBOW ROBOTICS Inc. All rights reserved.

6.3 Hauptfenster

Die Benutzeroberfläche besteht aus drei Hauptmenüs, wie unten gezeigt.

- Make: Der Bereich, in dem das eigentliche Programm (Ablauf) erstellt wird.
- Play: Der Bereich, in dem das erstellte Programm einfach ausgeführt wird.
- Setup: Der Bereich zum Einstellen verschiedener Parameter.

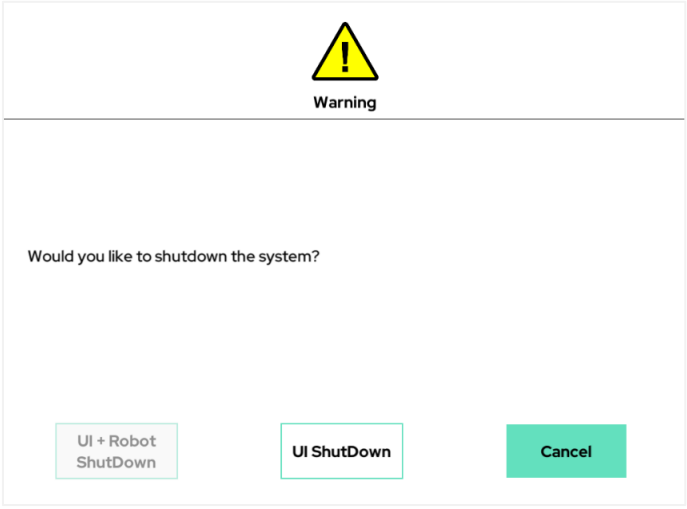
Auf dem Hauptbildschirm können Sie über drei Menüs in die Phase der Erstellung einer Bewegung des Roboters (Make), der Ausführung eigentlicher Bewegung (Play) oder der Einstellung der Arbeitsumgebung (set-up) eintreten.



© RAINBOW ROBOTICS Inc. All rights reserved.

- Ausschalten

  Wenn Sie den Netzschalter in der unteren rechten Ecke berühren, wird das Fenster zum Herunterfahren angezeigt. Und wenn Sie UI-Shutdown drücken, wird das Programm beendet. Wenn der Roboter aktiviert ist und die Steuerbox mit dem UI-Tablet-PC verbunden ist, wird auch die Stromversorgung des Roboters ausgeschaltet.



© RAINBOW ROBOTICS Inc. All rights reserved.

6.4 Betriebsbildschirm (MAKE)

■ Make

Der Make-Bildschirm ist das interaktive Menü zu Teaching(Programmieren) des Roboters. Mit den Symbolen oben auf dem Bildschirm können Sie den Roboter anlernen. Der Roboterarm kann mit Symbolen auf der rechten Seite des Bildschirms bewegt werden. Die Lehrinhalte können mit den Symbolen auf der linken Seite bearbeitet werden.

- Linke Symbole: Kopieren & Einfügen, Speichern, Löschen, Anmerkungen hinzufügen usw.
- Rechte Symbole: Joggen(Jog/Jogging) bzw. weitere Einstellungen.
- Mittlere Symbole: Programmfunktionen usw.
- Untere Symbole: Speichern/Laden, Wiedergabe, Anpassung der Bewegungsgeschwindigkeit usw.

※ Weitere Einzelheiten zu Symbolen und Konfiguration finden Sie in Kapitel 7.

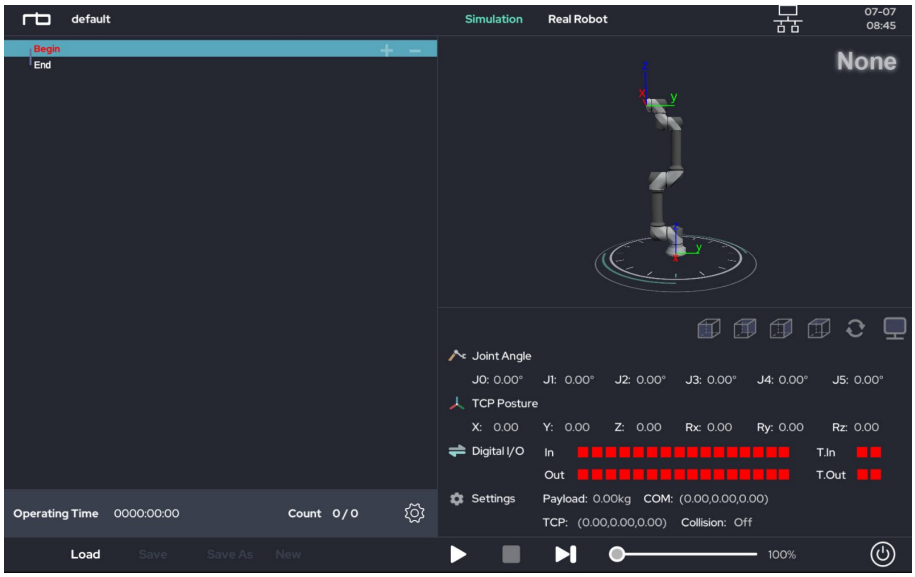
※ Im Make-Menü (Manueller Modus) bewegt sich der Roboter nur, wenn eine Taste gedrückt gehalten wird. Einstellwerte für diese Funktion können im Setup-Schnittstelle verändert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

6.5 Wiedergabe-Bildschirm (PLAY)

- Play (Wiedergabe)

  Der Wiedergabebildschirm ermöglicht nur die physische Bewegung des Roboters und dem Benutzer, ein Lehrprogramm zu laden und ausführen. Die Anzahl der Wiederholungen kann im Setup-Schnittstelle eingestellt werden. In der unteren linken Ecke wird die kumulierte Zeit seit der ersten Wiedergabe angezeigt.



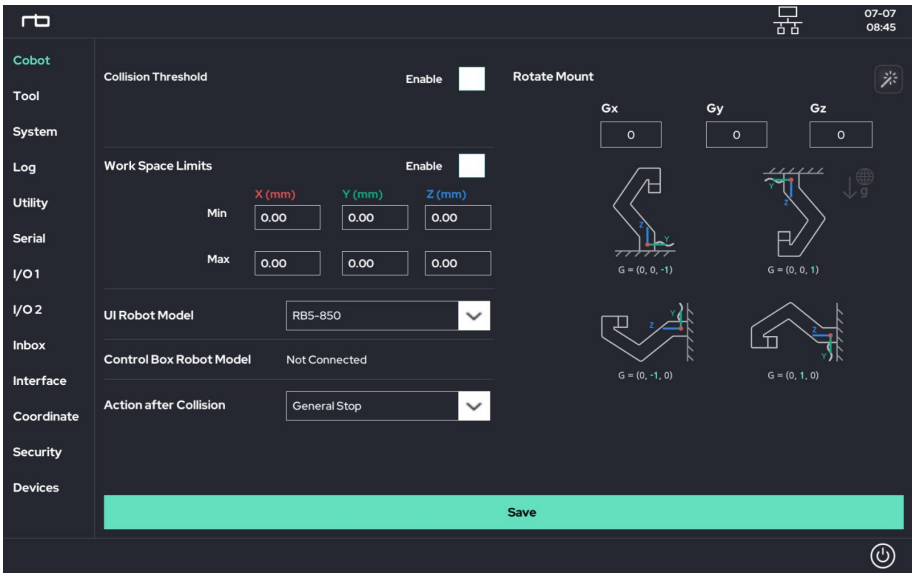
※ Weitere Einzelheiten finden Sie in Kapitel 8.

© RAINBOW ROBOTICS Inc. All rights reserved.

6.6 Einstellung (SETUP)

■ Einstellung (Setup)

Auf dem Bildschirm können verschiedene Einstellungen wie Empfindlichkeit für Kollisionserkennung, Konfiguration der Roboterinstallation, Betriebsbereich, Werkzeugeinstellungen, Systemprotokoll, Ein/Aus-Funktion oder Koordinationssystem usw. zur Verwendung des Roboters vorgenommen werden.



※ Weitere Einzelheiten finden Sie im Kapitel 9.

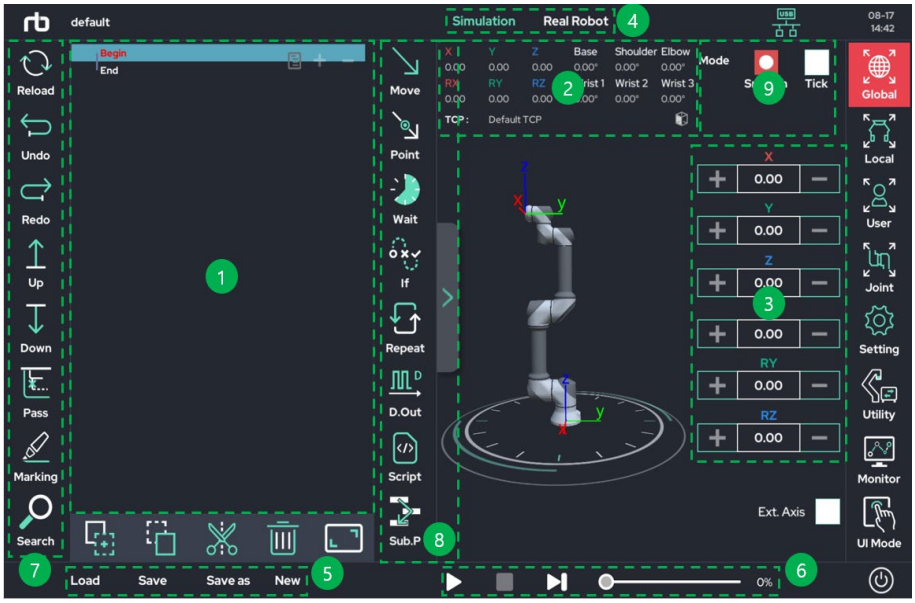
© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 7 Programmieranleitung

7.1 Beschreibung von Symbolen und Bedienbildschirm

■ Make

Die Elemente, aus denen der Bildschirm besteht, werden dargestellt.



Nr.	Beschreibung
①	Programmliste in Baum-Form wird angezeigt.
②	Der Winkel jedes Gelenks vom Roboterarm und die kartesische Koordinatenposition der TCP werden angezeigt
③	TCP-Jog: Kartesische Koordinatenposition kann geändert werden. (Basis/ Werkzeug/benutzerdefiniertes Koordinatensystem wählbar) Gelenk(Joint)-Jog: Der Winkel jedes Gelenks kann geändert werden.
④	Schaltfläche zum Wechseln in den Simulations- oder Realmodus. ※Real-Modus muss ausgewählt werden, um den Roboterarm anzutreiben.
⑤	Gespeicherte Projekte können importiert werden und neue Projekte werden erstellt werden.
⑥	Mit der Schaltfläche können angelernte Programme gestartet oder gestoppt werden. Bewegungsgeschwindigkeits-Einstell-Leite ist vorhanden.
⑦	Es befinden sich verschiedene Bearbeitungswerkzeuge wie Kopieren/ Einfügen/ Anmerkungen usw.
⑧	Verschiedene Programmfunktionen (Befehle/Aktionen) sind angeordnet. Wenn Sie auf den Pfeil rechts klicken, werden weitere Funktionen angezeigt.
⑨	Eine Schaltfläche zum Auswählen des Jog-Modus zwischen Smooth-Modus und Tick-Modus.

© RAINBOW ROBOTICS Inc. All rights reserved.

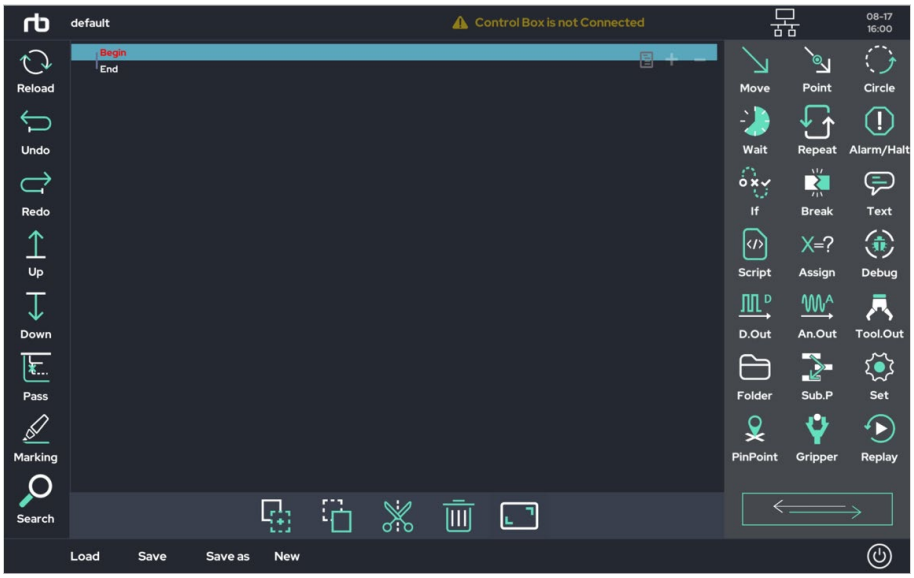
※ Teaching: Der Roboterarm ist so zu programmieren, dass er wie vom Benutzer gewünscht bewegt wird.

※ TCP(Tool Center Point): Der Punkt, der im Koordinatensystem der Maschinenverbindung für eine bestimmte Anwendung als Werkzeugarbeitspunkt definiert ist und der Ursprung des Endeffektors ist.

[Grundansichtsmodus]

[Ansichtsmodus mit erweiterten Symbolen]

© RAINBOW ROBOTICS Inc. All rights reserved.



[Programm-Modus]

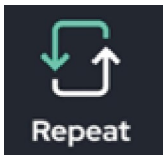
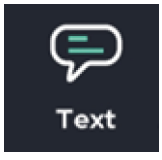
© RAINBOW ROBOTICS Inc. All rights reserved.

- Beschreibung der beim ‘Teaching‘ verwendeten Symbole (Funktionssymbole oben auf dem Make-Bildschirm)

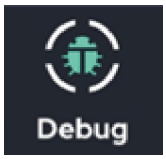
※ Detaillierte Beschreibung der einzelnen Funktionen wird in einem nachfolgenden Kapitel erläutert.

Symbol	Beschreibung
	Die Bewegungseigenschaft des Roboters wird eingestellt. In MoveJ wird jedes Gelenk mit einem bestimmten gewünschten Winkel bewegt. In MoveL wird der TCP linear mit bestimmten gewünschten TCP-Werten bewegt. Darüber hinaus sind insgesamt 7 Bewegungen vorhanden, darunter MoveJB, MoveLB, MovePB, MoveJL und MoveITPL, die zwei Grundbewegungen anwenden können.
	Ein Unterelement von Move. Die Zielwerte der Bewegung werden festgestellt. In MoveJ und MoveJB kann ein gewünschter Gelenkwinkel eingestellt werden In MoveL, MoveLB, MovePB, MoveJL und MoveITPL kann eine gewünschte TCP-Position und Konfiguration (x, y, z, Rx, Ry, Rz) in kartesischen Koordinaten eingestellt werden.
	Abgesehen von Move-Funktion wird eine Funktion für Kreisbewegungen angeboten. Es gibt einen Modus, der einen Bogen zeichnet, der durch 3 Punkte verläuft, indem der Startpunkt und zwei Punkte genommen werden. Es gibt einen weiteren Modus, der einen Kreis zeichnet, indem der Mittelpunkt und die Rotationsachse festgestellt werden.
	Der Roboter kann nach vom Benutzer festgelegten bestimmten Bedingungen kurz angehalten werden. 3 Modi sind vorhanden. Eine Funktion zum Warten für eine bestimmte Zeitspanne, eine Funktion zum Warten, während die Bedingung zutreffend ist, und eine Funktion zum Beenden des Wartens, wenn die Bedingung zutreffend ist.
	Eine konditionale Anweisung wird erstellt. Benutzer kann abhängig von Bedingungen Bewegungsprogrammzweige einstellen, indem er If/else if /else verwendet.

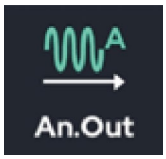
© RAINBOW ROBOTICS Inc. All rights reserved.

	Dies ist Switch-Anweisung. Verzweigungen können eingerichtet werden, sodass der Roboter je nach Bedingungen verschiedene Befehle ausführen kann. Switch / Case-Anweisung kann verwendet werden.
	Ein Unterprogramm wird unter bestimmten Bedingungen wiederholt. 3 Modi sind vorhanden. Eine Funktion zum Wiederholen einer bestimmten Zahl, eine Funktion zum Wiederholen, solang eine bestimmte Bedingung zutreffend ist, und eine Funktion zum Beenden der Wiederholung, wenn die Bedingung zutreffend ist.
	Mit dieser Funktion wird eine Wiederholung-Anweisung zwangsweise beendet. Sie wird unterhalb der Wiederholung-Anweisung(Repeat-Funktion) verwendet, und selbst wenn eine bestimmte Bedingung in der Wiederholung-Anweisung deklariert ist, wird die Wiederholung-Anweisung bei Verwendung der Break-Funktion im Voraus beendet.
	Mit dieser Halt-Funktion wird das Programm beendet. Wenn das Programm unter bestimmten Bedingungen beendet werden muss, kann diese Funktion verwendet werden.
	Mit dieser Funktion können Variablen deklariert werden, die aus vier Typen bestehen: Einem Variablentyp, bei dem eine einzelne Zahl gespeichert ist, einem Array-Typ, bei dem mehrere Zahlen gespeichert sind, und einem Zeichenfolgetyp, bei der eine Zeichenfolge gespeichert ist.
	Mit dieser Funktion kann der Benutzer manuell programmieren. Es wird normalerweise verwendet, wenn eine benutzerspezifische Berechnung oder Substitution erforderlich ist.
	Ein Kommentar oder eine Notiz kann dem Programm hinzugefügt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

	Mit dieser Funktion werden die Befehle des erstellten Programms in den Unterpunkten des Ordners gebündelt. Jeder spezifische Name wird basierend auf der Ordnerfunktion vergeben, um die Verwaltung zu erleichtern.
	Mit dieser Funktion kann die Programme anderer Benutzer(Teaching-Datei) importiert werden und in das aktuelle Programm eingefügt werden (Sub-Routine-Funktion). Die importierten Programme können im aktuellen Programm nicht bearbeitet werden. Sind Änderung erforderlich, muss das jeweilige Projekt separate geöffnet werden.
	Eine Pre-Program-Funktion, die den Inhalt des Untergeordneten nur einmal zum ersten Mal ausführt. Wie im Play-Modus werden beim Weiterlaufen des Programms die unter dem Pre-Program deklarierten Inhalte nur einmal zum ersten Mal ausgeführt. Bei Einzelbefehlen wie Variablendeklaration oder Kommunikationsverbindung können Sie dies in der unteren Ebene des Pre-Programs verwalten.
	Thread-Funktion, Die Unterelement der Thread-Funktion laufen parallel zum Hauptprogramm. Befehl, die sich auf die Roboterbewegung beziehen, können jedoch nicht unter den Thread gestellt werden.
	Mit dieser Funktion wird die Alarm-Popup-Meldung angezeigt. Sie können dem Benutzer eine Bestätigung anfordern, indem Sie eine Alarmmeldung in einem bestimmten Zustand oder Abschnitt platzieren.
	Debug-Funktion zum Debuggen interner Werte. Wenn Sie einen gewünschten Variablen oder internen Parameter anfordern, können Sie den Wert in einem Popup-Format überprüfen.
	Mit der Set-Funktion können im Setup-Menü eingestellte Einstellparameter innerhalb des Programms separate eingestellt werden. Es hat die Funktion, bestimmte Einstellparameter während der Programmausführung zu ändern.

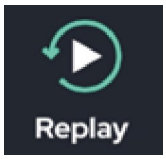
© RAINBOW ROBOTICS Inc. All rights reserved.

	Ein Befehl zur Steuerung des digitalen Ausgangs der Steuerbox. Diese Funktion gibt das ausgewählte Signal über den ausgewählten Port von 16 Ports aus. Jeder Port kann 3 Einstellungen haben: High-Signal, Low-Signal, Bypass.
	Dieser Befehl steuert den analogen Ausgang der Steuerbox. Diese Funktion gibt die ausgewählte Spannung über den ausgewählten Port unter den 4 analogen Ports aus. Jeder Port kann mit einem Spannungsbereich von 0 bis 10 V ausgeben.
	Der Werkzeugflansch hat zwei digitale Ausgänge. Signale von zwei digitalen Ausgängen können zugeordnet werden. Außerdem kann die Höhe der vom Werkzeugflansch auszugebenden Spannung (0V oder 12V oder 24V) gemeinsam eingestellt werden.
	Greifer-Funktion, die zur Verwendung von Greifer von Drittanbietern vorgefertigt sind. Dies ist eine Funktion, die automatisch hilft, den Greifer einfach zu verwenden, indem Sie den Greifer auswählen, den Sie verwenden möchten, und die vom Greifer bereitgestellte Funktion auswählen.
	Funktion für RS485/232-Ausgang über Werkzeugflansch / Steuerbox. Sie können die Ausgabe im ASCII-Modus oder im Hex-Modus ausgeben. Das entsprechende Kommunikationsprotokoll wird im Setup-serial-Menü eingestellt.
	Diese Funktion dient der Socket-Kommunikation. Bietet Funktionen zum Herstellen einer Verbindung zu einem bestimmten Server durch Öffnen eines Sockets., zum Senden einer Anforderungsnachricht an den Server und zum Empfangen bestimmter Daten vom Server. Die Socket-Kommunikation kann bis zu 5 separate Server verbinden.
	Modbus-Client-Funktion bietet die Möglichkeit, Daten von einer bestimmten IP/Adresse anzufordern und die zurückgegebenen Daten abzurufen. Sie können den Zeitraum und das Format der Datenanforderung festlegen. Protokolle im Zusammenhang mit dem in diesem Produkt bereitgestellten Modbus-Server werden separat bereitgestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

	Förderer-Tracking-Funktion. Nach dem Einstellen der Fördergeschwindigkeit und -richtung werden die Teilbewegungen der Förderfunktionen (MoveL, MoveLB, MovePB, MoveITPL, Circle) dem Förderfluss hinzugefügt.
	Post-Programmfunktion, die die Funktion definiert, die als Endbearbeitung nach dem Ende des Programms ausgeführt werden soll. Die unten deklarierten Funktionen werden nach Programmende sequenziell ausgeführt. Befehle, die sich auf den Betrieb beziehen, können in der unteren Ebene nicht verwendet werden.
	Eine Funktion zum Einfügen anderer vorgefertigten Programmdateien (Teaching File) in bearbeitbarer Form in das aktuelle Dokument. Sie ähnelt der Sub.P-Funktion aber im Fall von Sub.P, wenn es sich um eine Funktion zum einfachen Aufrufen des entsprechenden Programms handelt, ist die Template-Funktion eine Funktion zum Kopieren und Bringen eines anderen Programms in eine Form, di es sein kann gegenüber dem aktuellen Programm geändert.
	Dies ist eine Funktion zum Deklarieren von Variablen (Einzelvariable, Array, Punktvariable usw.), die in Echtzeit beobachtet werden sollen, während das Programm ausgeführt wird. In der Monitor-Funktion deklarierte Variablen können in Echtzeit beobachtet, während das Programm läuft, indem Sie auf da Monitor-Symbol auf der rechten Seite der Make-/Play-Seite klicken.
	Mit dieser Funktion können sich wiederholende Aktion definiert werden. Durch das Definieren von Informationen über den Raum, in dem die sich wiederholenden Aktion auszuführen sind, sowie durch das Definieren der sich wiederholenden Aktionen, die an jedem Ort auszuführen sind, führt der Roboter an jedem Punkt dieselbe Aktion aus. Durch diese Funktion kann eine Palettierung realisiert werden.
	Eine punktgenaue Funktion zum Speichern und Variieren spezifischer Haltungs-/Positionsinformationen. Es bewegt sich nicht in diese Position. Es ist möglich, den Position-/Haltungswert einer bestimmten Haltung als Variable zu speichern und dann auf andere Bewegungsfunktionen zu beziehen (aufzurufen) und zu verwenden.

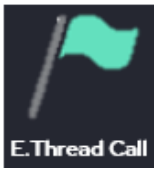
© RAINBOW ROBOTICS Inc. All rights reserved.

	Mit dieser Funktion springt der Programmablauf an eine bestimmte Stelle/Zeile. Kehren Sie mit einer bestimmten Zeilennummer zur Ausgangsposition zurück oder der Programmablauf kann diskret mit bestimmten Zeilennummern gesteuert werden.
	Dies ist eine Funktion zur Wiedergabe der aufgezeichneten Bewegung durch die Bewegungsaufzeichnung. Durch Direkt-Teaching usw. aufgezeichnete Bewegungen im J- oder L-Typ wiedergegeben werden.
	Dies ist eine dedizierte Funktion für den Webbetrieb(Weaving). Bewegungen wie Move Loder Circle, die unter der Webbewegung eingestellt sind, werden durch Kombinieren der eingestellten Weaving-Optionen ausgeführt.
	Eine Funktion dient zur Kraftsteuerung. Wählen Sie die gewünschte Kraftrichtung und das Koordinatensystem aus. Die in der Kraftsteuerung unterlagerten Bewegungen erhalten automatisch die Kraftsteuerungsfunktion
	Spezielle Makrofunktion für das Lichtbogenschweißen. Dies ist eine dedizierte Funktion, die Funktionen bündelt, die durch allgemeine D.out- oder Wait-Funktionen implementiert werden können, damit sie in Form von Makros schnell verwendet werden können. Die Einstellung des Lichtbogenschweißgeräts werden im Gerät auf der Setup-Seite vorgenommen.
	Die TCP-Einstellungsfunktion ist eine Funktion zum vorübergehenden Ändern des TCP-Werts durch Abrufen des zuvor gespeicherten TCP-Werts, während das Programm läuft. Sie können TCP-Werte vorab in der Tool-Liste auf der Setup-Seite speichern.
	Die Handbetriebsfunktion ist eine Funktion, die das laufende Programm anhält und ein Direkt-Teaching ermöglicht.

© RAINBOW ROBOTICS Inc. All rights reserved.

	G-Code-Funktion kann verwendet werden, wenn eine G-Code-Datei im angegebenen Ordner abgelegt wird. Der Roboter implementiert den Pfad des entsprechenden G-Codes.
	Diese Funktion ist integriert, sodass andere Produkte wie HMI und PLC problemlos verwendet werden können. Sie können das Produkt wählen, das Sie verwenden möchten, und Sie können detaillierte Funktionen auswählen, um das ausgewählte Produkt zu verwenden.
	Diese Funktion ist verfügbar, wenn Ein/Aus-Erweiterungsmodul hinzugefügt wird. Sie können den Digital-/Analogausgang des Ein/Aus-Erweiterungsmoduls einstellen, und die Einstellungsmethode ist die gleiche wie beim bestehenden D.-Ausgang und An.-Ausgang.
	Die Benutzereingabefunktion ist eine Funktion, die es dem Benutzer ermöglicht, die Werte von Variablen/Arrays/Point/Strings/Global/ ROM einzugeben und zu ändern, indem die Programmausführung vorübergehend gestoppt wird. Es ist möglich, den Wert entsprechend der Situation des Benutzers zu ändern oder zu ignorieren und zu überspringen.
	Diese Funktion ist für Schweißanwendungen vorgesehen. Es erkennt die Bewegung des Grundmaterials und spiegelt die Bewegung und Bewegungsausrichtung wider, damit das Schweißen fortgesetzt werden kann.
	Mit dieser Funktion wird der Roboter zur Startposition des Programms oder zur Position des Gelenk-0-Punkt des Roboters bewegt. Die Bewegungsmethode kann zwischen MoveJ und MoveL ausgewählt werden.
	Diese Funktion dient zur Verwendung des digitalen Schweißgeräts. Nachdem Sie die Marke des digitalen Schweißgeräts ausgewählt haben, das Sie verwenden möchten, können Sie das digitale Schweißgerät einfach verwenden, indem Sie den Modus und die Option auswählen.

© RAINBOW ROBOTICS Inc. All rights reserved.

	Anders als der normale/Non-Stop-Thread, der gleichzeitig mit dem Programm läuft, ist es eine Funktion, die das Normal-Event-/Not-Stop-Thread aufruft, das ausgeführt wird, wenn es vom Hauptprogramm aufgerufen wird.
	Diese Funktion wird verwendet, um das Hauptprogramm auf ein anderes Projekt umzustellen.
	Funktion, das Robotergelenk und die Hilfsachse gleichzeitig zu betreiben.
	Funktion zur Steuerung der Hilfsachse (zusätzlicher Motoren) außer dem Roboter. Bis zu 6 können die Hilfsachsen hinzugefügt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Symbolbeschreibung – Bearbeitung (linke Seite des Make-Bildschirms)

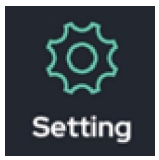
Symbol	Beschreibung
	Die aktuell geöffnete Datei wird erneut geladen. ※ Wenn Sie die Schaltfläche 'Reload' drücken, ohne, Save' drücken, verlieren Sie die zuletzt geöffnete Datei. Achten Sie darauf.
	Die letzte Aktion wird rückgängig gemacht. Bis zu 50-mal möglich.
	Mit dieser Funktion wird Ihr letztes Rückgängigmachen noch rückgängig und diese Funktion kann bis zu 50-mal verwendet werden.
	Befehle werden um ein Feld nach oben verschoben.
	Befehle werden um ein Feld nach unten verschoben.
	Diese Anmerkungsfunktion verhindert, dass der ausgewählte Befehl ausgeführt wird. Kommentierte Befehle sind im Programm vorhanden, werden aber nicht ausgeführt.
	Mit der Funktion wird die gewünschte Programmzeile hervorgehoben (markiert), damit wichtige Programmzeile unterstrichen werden können.
	Nach anderen im Programm verwendeten Funktionen werden gesucht, die nur auf Englisch durchsuchbar sind.

© RAINBOW ROBOTICS Inc. All rights reserved.

	Ausgewählte Befehle werden kopiert und können an eine andere Stelle eingefügt werden.
	Kopierte oder geschnittene Befehle werden an die ausgewählte Stelle eingefügt.
	Ausgewählte Befehle werden geschnitten. Geschnittene Befehle können an eine andere Stelle eingefügt werden.
	Der ausgewählte Befehl wird gelöscht.
	Der Bearbeitungsmodus wird in den Zoommodus gewechselt.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Bewegungen des Roboterarms und andere Funktionen (rechte Seite des Make-Bildschirms)

Symbol	Beschreibung
	Die Position des TCP kann basierend auf dem an der Basis fixierten globalen Koordinatensystem verschoben werden.
	Die Position des TCP basierend auf dem lokalen Koordinatensystem (Werkzeugkoordinatensystem), das am TCP fixiert ist.
	Mit dieser Funktion wird die Position des TCP basierend auf dem benutzerdefinierten Koordinatensystem (Benutzerkoordinate).
	Mit dieser Funktion kann jedes Gelenk des Roboterarms separate bewegt werden.
	Dies ist eine Sammlung von häufigen Sonderfunktionen. Die Status- und Einstellwerte wie Ein/Aus-Statusinformationen des Systems. Einstellung von Benutzerkoordinaten oder Strom-/Temperaturinfos des Roboters können angezeigt werden.
	Dies ist eine Sammlung von Einstellungen wie Benutzerkoordinatensystem, automatische TCP-Suche und andere einfach zu verwendende Einstellungen mit dem Jog-Modus. Diese Einstellungen können standardmäßig auch im Setup-Menü vorgenommen werden.
	Durch die Monitor-Funktion werden die Werte ausgewählter Variablen auf dem Fenster echtzeitig beobachtet. Neben den ausgewählten Variablen werden auch häufig zu überprüfende Systemvariablen angezeigt.
	Mit dieser Funktion kann der Benutzer den UI-Modus auswählen. Benutzer können den UI-Modus basierend auf ihrem Level und Ihrer Umgebung auswählen.

© RAINBOW ROBOTICS Inc. All rights reserved.

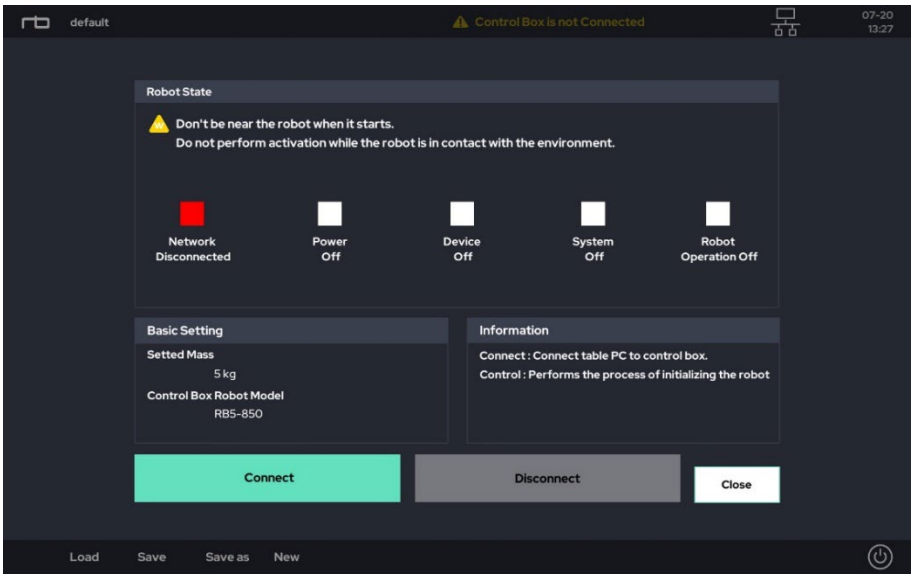
Systemfunktionstasten

Symbol	Beschreibung
	Taste zum Navigieren zum Startbildschirm und zu anderen Seiten. (Das befindet sich in der oberen linken Ecke).
	Die Benutzeroberfläche(UI) wird ausgeschaltet. Wenn der Tablet-PC mit dem Roboter verbunden ist, wird der Roboter ebenfalls ausgeschaltet. Die Bildschirmsperrfunktion ist hier enthalten.

© RAINBOW ROBOTICS Inc. All rights reserved.

7.2 Teaching-Umgebung schaffen

- Verbinden des Tablet-PCs mit der Steuerbox



Drücken Sie die Connect-Schaltfläche auf dem Bildschirm ‘Robot State’ , um den Tablet-PC mit der Robotersteuerbox zu verbinden.

- Connect-Schaltfläche: Die Kommunikation zwischen Steuerbox ↔ Tablet-PC wird verbunden.

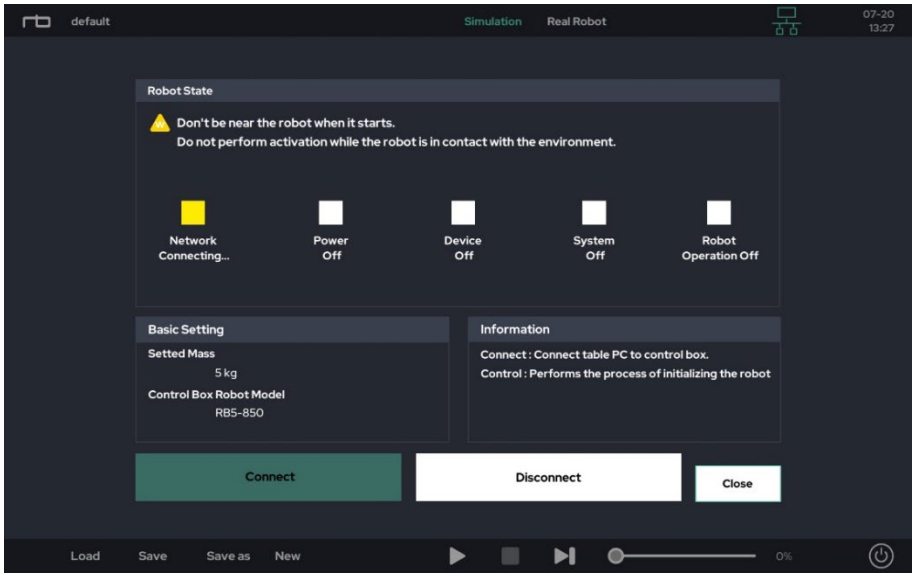
Vorsicht

Vorsicht:

1) Stellen Sie sicher, dass die Steuerbox eingeschaltet und der Not-Aus-Schalter ausgeschaltet ist Wenn die Steuerbox nicht eingeschaltet ist, wird ‚Device Off‘ rot und der Bildschirm kehrt zum obigen zurück.

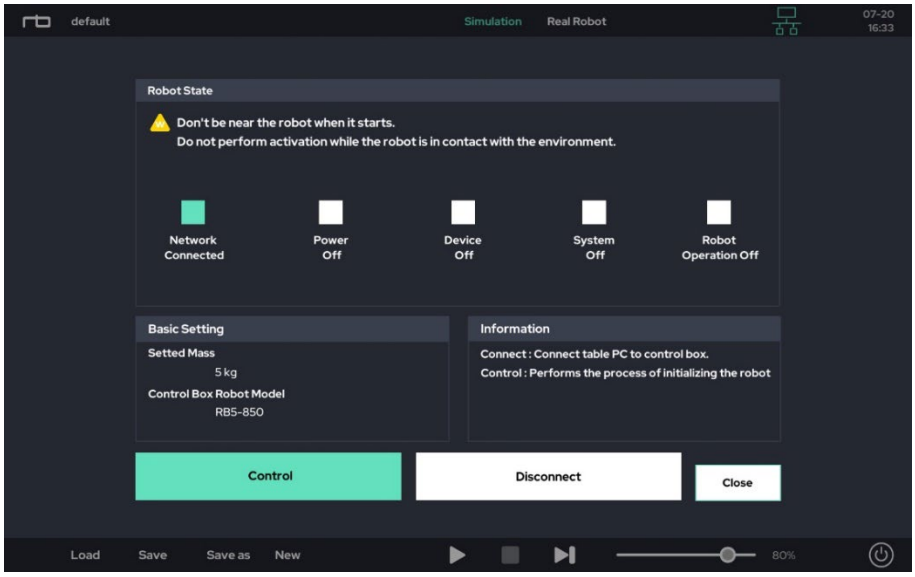
© RAINBOW ROBOTICS Inc. All rights reserved.

Der folgende Bildschirm erscheint, wenn die Steuerbox mit dem Computer verbunden sind.



Beim Verbindungsversuch leuchtet die erste Statusleuchte gelb.

Wenn die Verbindung hergestellt ist, wird der folgende Bildschirm angezeigt und die ‚Control'-Schaltfläche wird aktiviert.



© RAINBOW ROBOTICS Inc. All rights reserved.

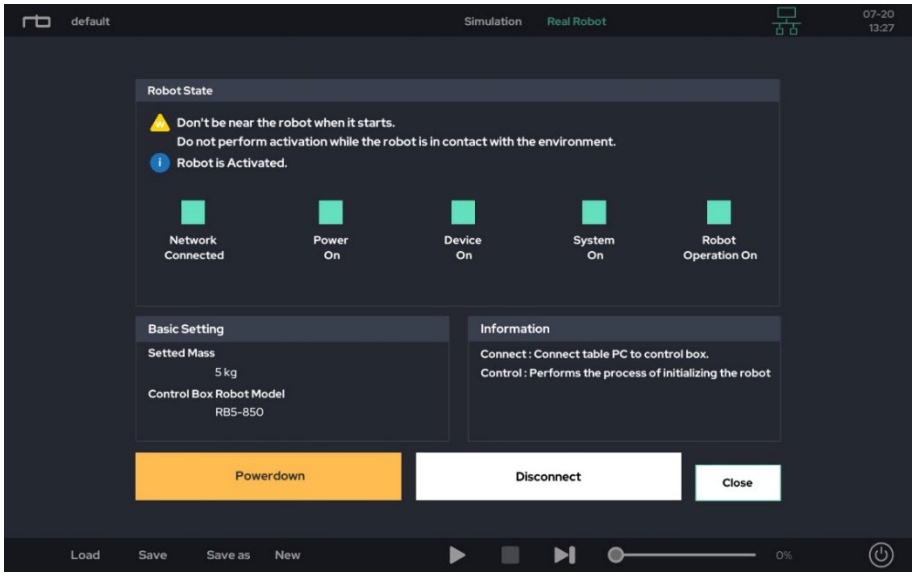
Nach der Verbindung mit der Steuerbox muss die Robotersteuerung eingeschaltet werden.

Der Roboter wird durch Drücken der aktivierten Steuertaste eingeschaltet.

- Control-Taste: Für den Betrieb wird der Roboterarm initialisiert.

Wenn die Control-Taste gedrückt wird, wird der Roboterarm in einen steuerbaren Zustand initialisiert. Dabei wird die Gelenkbremse gelöst und der Roboterarm gibt nacheinander Geräusche von sich.

Wenn alle Systeme eingeschaltet und verbunden sind, leuchten alle Statusanzeigen blau, wie im folgenden Bildschirm gezeigt.

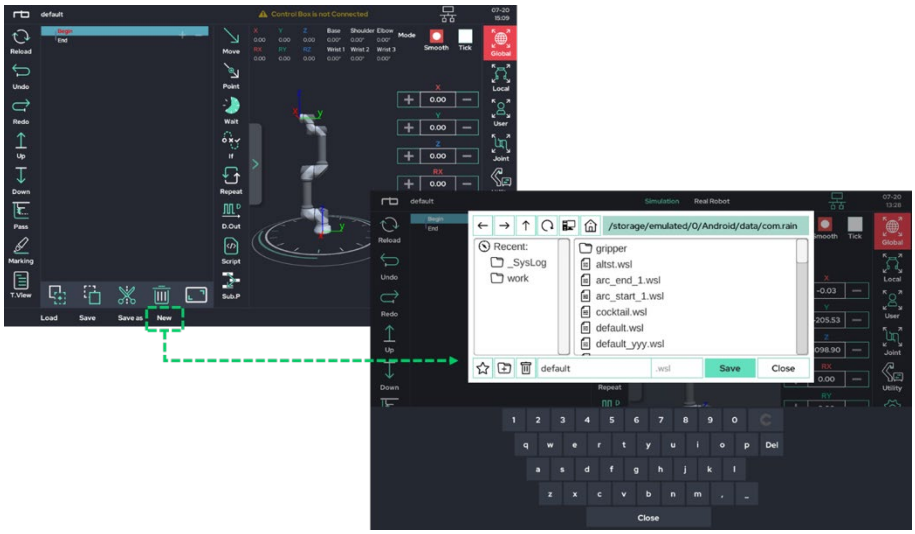


※ Wenn das blaue Licht bei 'Robot Operation On' nicht aufleuchtet und das rote Licht aufleuchtet, sollten Sie die Einstellungen und den Betriebsstatus gemäß den Popup-Meldungen überprüfen: ob der Not-Aus-Schalter eingeschaltet ist oder Einstellung der Werkzeuglast und Installationswinkel des Roboters usw.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Neues Projekt erstellen

  Um ein neues Projekt zu erstellen, klicken Sie auf die ‘New‘-Schaltfläche unten im ‚Make‘-Bildschirm. Daraufhin wird ein Bildschirm zum Zuweisen eines Dateinamens angezeigt.



  Der Standardname eines neuen Projekts ist ‘default’. Wenn Sie das Feld mit dem Standardnamen auswählen, um das Projekt zu umbenennen, wird eine virtuelle Tastatur angezeigt, und der Benutzer kann den Namen wie auf dem Bildschirm angezeigt ändern. Nachdem Sie den Namen geändert haben, müssen Sie auf die Schaltfläche Speichern klicken, um das Projekt zu erstellen.

© RAINBOW ROBOTICS Inc. All rights reserved.

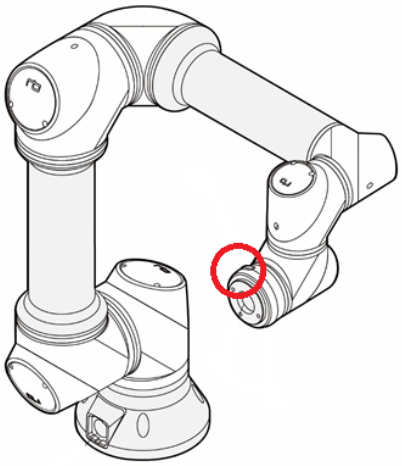
7.3 Teaching (Programmieren)

- So bewegen Sie den Roboter

  Es werden zwei Roboterantriebsmethoden bereitgestellt.

  - Direkt-Teaching: Direkt-Teaching per Knopfdruck am Roboterarm möglich.
  - Jogging: Mit dem Jog auf der Benutzeroberfläche wird der Roboter angetrieben.

- Direkt-Teaching (Direct-teaching / Schwerkraftkompensation)



Der distale Werkzeugflansch des Roboters verfügt über einen Silikonknopf für Direkt-Teaching. Für Direkt-Teaching muss ein Benutzer den mechanischen Knopf gedrückt halten. Durch Drücken dieser Taste kann sich jedes Gelenk frei bewegen.

Vorsicht

Vorsicht:

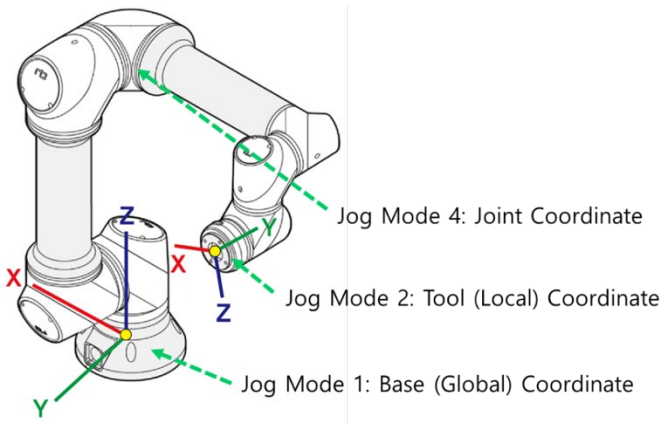
1) ‘Direkt-Teaching‘ kann nur verwendet werden, wenn der Roboter gestartet wird.
2) Wenn eine Last am Werkzeugflansch installiert ist, kann die ‘Direkt-Teaching‘ versagen, wenn die Lasteinstellung nicht im Voraus erfolgt ist. Die Ladeeinstellung erfolgt im Setup-Tool-Bildschirm.
3) Die Empfindlichkeit für jedes Gelenk kann in der Setup-Interface angepasst werden.
4) Verwenden Sie ‘Direkt-Teaching‘, nachdem der Roboter vollständig gestoppt ist.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Jogging

Es gibt vier Jogging-Modi.

Modus 1	Global	**Bewegung im kartesischen Koordinatensystem (basierend auf dem Basiskoordinatensystem).**
Modus 2	Local	Bewegung im kartesischen Koordinatensystem (basierend auf dem lokalenWerkzeugkoordinatensystem).
Modus 3	User	Bewegung im kartesischen Koordinatensystem (basierend auf dem benutzerdefinierten - Koordinatensystem).
Modus 4	Joint	Einzelne Gelenkbewegungen.



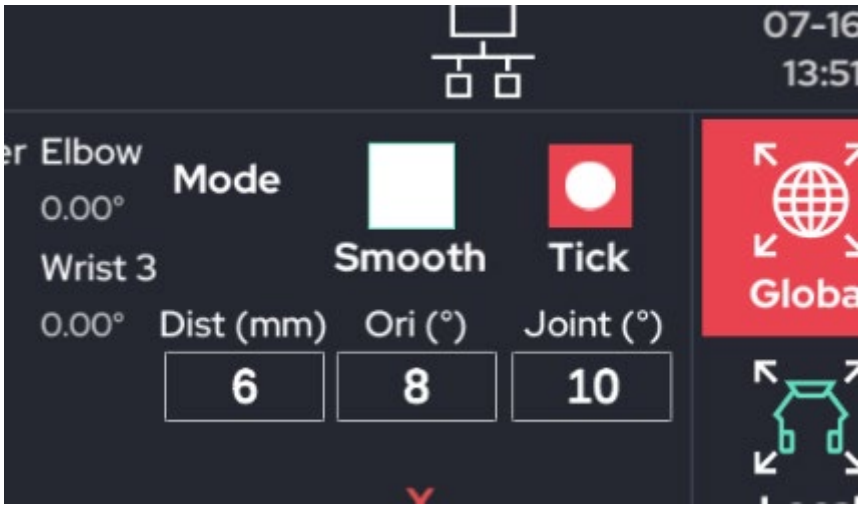
© RAINBOW ROBOTICS Inc. All rights reserved.

Es gibt zwei Möglichkeiten, das Joggen zu steuern.

- Smooth: Während die +-. –Taste gedrückt wird, bewegt sich der Roboterarm.
- Tick: Der Roboterarm bewegt sich Schritt für Schritt um einen bestimmten Abstand oder Winkel

※ Die beiden oben genannten Betriebsmethoden können mit der Umschalttaste oben rechts auf dem Make-Bildschirm ausgewählt werden.

※ Wenn 'Tick' ausgewählt ist und ‚Jog' verwendet wird, kann die Bewegungsdistanz oder der Bewegungswinkel im Setup-Interface eingestellt werden. Oder die Werte können im Echtzeit-Popup-Fenster wie unten gezeigt angegeben werden.



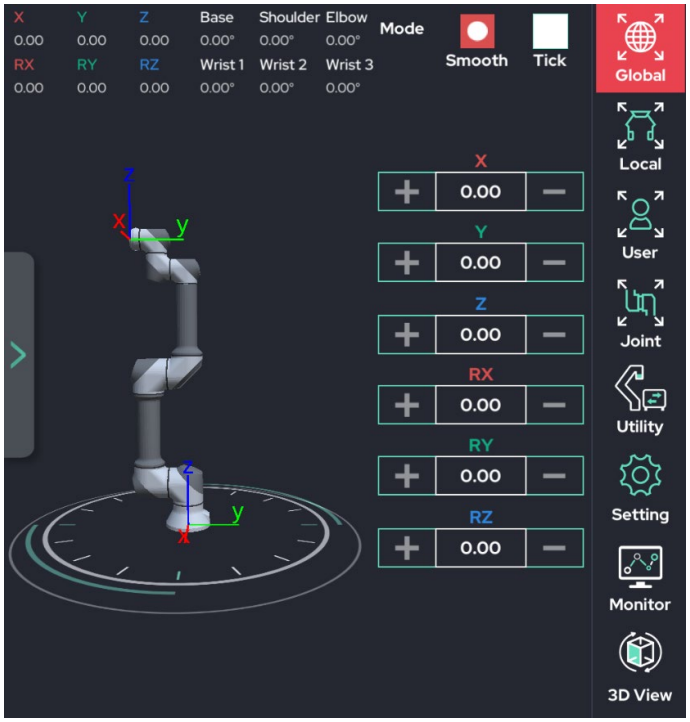
Warnung

Warnung:

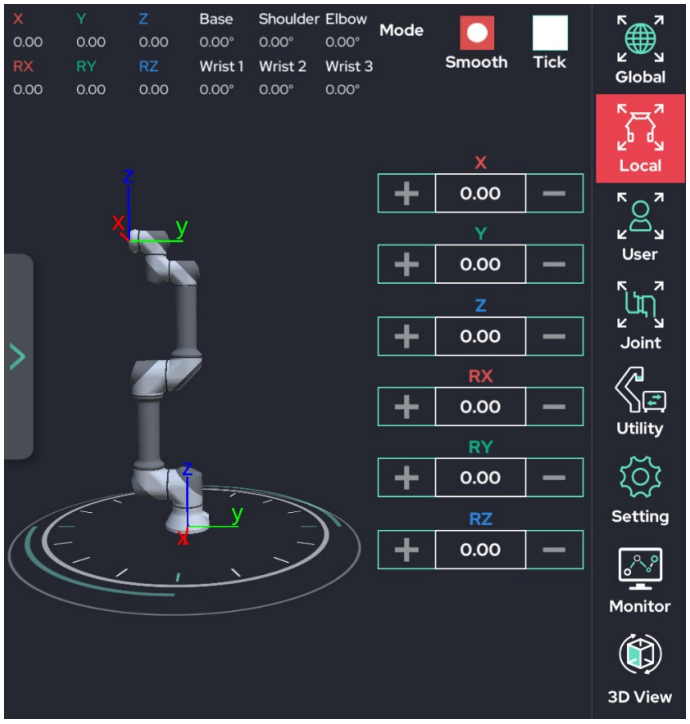
1) Wenn Sie den Roboter mit Jog-Modus bewegen, sollten Sie vor dem Betrieb sicherstellen, dass sich keine Hindernisse oder Personen in der Nähe befinden.
2) Es wird empfohlen, den Jog in einer sicheren Umgebung zu betreiben, indem der Sicherheitsschieber in der Setup-Interface verwendet wird (Der Sicherheitsschieber ist werkseitig aktiviert).

© RAINBOW ROBOTICS Inc. All rights reserved.

[Jog-Typ 1: TCP-Jog basierend auf dem Basiskoordinatensystem]

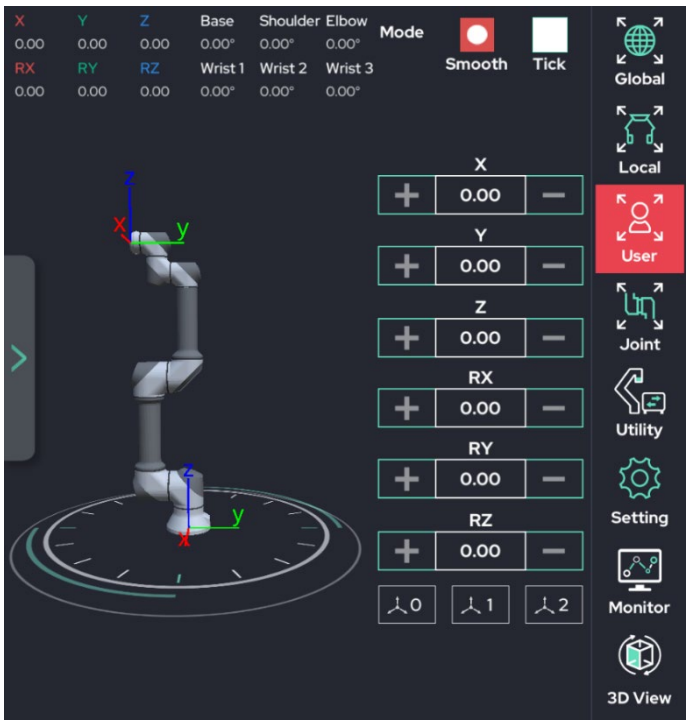


[Jog-Typ 2: TCP-Jog basierend auf dem lokalen Werkzeugkoordinatensystem]

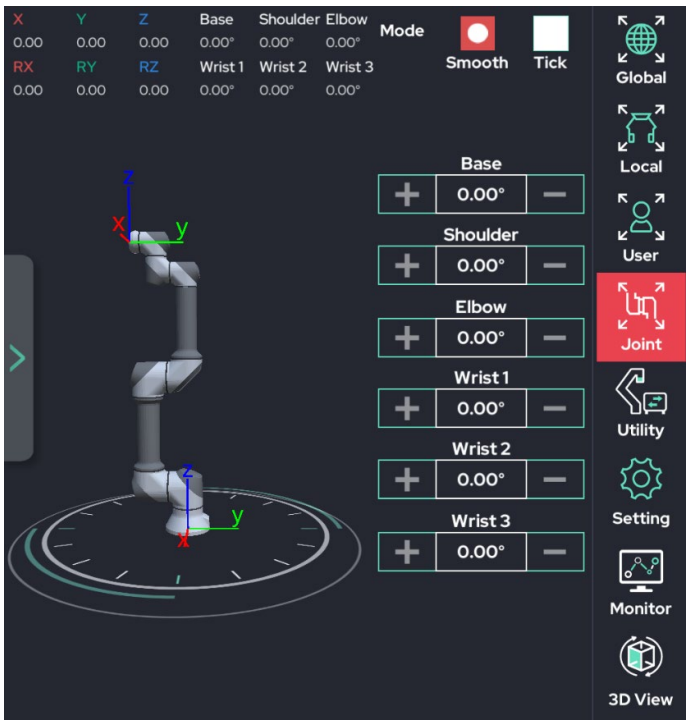


© RAINBOW ROBOTICS Inc. All rights reserved.

[Jog-Typ 3: TCP-Jog basierend auf dem benutzerdefinierten -Koordinatensystem]



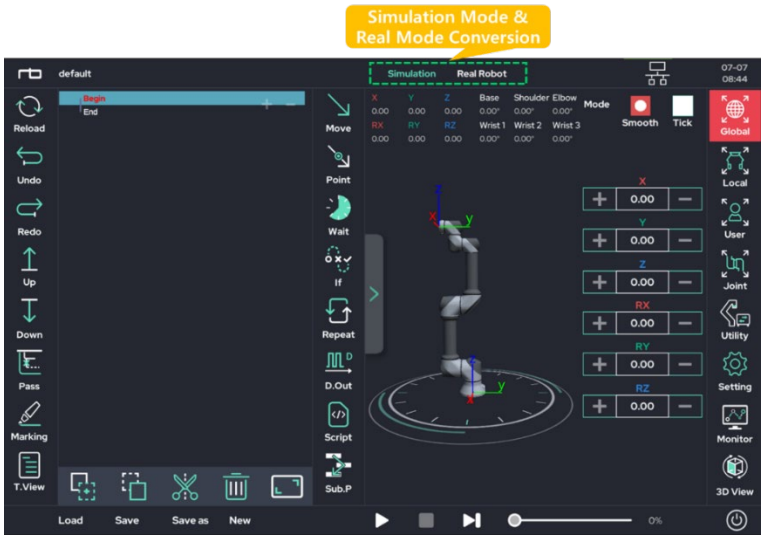
[Jog-Typ 4: Jog basierend auf einem einzelnen Gelenkkoordinatensystem]



© RAINBOW ROBOTICS Inc. All rights reserved.

- Real-Robot-Modus und Simulation-Modus

  Beim Testen oder Betrieben von Roboterbewegung stehen zwei Modi zur Verfügung.



  - Simulation-Modus:

    Durch diesen Modus wird der Roboterarm virtuell auf dem UI-Bildschirm zu bewegt, ohne dass der echten Roboter sich bewegt.

    Bei Teaching einer neuen Bewegung wird aus Sicherheitsgründen empfohlen, zuerst den Simulationsmodus auszuführen.
  - Real-Roboter(Real Robot)-Modus:

    Durch diesen Modus bewegt sich der Roboter virtuell auf dem UI-Bildschirm sowie der echte Roboter.

Vorsicht

Vorsicht:

1) Der Roboterbetrieb im Realmodus ist nur verfügbar, wenn der Roboter aktiviert ist.
2) Der Simulationsmodus ist ohne Aktivierung des Roboters verfügbar, solange die Steuerbox und die Tablet-UI verbunden sind.
3) Wenn der Roboter im Realmodus betrieben wird, bewegt sich der echte Roboter. Verwenden Sie es daher, nachdem Sie die Sicherheit der Umgebung bestätigt haben.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Roboterbewegung-Teaching

- **Move**-Funktion: Bewegungseigenschaften des Roboterarms wird definiert.
- **Point**-Funktion: Als Unterpunkt von **Move** ist es ein Teil, der die Haltung oder Position festlegt, die durch Bewegung erreicht werden sollte.

Wenn Sie das Programm mit diesen beiden Funktionen konfigurieren, sieht die Gesamtkonfiguration wie folgt aus.

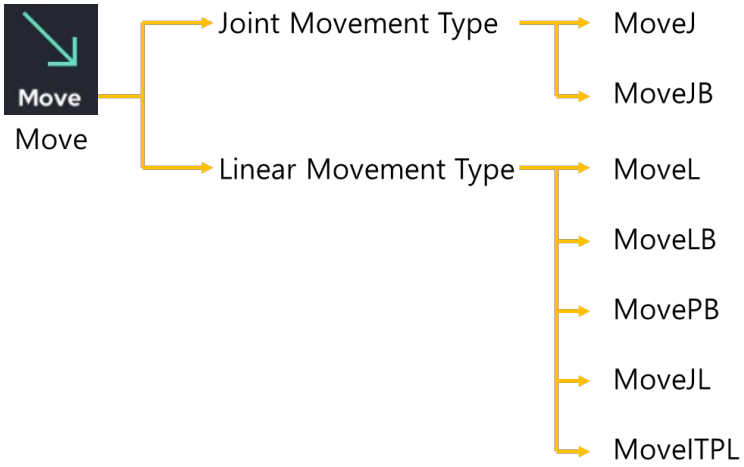
← Bewegungseigenschaft

← Erster Zielpunkt
← Zweiter Zielpunkt

Detaillierte Beschreibung der einzelnen **Move-** und **Point-** Funktionen finden Sie auf den folgenden Seiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Move-Funktion



Move-Funktion dient zum Einstellen der Bewegungseigenschaften und Bewegungsoptionen des Roboterarms. Move-Funktion besteht aus den Unterserien **Joint** und **Linear**.

① **Gelenk-Typ (Joint Movement Type)**

Die Gelenkserie ist eine Bewegungseigenschaft, die den Winkel jedes Gelenks unabhängig von der Bewegung des Roboterendes (TCP) auf einen Zielwert bewegt. Als untergeordnete Elemente sind MoveJ und MoveJB vorhanden.

▷ **MoveJ (Move Joint):**

Der Winkel jedes Gelenks wird auf den Wert des Zielpunkts geändert. Die Fahrgeschwindigkeit anderer Gelenke wird entsprechend dem Gelenk angepasst, das die meiste Fahrzeit benötigt.

▷ **MoveJB (Move Joint Blend):**

Der Roboter beginnt am Startpunkt und bewegt sich reibungslos zwischen jeweiligen Punkten auf die Move J-Weise, ohne anzuhalten.

© RAINBOW ROBOTICS Inc. All rights reserved.

② Linear-Typ (Linear Movement Type)

Die Linear-Serie ist eine Bewegungseigenschaft, die das Ende des Roboterarms (TCP) basierend auf dem kartesischen Koordinatensystem bewegt. Daher werden die Werte des Kartesischen Koordinatensystems als Zielwerte für die Bewegung verwendet. Als untergeordnete Elemente sind MoveL, MoveLB, MovePB, MoveJL, und MoveITPL vorhanden.

▷ MoveL (Move Linear):

TCP bewegt sich in einer geraden Linie vom Standpunkt (aktueller Position) zum Zielpunkt (Einstellposition). Die Ausrichtung des Werkzeugs wird gedreht, um den kleinsten Bewegungswinkel zu haben. Die Betriebsgeschwindigkeit für andere Aufgaben wird angepasst, um den zeitaufwändigen Aufgaben zwischen TCP-Bewegung und Orientierungsänderungen zu entsprechen.

▷ MoveLB (Move Linear Blend):

Der Roboter beginnt am Startpunkt (aktueller Position) und bewegt sich reibungslos zwischen jeweiligen Punkten auf die Move L-Weise, ohne anzuhalten.

Durch Angabe des Blend-Radius für jeden Punkt werden der vordere und der hintere Pfad um den entsprechenden Abstand geschnitten und entlang des Bogenpfads verschoben. Wenn der angegebene Radius größer als die Hälfte der Länge des vorderen und hinteren Pfads ist, wird er automatisch auf die Hälfte der Länge des kürzeren der beiden Pfade geändert.

Move LB besteht aus zwei Modi. Im Constant-Modus wird die Ausrichtung des Werkzeugs während der Bewegung am Startpunkt (aktueller Position) beibehalten. Im Intended-Modus wird die Ausrichtung des Werkzeugs nach gespeicherten Angaben jeweiligen Punkts geändert.

▷ MovePB (Move Point Blend):

Ähnlich wie MoveLB, aber mit einem allgemeinen Zweck verwendbar. Für jeden Wegpunkt kann der Grad der Überblendung (blend) in Distanzeinheiten oder in Prozent eingestellt werden. Außerdem kann die Geschwindigkeit für jeden Zwischenpunkt separate eingestellt werden.

▷ MoveJL (Move J with Linear Input):

Wie MoveL nimmt es den kartesischen Wert des Zielpunkts als Eingabe. Anstatt jedoch direkt auf den Punkt zu geben, wird die MoveJ-Weise verwendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Wenn die Eingabe des kartesischen Koordinatensystems empfangen wird, wird sie durch inverse Kinematik in den Zielgelenkwinkel umgewandelt und dann erneut in MoveJ eingegeben.

▷ **MoveITPL (Move Interpolation):**

Der Roboter beginnt am Startpunkt (aktueller Position) und bewegt sich reibungslos zwischen jeweiligen Punkten auf die Move L-Weise, ohne anzuhalten.

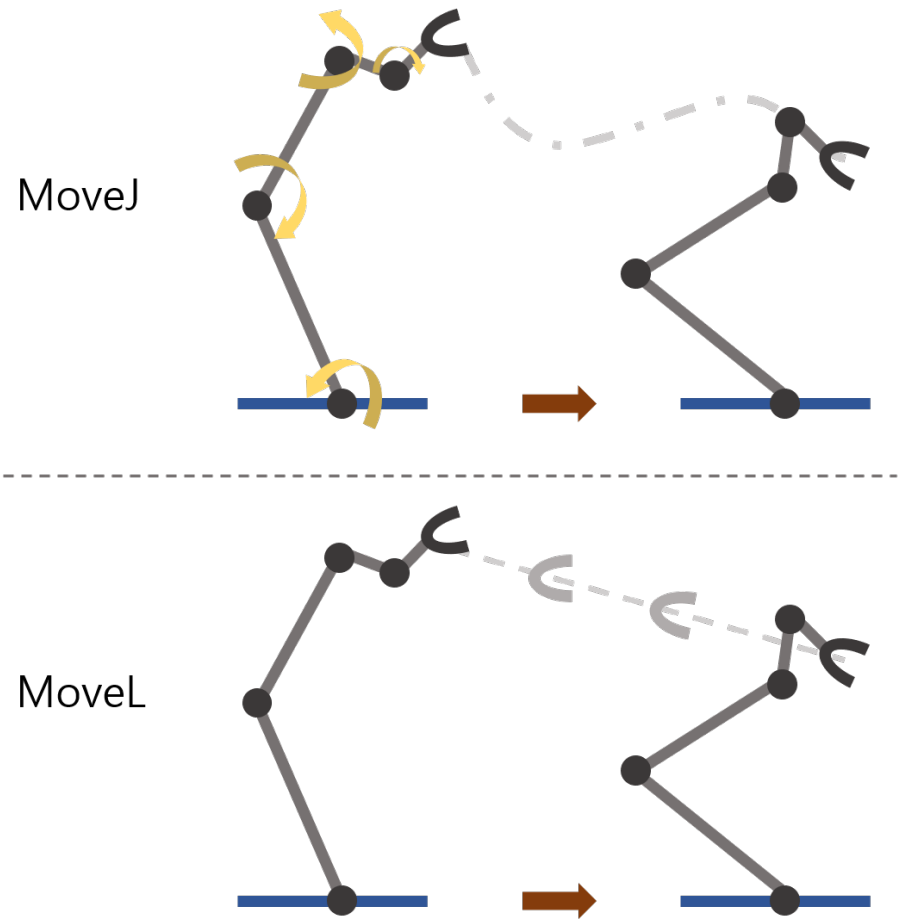
MoveLB und PB bewegen sich gemischt (blended), ohne jeden Wegpunkt zu passieren, aber MoveITPL bewegt sich mit einer Trajektorie, die jeden Wegpunkt genau passiert. Daher gibt es keine separate Blend-Einstellung.

MoveITPL besteht aus zwei Modi. Im Constant-Modus wird die Ausrichtung des Werkzeugs während der Bewegung am Startpunkt (aktueller Position) beibehalten. Im Intended-Modus wird die Ausrichtung des Werkzeugs nach gespeicherten Angaben jeweiligen Punkts geändert.

Die Geschwindigkeit kann für jeden Zwischenwegpunkt separate eingestellt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Unterschied zwischen MoveJ und MoveL

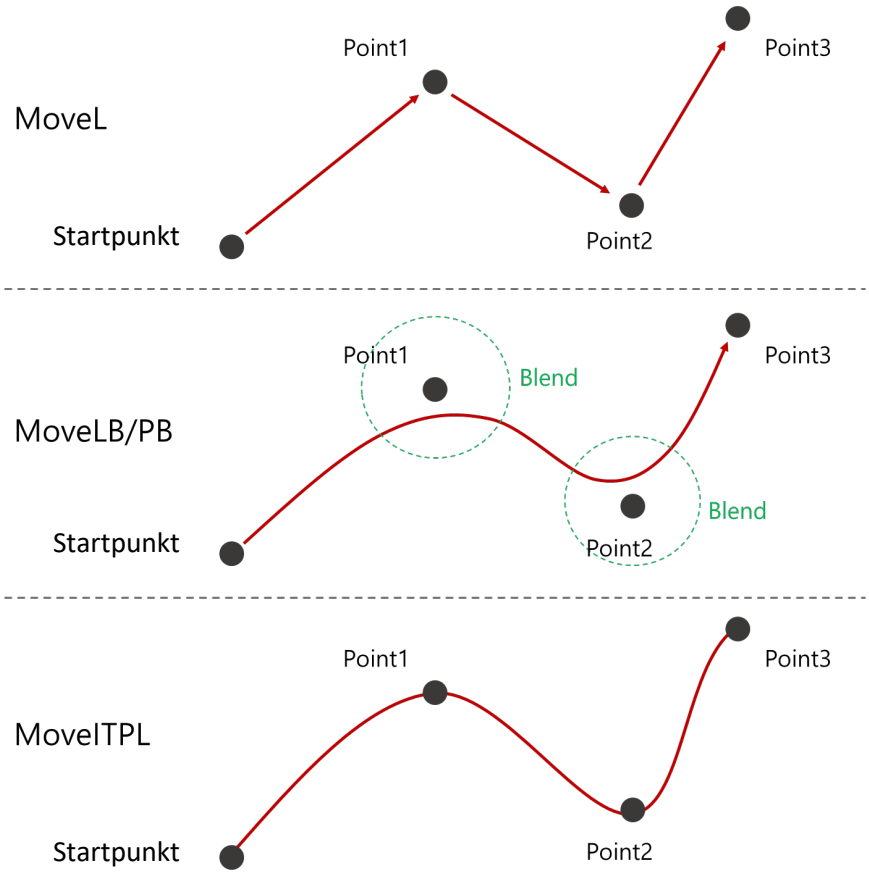


MoveJ ist ein Betriebsmodus, der nur die Gelenkwinkelinformationen des Startpunkts und des Zielpunkts verwendet, ohne die Trajektorie des distalen Endes (TCP) zu berücksichtigen. Die Fahrgeschwindigkeit anderer Gelenke wird entsprechend dem Gelenk angepasst, das die meiste Fahrzeit benötigt. Daher wird der Gelenkwinkel von 6 Gelenken als Zielpunktwert eingegeben.

MoveL ist ein Betriebsmodus, der inverse Kinematik verwendet, um die Trajektorie des Endes (TCP) linear vom Startpunkt zum Zielpunkt zu bewegen. Daher werden 6 kartesische Koordinatensystemwerte (x, y, z, Rx, Ry, Rz) als Zielpunktwert eingegeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Unterschied zwischen MoveL, MoveLB/PB und MoveITPL



Im Fall von MoveL bewegt der Roboter sich auf einer geraden Linie zwischen dem Startpunkt und dem Zielpunkt auf einem linearen Pfad. Das Ziel wird genau erreicht, aber die Bewegungsgeschwindigkeit wird am Ziel vorübergehend auf null gesetzt.

Im Fall von MoveLB/PB werden die restlichen Punkte außer dem Zielpunkt als Wegpunkte verwendet, und je nach angegebener Blendrate oder Blenddistanz wird der Wegpunkt nicht genau passiert, ohne anzuhalten.

Im Fall von MoveITPL bewegen sich alle Punkte außer dem Zielpunkt zum Wegpunkt, und es wird eine Trajektorie erstellt, die durch den Wegpunkt verläuft. Das durchläuft die erstellte Trajektorie, ohne anzuhalten, und es ist möglich, für jeden Wegpunkt eine separate Geschwindigkeit einzustellen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht:

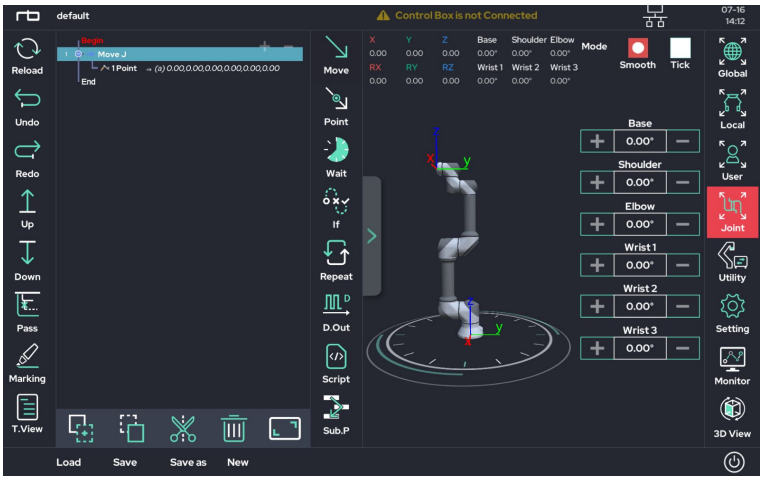
1) 5 Betriebsmodi wie MoveL, MoveLB, MovePB, MoveJL, MoveITPL bewegen den Roboter durch inverse Kinematik. Daher kann die Bewegung in der singulären Position eingeschränkt sein, in der eine inverse kinematische Berechnung unmöglich ist.

2) Außerdem können sich in oder in der Nähe der toten Zone(Dead-Zone) des Roboters bestimmte Gelenke schneller als erwartet bewegen oder ihre Bewegung kann eingeschränkt sein. Informationen zum Bereich der toten Zone sind in Abschnitt 1.7 beschrieben.

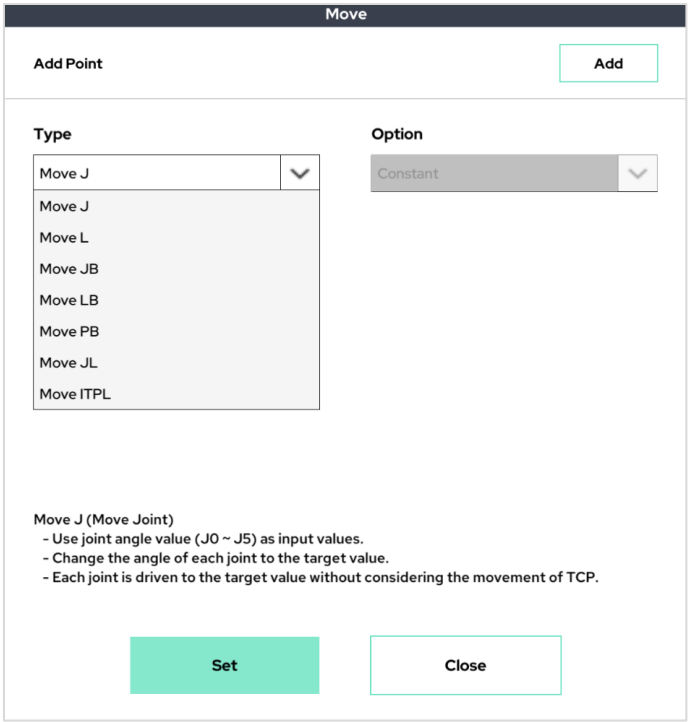
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Bewegungseigenschaften von Move ändern

Wenn die Move-Funktion zum ersten Mal ins Programm eingefügt wird, wird die Move-Funktion im Programmbaum erstellt, wie in der folgenden Abbildung gezeigt. Beim ersten Erstellen wird es standardmäßig mit der MoveJ-Option erstellt.



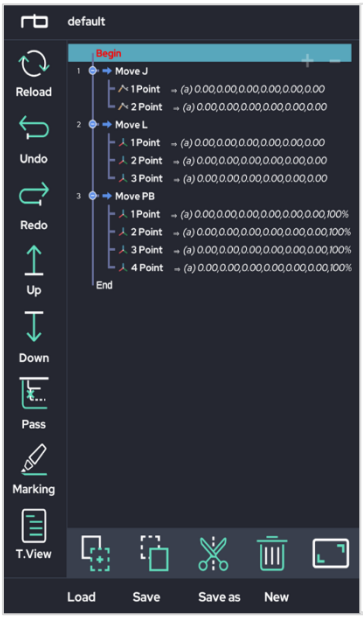
Um eine andere oben beschriebene Option zu ‘Option’ auszuwählen, klicken Sie auf das erstellte MoveJ-Teil und das folgende Popup-Fenster wird angezeigt.



© RAINBOW ROBOTICS Inc. All rights reserved.

Wenn Sie die gewünschte Bewegungseigenschaft/ Option auswählen und einstellen, wird die Bewegungseigenschaft von Move geändert.

Das Programmbeispiel bei Teaching des Betriebs mit den obigen Funktionen ist wie folgt.



Programm starten
Bewegung (Typ: Move Joint)
- Zielpunkt 1: Gelenkwinkelwert als Zielwert angeben
- Zielpunkt 2: Gelenkwinkelwert als Zielwert angeben

Bewegung (Typ: Move Linear)
- Zielpunkt 1: Kartesische Koordinatenwerte als Zielwert angeben
- Zielpunkt 2: Kartesische Koordinatenwerte als Zielwert angeben
- Zielpunkt 3: Kartesische Koordinatenwerte als Zielwert angeben

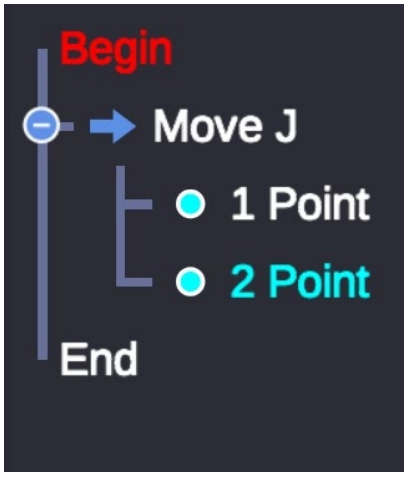
Bewegung (Typ: Move Point Blend)
- Wegpunkt 1: Kartesische Koordinatenwerte als Zielwert angeben
- Wegpunkt 2: Kartesische Koordinatenwerte als Zielwert angeben
- Wegpunkt 3: Kartesische Koordinatenwerte als Zielwert angeben
- Zielpunkt: Kartesische Koordinatenwerte als Zielwert angeben

Programm endet

MoveJ, MoveJB
Da es sich um eine Bewegungseigenschaft handelt, die sich basierend auf einem Gelenk bewegt, wird der zu bewegenden Zielhaltung im unteren Element ‘Point’ eingestellt. ※ Da der Roboterarm aus 6 Gelenken besteht, werden Informationen über die 6 Gelenkwerte in den unteren Punkten von MoveJ und JB eingestellt.
MoveL, MoveLB, MovePB, MoveJL, MoveITPL
Da es sich basierend auf oder zu dem Wert des kartesischen Koordinatensystems bewegt, wird der Wert des kartesischen Koordinatensystems des zu verschiebenden oder zu passierenden Zielpunkts in Point festgelegt, was ein untergeordnetes Element ist. ※ Da das kartesische Koordinatensystem aus 6 Werten besteht (x, y, z, Rx, Ry, Rz), werden Informationen über 6 kartesische Koordinatensystemwerte an den unteren Punkten von **MoveL, LB, PB, JL** und **ITPL** gesetzt.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Point-Funktion



← Bewegungseigenschaft

← Erster Zielpunkt

← Zweiter Zielpunkt

Wie bereits bei der Move-Funktion erläutert, ist die Point-Funktion eine Unterfunktion der Move-Funktion. Die Move-Funktion gibt die Eigenschaften der Bewegung an, und die Point-Funktion ist für die Einstellung der Zielpunktinformationen verantwortlich, um den Roboterarm gemäß jeder Eigenschaft zu bewegen.

In der Point-Funktion variiert der einzustellende Zielwert entsprechend den in der Move-Funktion eingestellten Bewegungseigenschaften.

▷ Unterpunkt von Joint Movement Type (MoveJ, MoveJB):

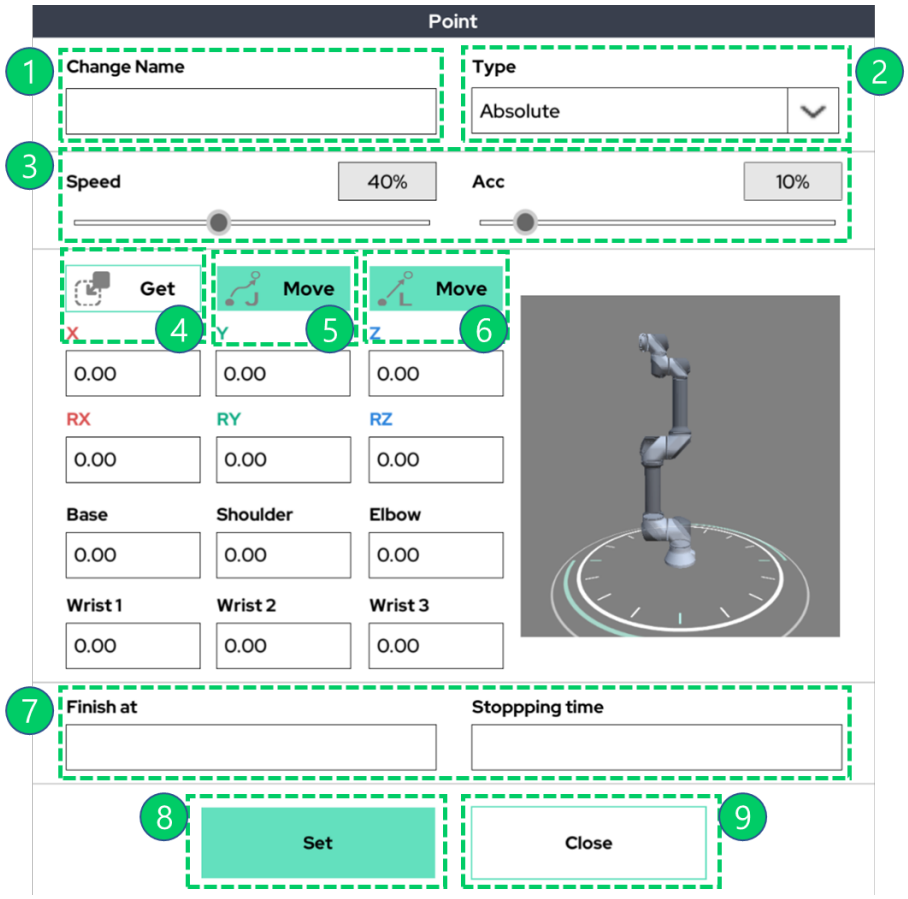
6 Gelenkwinkelwerte der zu bewegenden Zielhaltung werden eingestellt.

▷ Unterpunkte von Linear Movement Type (MoveL, MoveLB, MovePB, MoveJL, MoveITPL):

Der kartesische Koordinatensystemwert des zu verschiebenden oder zu passierenden Zielpunktes wird eingestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Die grundlegende Konfiguration des Popup-Fensters zum Einstellen der Point-Funktion ist wie folgt.



Weitere Beschreibung der einzelnen Bereiche finden Sie in der folgenden Tabelle.

© RAINBOW ROBOTICS Inc. All rights reserved.

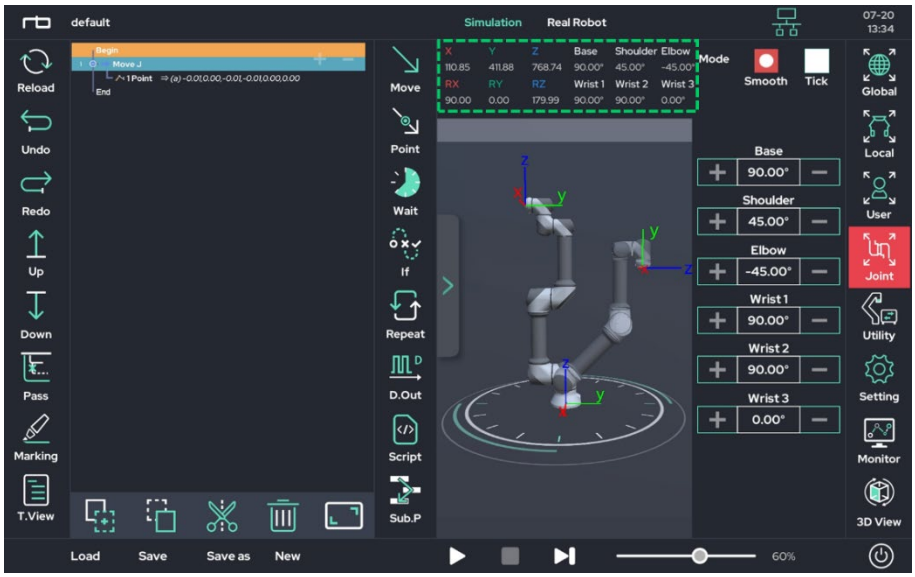
	Beschreibung
①	Sie können dem Punkt einen Namen geben. Das Festlegen eines Namens ist nicht erforderlich. Ist der Name gesetzt, kann später die Ortsangebe des entsprechenden Punktes als Variable verwendet werden.
②	Eine Option zur Auswahl des Einstellungstyp von Point. Der Unterpunkt der Joint Move-Eigenschaft hat drei Einstellungsmöglichkeiten, für den Unterpunkt der Linear Move-Eigenschaft gibt es 4 Einstellungsmöglichkeiten. Wenn Sie zum ersten Mal einen Point erstellen, ist der Standardtyp 'Absolute'.
③	Stellen Sie die Geschwindigkeit und Beschleunigung ein, um zum Zielpunkt zu gelangen.
④	Erfassen Sie Informationen über die aktuelle Haltung /Winkel des Roboters und bringen Sie diese als Information mit. Nachdem Sie den Roboter durch Joggen oder direktes Teaching an den gewünschten Punkt/die gewünschte Position bewegt haben, werden die Informationen dieses Punktes in den Punktinformationen widergespiegelt, wenn Sie auf ‚Get (Importieren)' drücken. Um die neu refaktierten Informationen zu speichern, müssen Sie die Set-Taste (Nr. 7) drücken. Es gibt Typen, die abhängig vom Point-Typ(Nr. 2) diese Schaltfläche unterstützen, und es gibt Typen, die nicht unterstützt werden.
⑤	Eine Funktion, sich in eine bestimmte Haltung zu bewegen. Solange die Taste gedrückt gehalten wird, fährt der Roboter in die gespeicherte Position. Bewegung wird als Joint Movement-Typ eingestellt. Wenn der Bewegung abgeschlossen wird, erscheint eine Popup-Meldung, die Sie darüber informiert. Abhängig vom Punkttyp (Nr. 2) gibt es Typen, die diese Schaltfläche unterstützen und nicht unterstützen.
⑥	Eine Funktion, sich in eine bestimmte Haltung zu bewegen. Solange die Taste gedrückt gehalten wird, fährt der Roboter in die gespeicherte Position. Bewegung wird als Linear Movement-Typ eingestellt. Wenn der Bewegung abgeschlossen wird, erscheint eine Popup-Meldung, die Sie darüber informiert. Abhängig vom Punkttyp (Nr. 2) gibt es Typen, die diese Schaltfläche unterstützen und nicht unterstützen.

© RAINBOW ROBOTICS Inc. All rights reserved.

⑦	Geben Sie die Bedingung zum Verlassen der Bewegung (in Finish at eingeben) und die Zeit zum Verlassen der Bewegung (in Stopping time eingeben) an. Sie sind kein Pflichtfeld. Wenn Sie dieses Feld nicht ausfüllt, wird die Bewegung normalerweise nach Erreichen des Zielpunkts/der Zielposition beendet. Wenn Sie für diesen Punkt eine bestimmte Bedingung und Zeit einstellen, wird der Vorgang bereits vor dem Erreichen des Zielpunkts/der Zielposition. Wenn die Bedingung erfüllt ist, wird der Bewegung angehalten und die nächste Bewegung fortgesetzt, egal ob die Bewegung abgeschlossen ist. Die minimale Zeit beträgt 0 Sekunde.
⑧	Die geänderten Einstellungen werden gespeichert.
⑨	Einstellungsfenster wird geschlossen. Wenn es eine geänderte Einstellung gibt, müssen Sie die Set-Taste (Nr. 7) drücken.

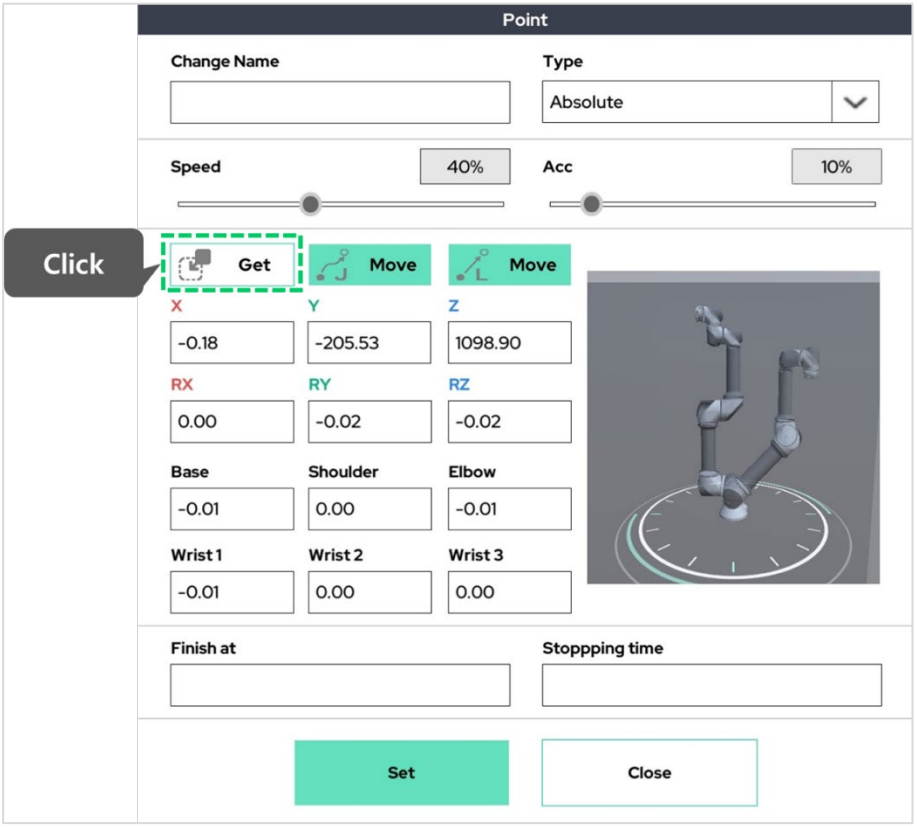
※ Das Schema der in Get (Importieren), Element 4 in der obigen Tabelle, bereitgestellten Funktionen ist wie folgt.

1. Mit der Jog-/ Direkt-Teaching-Funktion in die gewünschte Haltung/ Position fahren



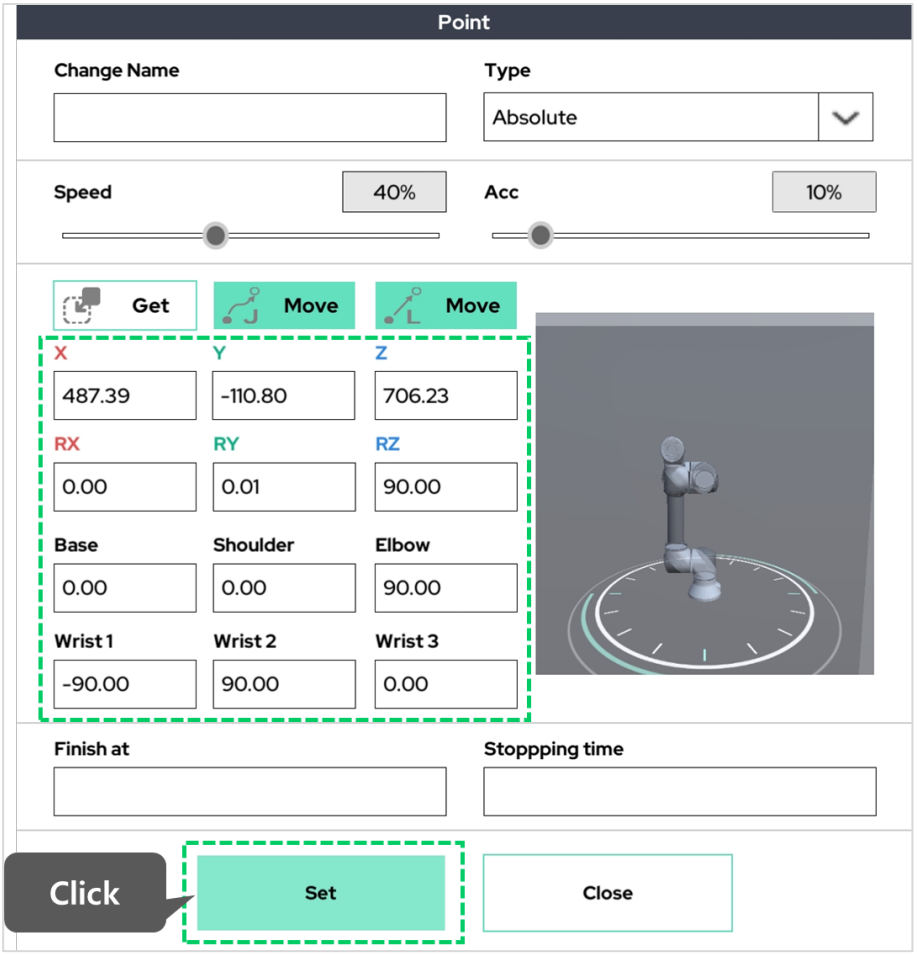
© RAINBOW ROBOTICS Inc. All rights reserved.

2. Informationen zur aktuellen Körperhaltung/ Position abrufen, indem Sie auf die Schaltfläche ‘Get’ klicken



© RAINBOW ROBOTICS Inc. All rights reserved.

3. Speichern nach der Bestätigung



※ Die unter ‘Finish at/Stopping time’ in Tabelle 6 oben bereitgestellten Funktionen lauten wie folgt.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Wenn die ‚Finish at'-Funktion nicht verwendet wird

  (Falls leer gelassen)

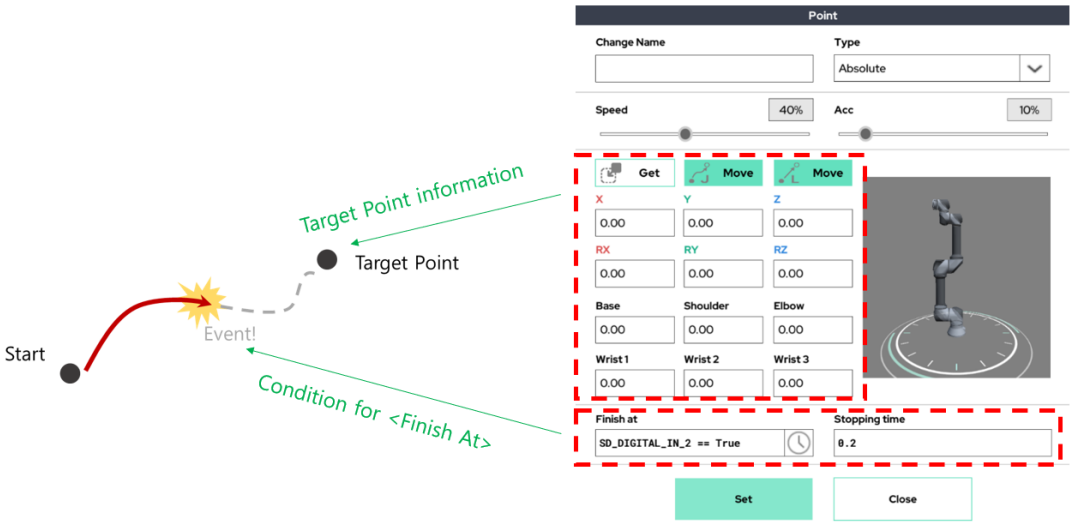
  Nach Erreichen des ursprünglich eingestellten Zielpunktes endet der Betrieb und der nächste Befehl wird ausgeführt

- Wenn die 'Finish at'-Funktion verwendet wird

  (Bei der Eingabe von bestimmten Bedingungen)

  Auch wenn der Zielpunkt nicht erreicht wird, wird der Betrieb beendet, wenn die , Finish at'-Funktion eintritt.

  Tritt die Funktion während des Betriebs nicht ein, wird der nächste Befehl nach Erreichen des Zielpunktes normal ausgeführt



© RAINBOW ROBOTICS Inc. All rights reserved.

Je nach Move-Typ sind folgende Einstellungen möglich.

Unterpunkt von Joint Move-Typ		
Optionen	Absolute	>Als gewünschter Gelenkwinkel wird ein fester Wert verwendet. >Sie können den Roboter durch Joggen/Direkt-Teaching in die gewünschte Haltung bringen und den Gelenkwinkel über die Get(Importieren)-Funktion einstellen.
	Variable	>Der Benutzer kann den Zielgelenkwinkel frei eingeben. >Sie können den Roboter durch Joggen/Direkt-Teaching in die gewünschte Haltung bringen und den Gelenkwinkel über die Get(Importieren)-Funktion eingeben. >Eine Eingabe über Variablen und mathematische Operationen ist möglich.
	Relative	>Stellen Sie den Betrag der Winkeländerung ein, den Sie von der vorherigen Winkelhaltung ändern möchten. >Wenn Sie 0 für alle Winkeländerung aller Gelenke eingeben, ist dies ein Befehl, sich in der gleichen Winkelposition wie die vorherige Position zu bewegen. >Eine Eingabe über Variablen und mathematische Operationen ist möglich.

>> nächste Seite

© RAINBOW ROBOTICS Inc. All rights reserved.

Unterpunkt von Linear Move-Typ		
Optionen	Absolute	>Als gewünschte kartesische Koordinatenwerte werden feste Werte verwendet. >Nachdem Sie den TCP des Roboters durch Joggen/Direkt-Teaching bewegt haben, können Sie den kartesischen Koordinatenwert über die Get(Importieren)-Funktion einstellen. >Das Referenz des eingestellten kartesischen Koordinatenwerts ist das Basiskoordinatensystem des Roboterarms (das vom Hersteller eingestellte Standardkoordinatensystem).
	Variable	>Benutzer kann die kartesische Zielkoordinatenwerte frei eingeben. > Das Referenz des eingestellten kartesischen Koordinatenwerts ist das Basiskoordinatensystem des Roboterarms (das vom Hersteller eingestellte Standardkoordinatensystem). > Nachdem Sie den TCP des Roboters durch Joggen/Direkt-Teaching bewegt haben, können Sie den kartesischen Koordinatenwert über die Get(Importieren)-Funktion einstellen. > Eine Eingabe über Variablen und mathematische Operationen ist möglich.
	Relative	>Sie können den Zielpunkt angeben, indem Sie den relativen Abstand/Winkel (offset) vom vorherigen Punkt oder einem bestimmten Punkt festlegen. >Mit dieser Funktion ist ein Punkt, zu dem relative verschoben werden soll, bei Referenzpunkt (Reference Point) auszuwählen. Der Standardwert ist PT_LAST_TCP und gibt den letzten Ankunftspunkt. Außerdem kann es auf einem beliebigen Punkt basieren. > In Referenzrahmen (Reference Frame) können sie entscheiden, welches Koordinatensystem die Relativbewegung bezeichnen soll. Der Standardwert ist Frame_Base. Der das Basiskoordinatensystem des Roboters angibt. Das kann auf das Benutzerkoordinatensystem oder das Lokalkoordinatensystem des Werkzeugs geändert werden. > Eine Eingabe über Variablen und mathematische Operationen ist möglich.

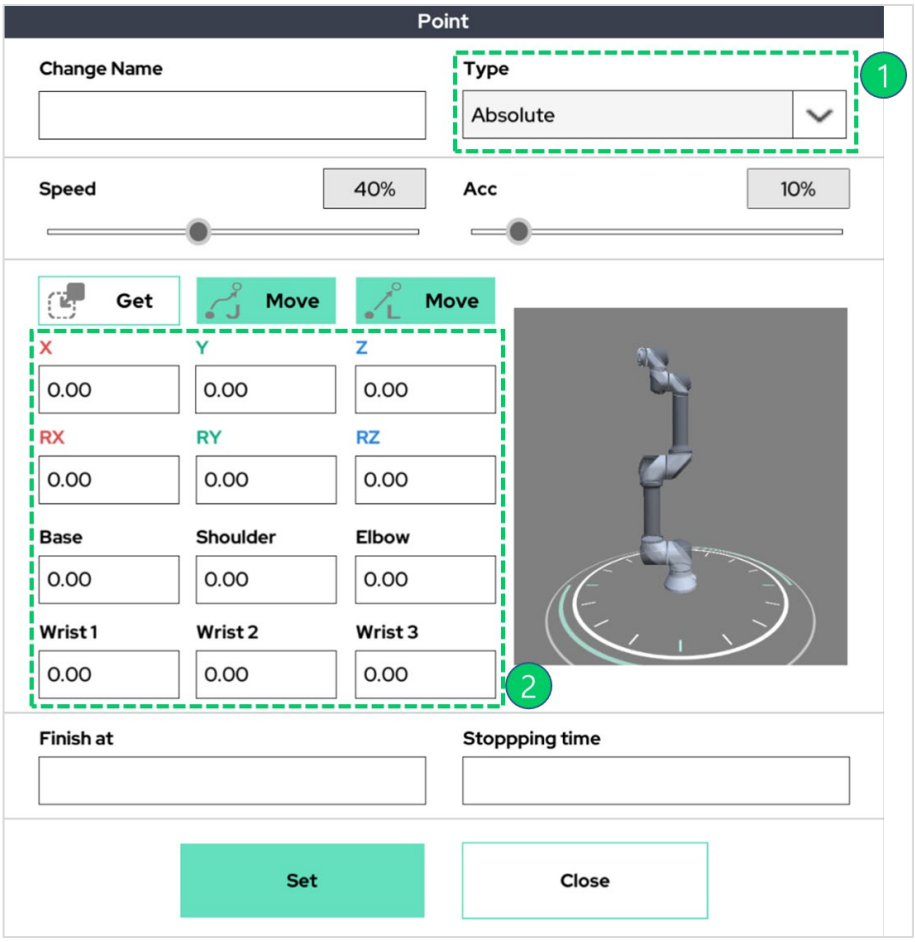
© RAINBOW ROBOTICS Inc. All rights reserved.

	User Coordinate	>Diese Funktion ähnelt Variable, aber sie ist eine Funktion zum Festlegen des Zielpunkts basierend auf dem vom Benutzer festgelegten Koordinatensystem. >Durch Einstellen des Referenzrahmen (Reference Frame) können Sie das Benutzerkoordinatensystem auswählen, das Sie als Referenz verwenden möchten. >Wenn Sie das gewünschte Referenzkoordinatensystem auswählen und durch die Get (Importieren)-Funktion verwenden, können die Haltungs-/Positionsinformationen des aktuellen Roboters basierend auf dem ausgewählten Koordinatensystem automatisch eingegeben werden. >Wenn beispielsweise im Benutzerkoordinatensystem 0 ausgewählt und in allen kartesischen Koordinatenwerten 0 eingegeben wird, bewegt sich der TCP zum Ursprung des Benutzerkoordinatensystems. > Eine Eingabe über Variablen und mathematische Operationen ist möglich.

© RAINBOW ROBOTICS Inc. All rights reserved.

Die UI-Konfiguration für jeden Point-Typ ist wie folgt.

▷ Gelenktyp – Absolute-Option

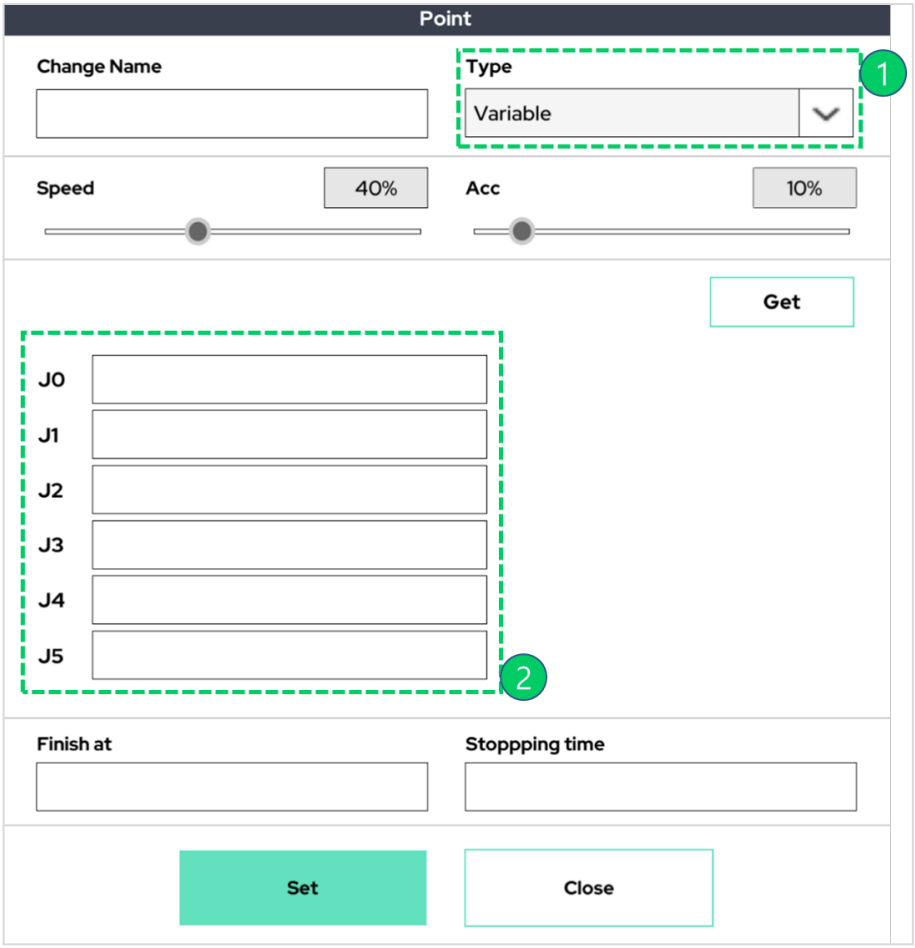


① Absolute-Option.

② Der Haltungs-/ Winkelwert des Roboters wird über die Schaltfläche ‘Get (Importieren)’ gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

▷ Gelenktyp - Variable

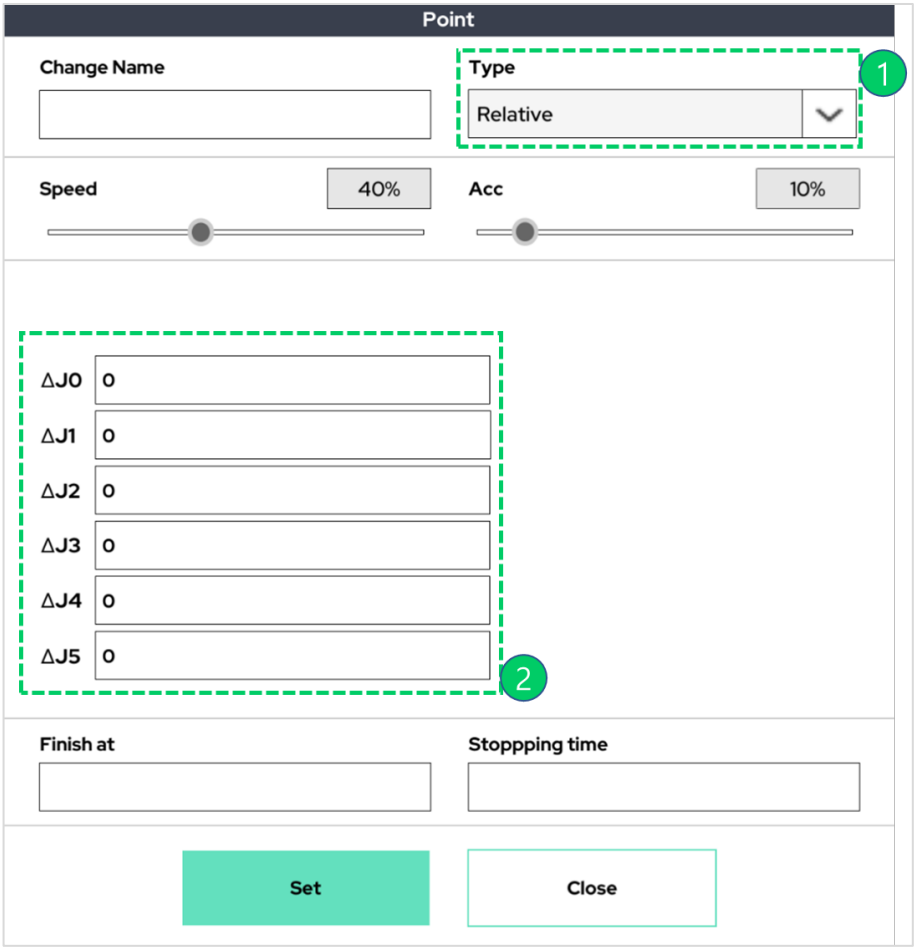


① Variable-Option.

② Der Gelenkwinkel für die Zielhaltung kann direkt eingegeben oder die parametrisierten Informationen als Formel eingegeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

▷ Gelenktyp – Relative-Option

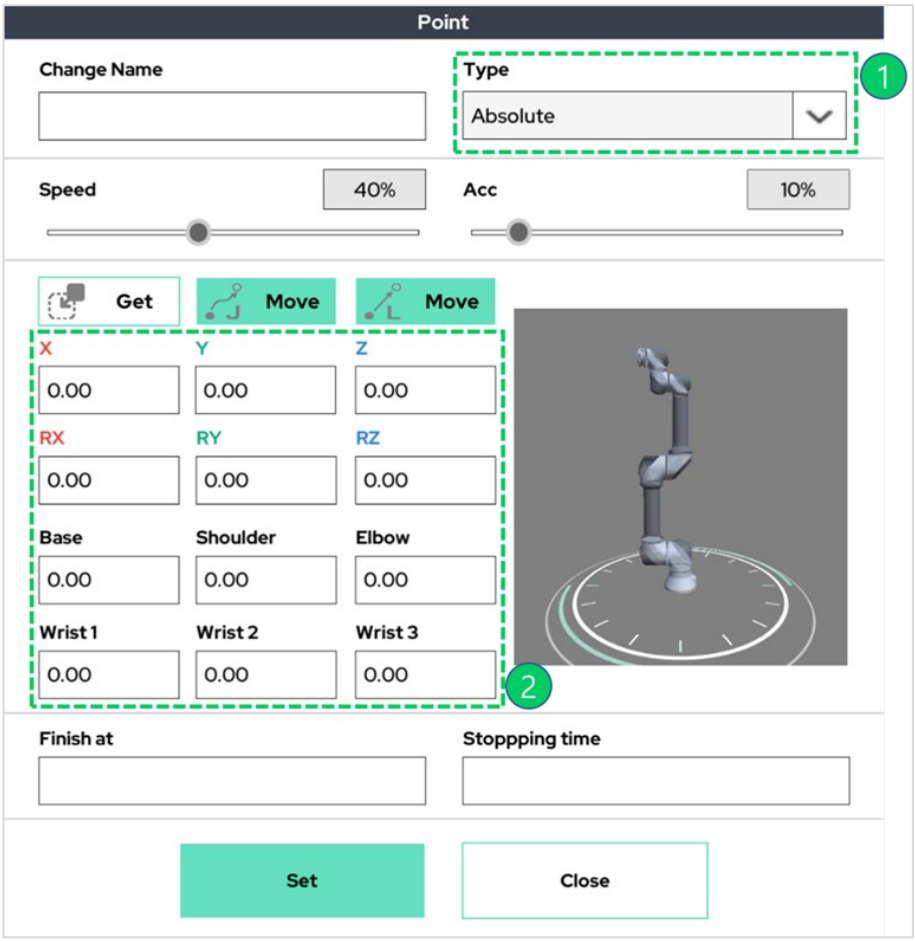


① Relative-Option.

② Wie viel Grad jedes Gelenk relativ zum vorherigen Gelenkwinkel bewegt werden soll, wird eingegeben. Parametrisierte Informationen oder Formeln können eingegeben werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

▷ Linear-Typ – Absolute-Option

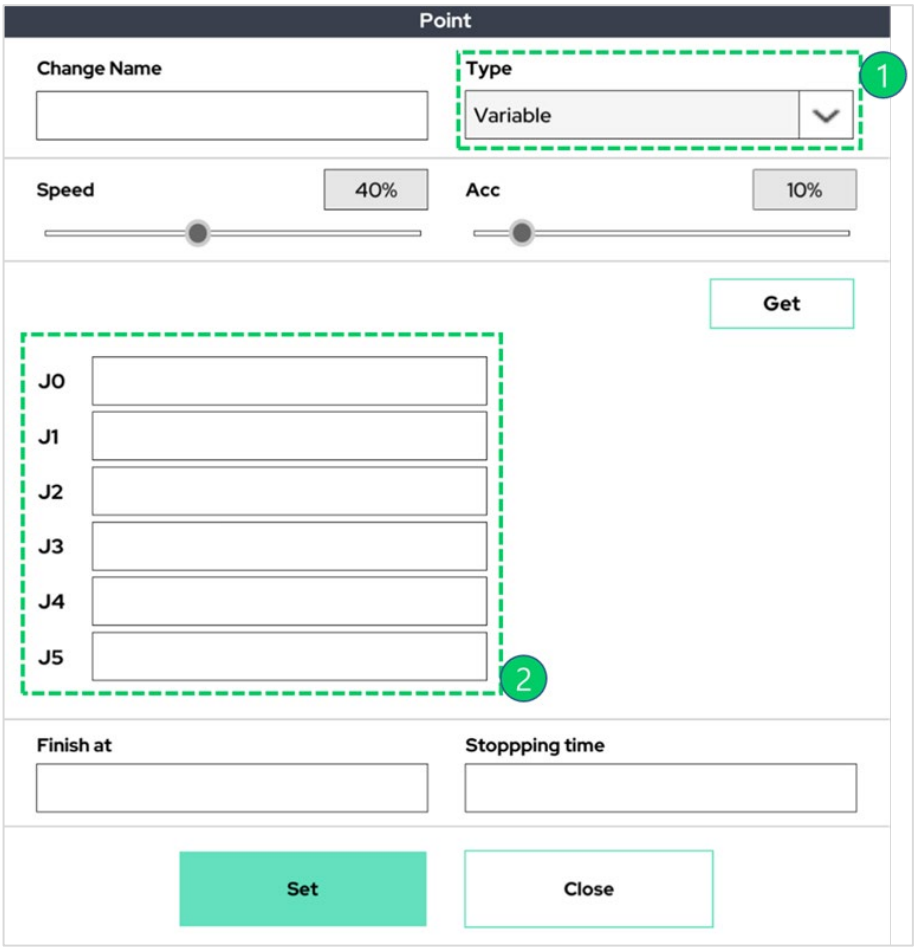


① Absolute-Option.

② Der Haltungs-/ Positionswert des Roboters wird über die Schaltfläche ‘Get (Importieren)’ gespeichert. Die Referenz des kartesischen Koordinatensystemwerts ist das Roboterbasiskoordinatensystem.

© RAINBOW ROBOTICS Inc. All rights reserved.

▷ Linear-Typ / Variable-Option

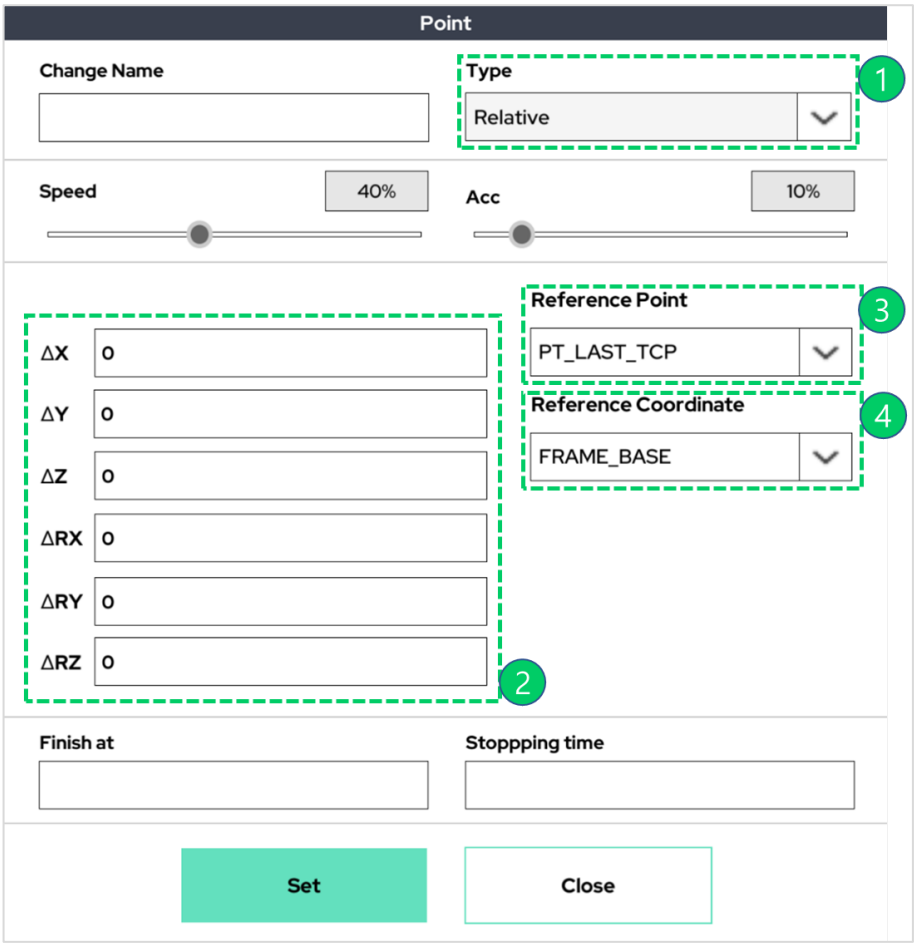


① Variable-Option.

② Benutzer kann die kartesischen Zielkoordinatenwerte frei eingeben. Variable Informationen können auch als Formel eingegeben werden. Die Referenz des eingestellten kartesischen Koordinatenwertes ist das Basiskoordinatensystem des Roboters.

© RAINBOW ROBOTICS Inc. All rights reserved.

▷ Linear-Typ/ Relative-Option



① Relative-Option.

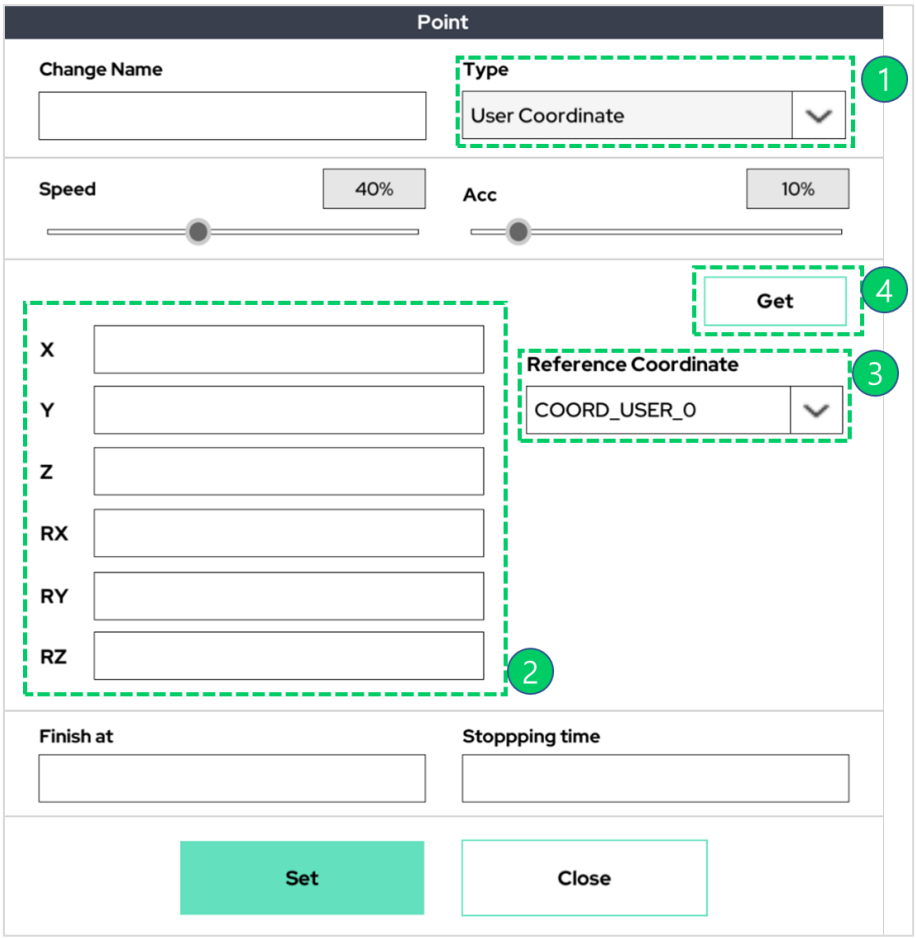
② Geben Sie den Abstand /Winkelversatz (Offset) ein, zu dem sich relative bewegt werden soll. Es ist auch möglich, die parametrisierten Informationen zu verwenden.

③ Mit dieser Funktion wird ein Punkt ausgewählt, zu dem sich relative bewegt werden soll. Der Standardwert ist ‘PT_LAST_TCP’, was den letzten Ankunftspunkt angibt. Außerdem kann es auf einem beliebigen Punkt basieren.

④ Sie können entscheiden, auf welchem Koordinatensystem die Relativbewegung angegeben werden soll. Der Standardwert ist ‚Frame_Base’, der das Basiskoordinatensystem des Roboterarms angibt. Es kann auf das Benutzerkoordinatensystem oder das lokale Koordinatensystem des Werkzeugs geändert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

▷ Linear-Typ - User Coordinate-Option



① User Coordinate (Benutzerkoordinatensystem)-Option.

② Ähnlich wie Variable, aber der Zielpunkt wird mit dieser Funktion basierend auf dem vom Benutzer festgelegten Koordinatensystem festgelegt. Sie können auch parametrisierte Informationen eingeben.

③ Sie können das Benutzerkoordinatensystem auswählen, das Sie als Referenz verwenden möchten.

④ Wählen Sie das gewünschte Referenzkoordinatensystem aus und verwenden Sie die ‚Get(Importieren)'-Funktion, um automatisch die Haltungs-/Positionsinformationen des Roboters basierend auf dem ausgewählten Koordinatensystem einzugeben .

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

Vorsicht:

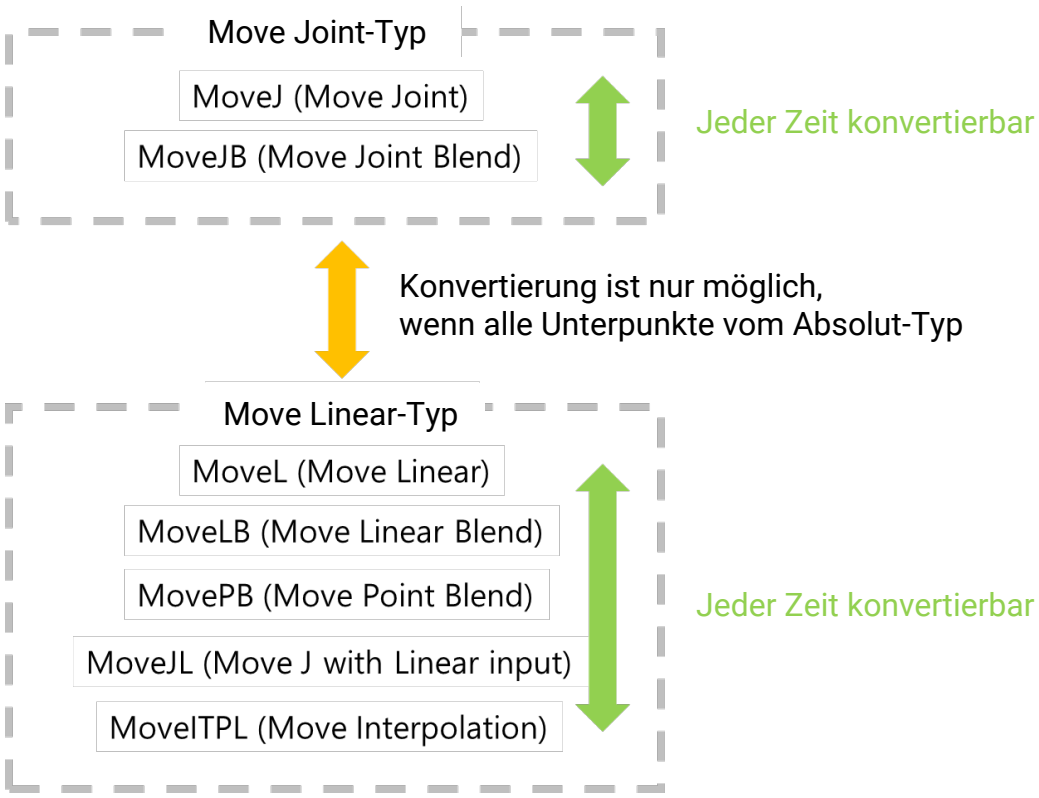
1) Einstellungen von Benutzerkoordinaten (User-Coordinate) können über das ‘Coordinate’-Menü auf dem ‘Setup’-Bildschirm oder ‘Setting’-Funktion auf dem ‘Make’-Bildschirm vorgenommen werden.
2) Bis zu 3 Benutzerkoordinatensysteme können eingestellt werden.
3) Bei Auslieferung ist das grundlegende Benutzerkoordinatensystem auf dasselbe Koordinatensystem wie das Basiskoordinatensystem des Roboters eingestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Ändern der Bewegungseigenschaften

Beim Ändern der Bewegungseigenschaften (Art von Move) in einer bereits konfigurierten Bewegung gelten die folgenden Bedingungen.

- Änderung innerhalb der Gleichen Serie ist ohne Einschränkung möglich.
- Bei der Änderung zu einer anderen Serie (Move Joint-Typ -> Move Linear-Typ / Move Linear-Typ -> Move Joint-Typ) ist sie möglich, wenn der Typ (Option) der in der unteren Ebene verwendeten Point-Funktion ‚Absolute' ist.



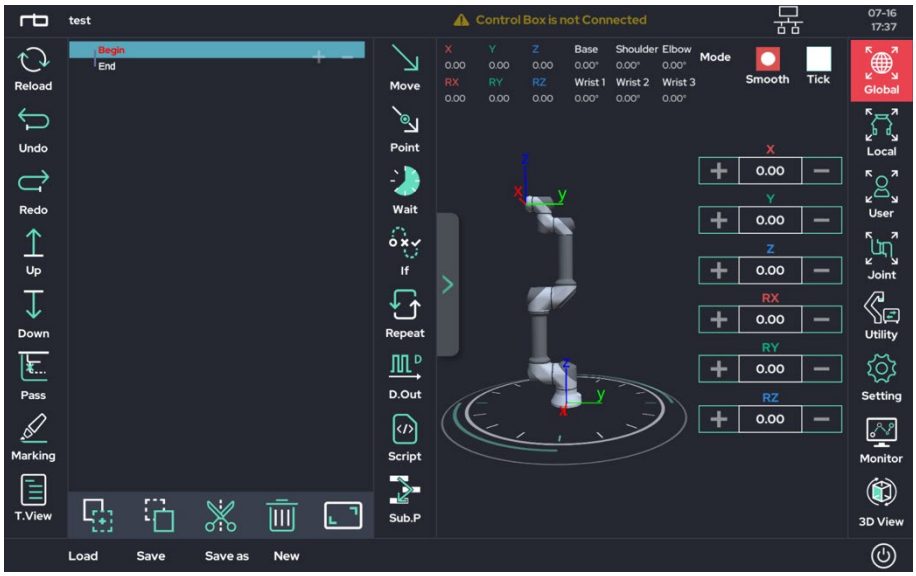
© RAINBOW ROBOTICS Inc. All rights reserved.

- Beispiel für das Teaching/ Erstellen eines Bewegungsprogramms

  Das Folgende ist ein Beispiel für das Erstellen und Ausführen eines einfachen Programms basierend auf den obigen Move- und Point-Funktionen.

[Step 1]

Ein neues Projekt erstellen. Der Projektname wird als ‘test’ genannt.



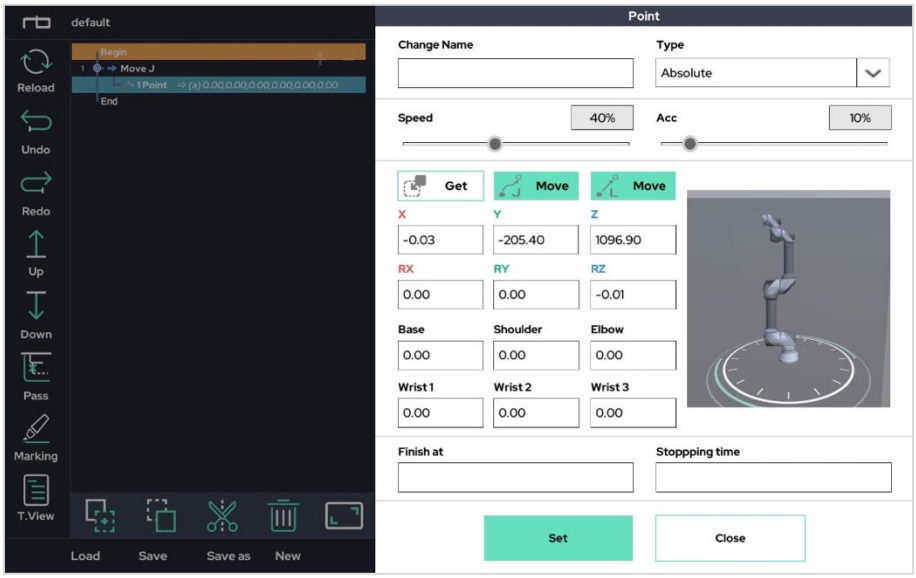
[Step 2]

© RAINBOW ROBOTICS Inc. All rights reserved.

[Step 3]

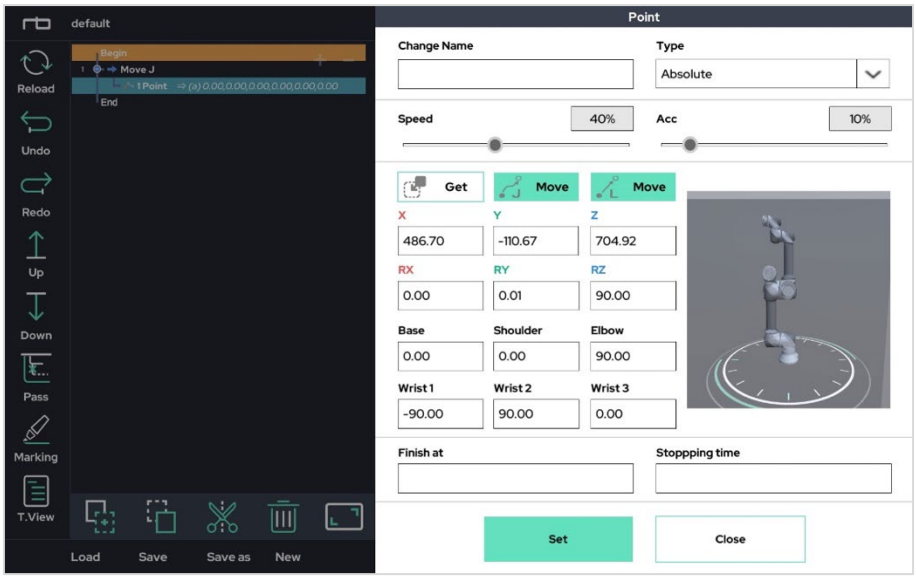
Den Roboter durch Joggen in eine bestimmte Haltung bewegen. In diesem Beispiel wird jede Achse so bewegt, dass der Gelenkwinkel des Roboters [Base:0', Shoulder:0', Elbow:90', Wrist1:-90', Wrist2:90', Wrist3:0'] beträgt.

Wenn Sie im Programmbaum auf die 'Point'-Funktion klicken, erscheint das Popup-Fenster der ‚Point'-Einstellung wie unten gezeigt.



[Step 4]

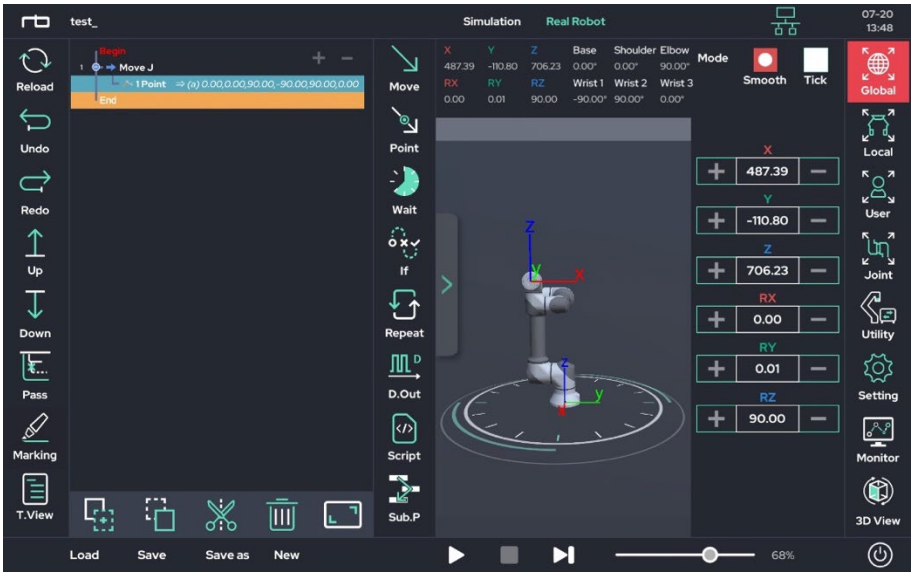
Im Popup-Fenster von 'Point' die Schaltfläche ‚Get (Importieren)' drücken, um den Haltungs- /Winkelwert des aktuellen Roboters abzurufen. Der zuletzte bewegte Haltungswinkel wird genommen. Drücken Sie 'Set(Einstellung)', um diese Einstellung am ' Point' zu speichern.



© RAINBOW ROBOTICS Inc. All rights reserved.

[Step 5]

Das Ergebnis des obigen Prozesses ist wie folgt.



[Step 6]

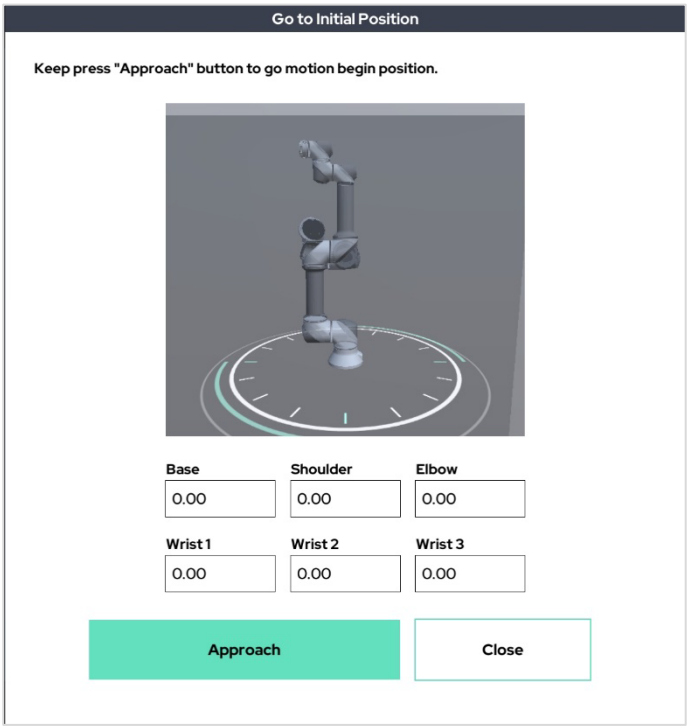
Die obigen Schritte 1 bis 4 werden mehrmals wiederholt, um die gewünschte Roboterbewegung zu lehren. Die Konfiguration des fertigen Beispielprogramms ist wie folgt.

© RAINBOW ROBOTICS Inc. All rights reserved.

[Step 7]

Nachdem das gesamte ‘Teaching‘ abgeschlossen ist, könne Sie das eingelernte Projekt auf dem Betriebsbildschirm ausführen, um den echten Roboterarm zu bewegen. Wenn Sie eine Simulation ausführen möchten, verwenden Sie den Simulation-Modus, und wenn Sie einen echten Roboter betreiben möchten, verwenden Sie den Real-Modus.

Wenn Sie auf die Play-Schaltfläche(▷) am unteren Rand des Betriebsbildschirms klicken, erscheint ein Bildschirm, der den Roboterarm in die Startposition (in ‚begin‘ definierte Position) bewegt, wie in der folgenden Abbildung gezeigt.



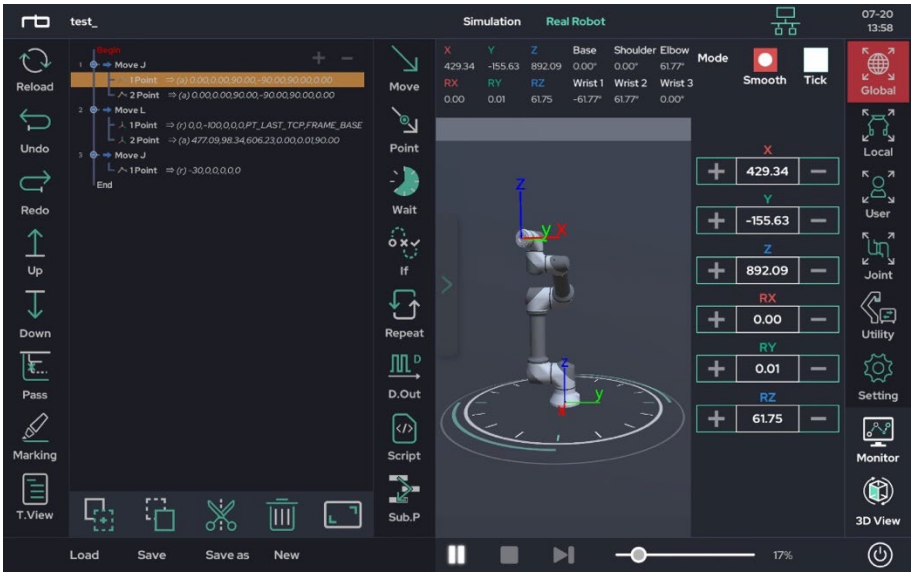
Wenn Sie hier die Move-Taste gedrückt halten, bewegt sich der Roboterarm in die Ausgangsposition. Wenn die Bewegung beendet ist, erscheint eine Popup-Meldung, dass die Startposition erreicht wurde.

© RAINBOW ROBOTICS Inc. All rights reserved.

[Step 8]

Hat der Roboter die Startposition erreicht, ist das Programm startbereit. Klicken Sie erneut auf die Play-Schaltfläche unten, um das Programm auszuführen.

Das folgende Bild zeigt das laufende Programm.



© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

Vorsicht:

1) Der aktuell in Betreib befindliche Teil wird auf der Benutzeroberfläche gelb angezeigt.

- Anfängliche Bewegungsposition

  Die Ausgangshaltung wird im ‘Begin‘ des Programms angegeben.

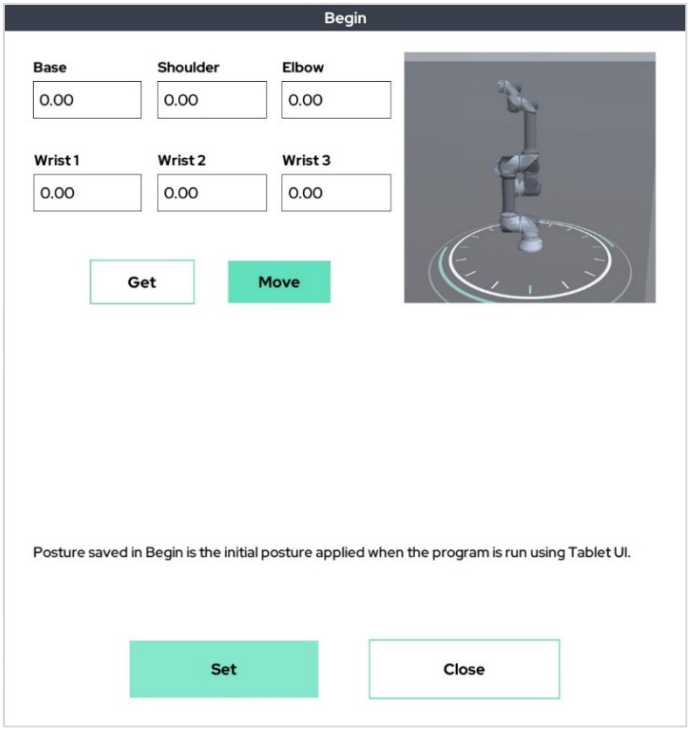
  Beim Betrieb des Roboters über Tablet-UI bewegt er sich immer in die Begin-Position (Startposition), bevor das Programm zum ersten Mal gestartet wird.

  Die Startposition kann auf folgende Weise geändert werden.

  1) Bewegen Sie den Roboter in die gewünschte Startposition. Verwenden Sie Jog oder Direkt-Teaching.

  2) Klicken Sie auf ‘Start’ im Programmbaum.

  3) Klicken Sie auf ‚Get (Importieren)’-Schaltfläche, um den aktuellen Haltungswinkel des Roboters einzugeben. Und klicken Sie auf ‚Set (Einstellung)‘-Schaltfläche, um diese Haltung zur Starthaltung zurückzubringen.



© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht:

1) Wenn das Programm zum ersten Mal erstellt wird, ist die standardmäßige Starthaltung (in Begin eingestellter Winkelwert) eine Haltung, in der alle Gelenkwinkel 0 betragen.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Kollisionserkennung während des Betriebs

Dieses Produkt verfügt über zwei integrierte Kollisionserkennungsfunktionen.

- Externe Kollisionserkennung (Out-Collision Detection)
- Selbstkollisionserkennung (Self-Collision Detection)

[Externe Kollision(External Collision)]

[Selbstkollision(Self Collision)]

● Externe Kollisionserkennung (Out Collision Detection)

- Ein nicht im Voraus definierter externer Schock / Kollision wird erkannt.

- Unerwartete Kollisionen mit Personen oder der äußeren Umgebung werden erkannt.

- Die Kollisionsempfindlichkeit kann im Setup geändert werden.

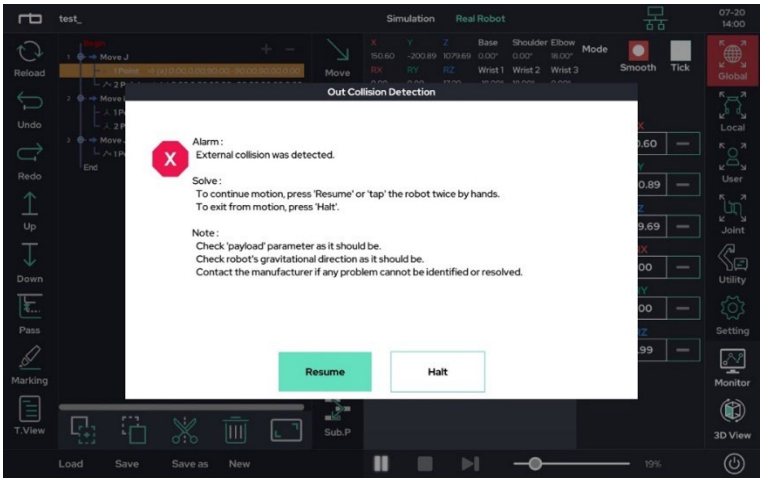
- Die Kollisionsempfindlichkeit während der Programmausführung wird in Echtzeit im Set-Funktionen geändert.

- Für eine genaue Kollisionserkennung muss die Last /der Schwerpunkt des Werkzeugs genau eingestellt werden.

- Mit der empfindlichen Kollisionserkennungseinstellung kann das Betrieb mit schneller Beschleunigung/ Verzögerung usw. als Kollision erkannt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

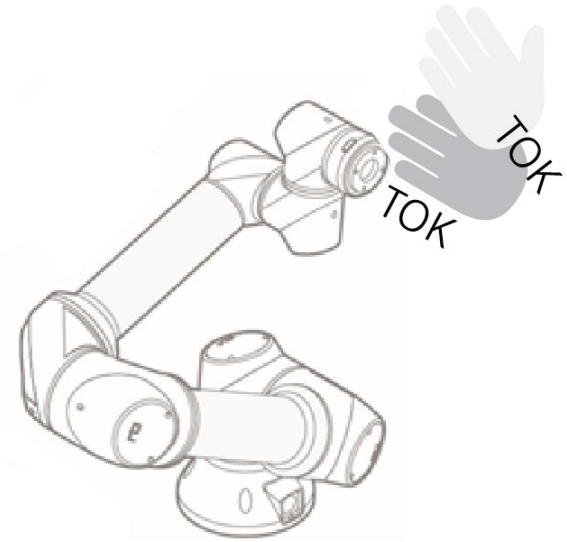
Wenn eine externe Kollision während des Betriebs der Roboterarms im Real-Modus erkannt wird, wird das folgende Popup-Fenster angezeigt.



- Resume(Fortsetzen)-Schaltfläche: Die Situation bestätigen und die Aktion fortsetzen.
- Halt(Halten)-Schaltfläche: Das Programm beenden.

Im obigen Popup-Fenster können Sie die Schaltfläche, Resume(Fortsetzen)‘ drücken, um die Bewegung des Roboters fortsetzen, oder die Schaltfläche, Halt(Halten)‘, um das Programm zu beenden.

Wenn Sie den Roboter außerdem zweimal mit der Hand antippen, wird die Aktion so fortgesetzt, als ob die Schaltfläche, Resume(Fortsetzen)‘ gedrückt worden wäre.



© RAINBOW ROBOTICS Inc. All rights reserved.

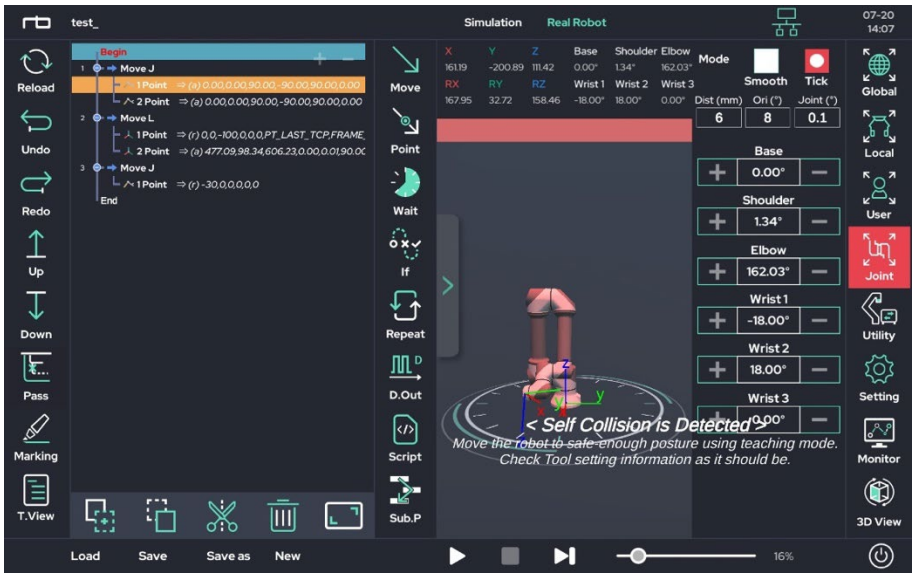
- Selbstkollisionserkennung (Self-Collision Detection)

  - Wenn der Benutzer einen bestimmten Bewegungsbefehl gibt, sagt der Roboter voraus, dass der Roboter mit sich selbst kollidieren und anhalten wird.

  - Der Roboter stoppt selbst, wenn er von dem voreingestellten Betriebsbereich (Workspace) abweicht. Die Einstellung für den Umgebungsbereich wird im Setup-Bildschirm fortgesetzt.

  - Sie können auch eine virtuelle Kollisionserkennungsbox auf dem Werkzeug einrichten, sodass der Roboter automatisch stoppt, wenn erwartet wird, dass die Box mit dem Roboter selbst kollidiert oder sich aus dem Umgebungsbereich bewegt. Die im Werkzeug installierte virtuelle Box wird im Setup-Tool eingestellt.

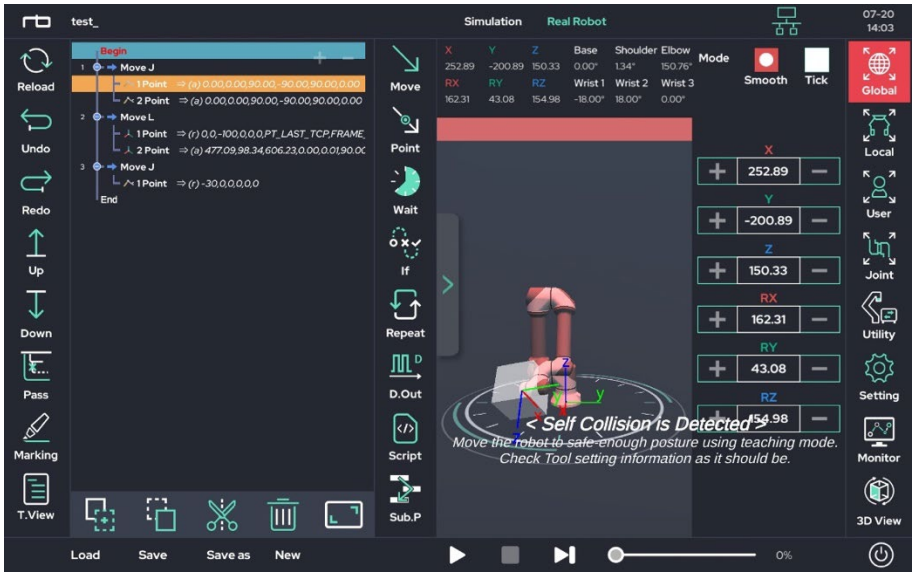
Die folgende Abbildung zeigt die Situation, in der der Benutzer den Roboter durch eine Fehlfunktion beim Jog von selbst kollidieren ließ. Unmittelbar vor der Kollision sagt der Roboter die Kollision selbst voraus und stoppt und zeigt eine Warnung in Rot auf der Benutzeroberfläche.



© RAINBOW ROBOTICS Inc. All rights reserved.

Die folgende Abbildung zeigt einen Fall, in dem der Roboter dabei ist, den benutzerdefinierten Betriebsbereich zu verlassen. Kurz bevor er den Betriebsbereich verlässt, stoppt der Roboter und fordert die Benutzeroberfläche auf, eine Warnung in Rot anzuzeigen.

Die folgende Abbildung zeigt einen Fall, in dem die vom Benutzer eingerichtete virtuelle Kollisionsbox eine Kollision erkennt/ vorhersagt. Der Roboter stoppt und fordert die Benutzeroberfläche auf, eine Warnung in Rot anzuzeigen.



Wenn der Roboter auf einer Selbstkollisionserkennung während des Betriebs im Real-Modus stoppt, verwenden Sie die direkte Teaching-Funktion, um die Roboterhaltung in eine sichere Position zu ändern und ihn weiter zu verwenden.

© RAINBOW ROBOTICS Inc. All rights reserved.

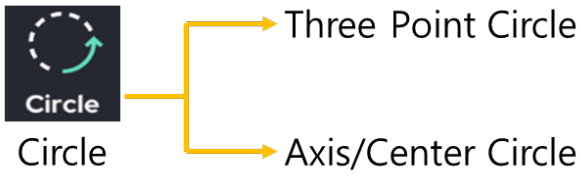
7.4 Detaillierte Beschreibung der wichtigsten Teaching-Symbole

Im vorherigen Abschnitt 6.3 wurden nur die grundlegende Teaching-Funktion (Move- und Point-Funktionen) beschrieben. Dieser Abschnitt ist den anderen Teaching-Symbolen /Funktionssymbolen gewidmet.

■ Circle-Funktion :

Abgesehen von Move werden Funktionen für Kreisbewegungen geboten.

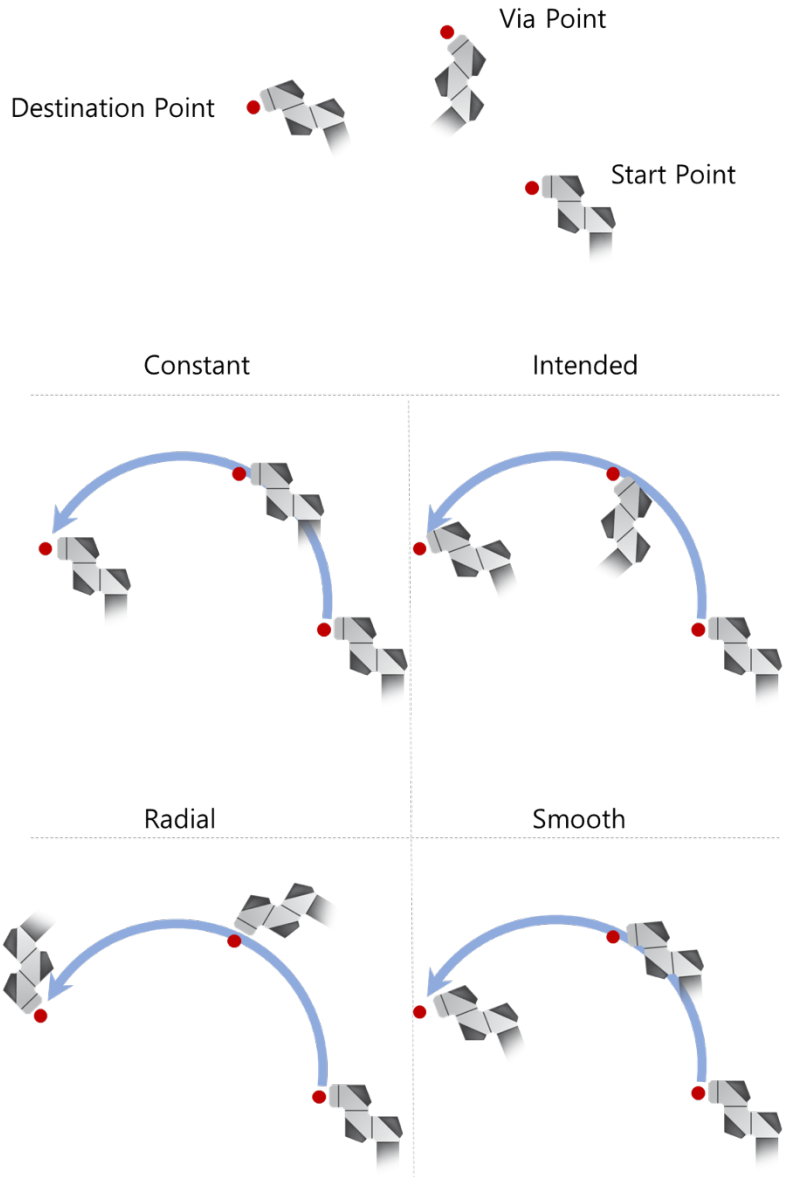
Es gibt ein Typ, der einen Bogen zeichnet, der durch 3 Punkte läuft, indem er den Startpunkt und zwei Punkte nimmt, und es gibt einen Typ, der einen Kreis zeichnet, indem er den Mittelpunkt und die Rotationsachse festlegt.



- Three Point Circle (Dreipunktkreis): Ein Modus zum Zeichen eines Bogens mit drei Punktinformationen
- Axis/Center Circle (Achse/Mittelkreis): Ein Modus zum Zeichen eines Bogens/ Kreises mit Rotationzentrum und Rotationachseninformationen

Abgesehen von der Art, einen Kreis zu zeichnen, bietet die Circle-Funktion vier Orientierungsoptionen (Orientation Option), wie unten gezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.

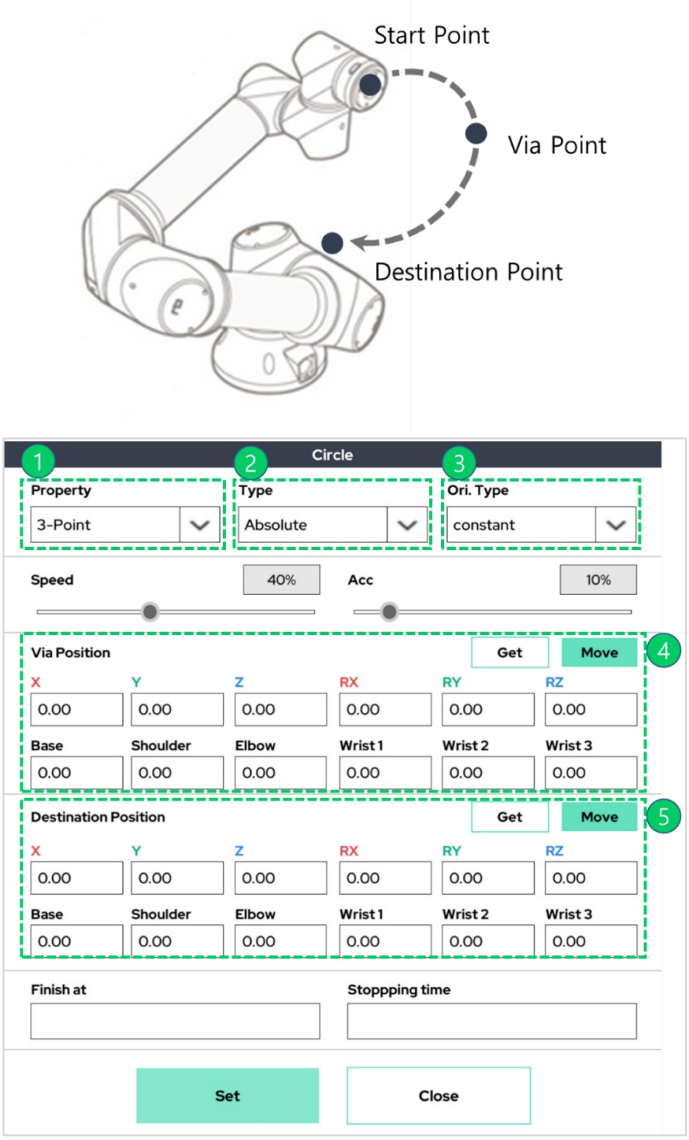


- Constant (Konstant): Die Rotationshaltung des Roboterendes (TCP) wird beibehalten, während eine Rotationsbewegung ausgeführt wird.
- Intended (Beabsichtigt): Der vom Benutzer hinterlegten Rotationshaltung wird gefolgt.
- Radial (Radial): Die Rotationshaltung des Roboterendes (TCP) wird in Bezug auf den Mittelpunkt der Rotation zusammen rotiert.
- Smooth: Die Rotation ändert sich direkt vom Startpunkt zum Endpunkt. Rotationsinformationen von Wegpunkten werden ignoriert.

© RAINBOW ROBOTICS Inc. All rights reserved.

Three Point Circle (3-Punkt-Einstellung)

Es ist ein Typ, der einen Bogen zeichnet, der drei Punkt verbindet, basierend auf den Informationen des Startpunkts, des Zwischenpunkts und des Zielpunkts.

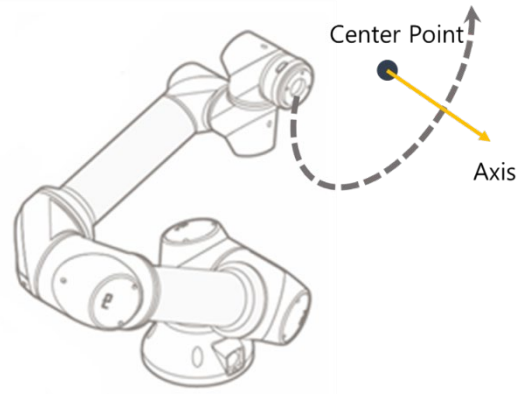


① Typ für kreisförmige Bewegung (Typ mit 3-Punkt-Einstellung)
② Punkteigenschaften (Absolute / Variable / Relative / UserCoord.)
③ Rotationsoptionen (Constant / Radial / Intended / Smooth)
④ Informationen zum Wegpunkt (Via Point)
⑤ Informationen zum Zielpunkt (Destination Point)

© RAINBOW ROBOTICS Inc. All rights reserved.

Axis/Center Circle (Achse-/ Mittelpunkt-Einstellung)

Dieser Typ führt kreisförmige Bewegung aus, indem er den Mittelpunkt, die Rotationsachse und den Rotationswinkel für die kreisförmige Bewegung festlegt.



① Typ für kreisförmige Bewegung (Achse-/ Mittelpunkt-Einstellung)
② Punkteigenschaft (Absolute / Variable / Relative / UserCoord)
③ Rotationsoptionen (Constant / Radial / Intended / Smooth)
④ Informationen zum Mittelpunkt (Center Point)
⑤ Informationen zur Rotationsachse (Axis)
⑥ Informationen zum Rotationswinkel (degree)

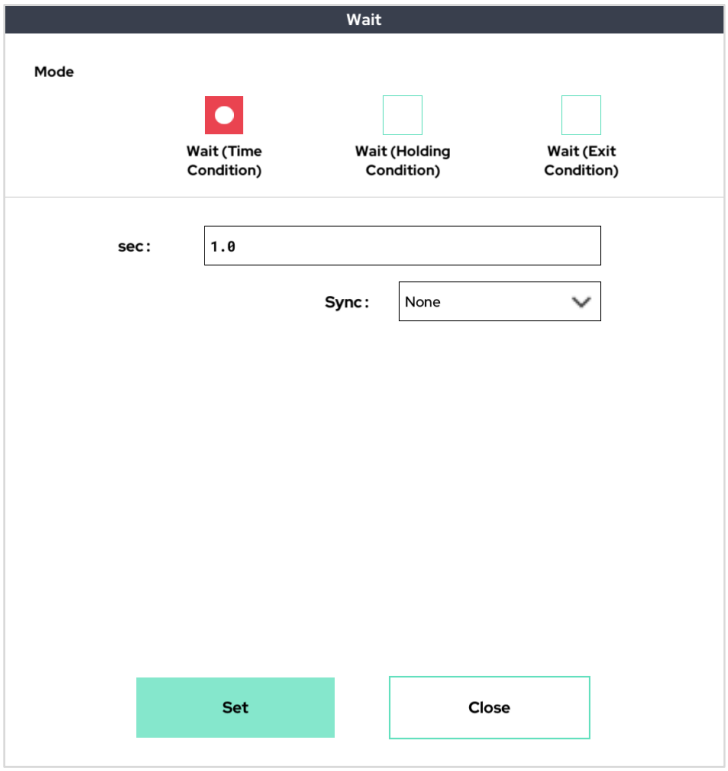
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Wait-Funktion:

Mit dieser Funktion wird auf den Betrieb unter bestimmten Bedingungen oder für eine bestimmte Zeit gewartet.

Es gibt 3 Modi. Eine Funktion zum Warten für eine bestimmte Zeit, eine Funktion zum Warten, solange eine Bedingung wahr ist, und eine Funktion zum Warten, bis die Bedingung wahr ist.

1) Die Funktion zum Warten für eine angegebene Zeit (Time Condition)

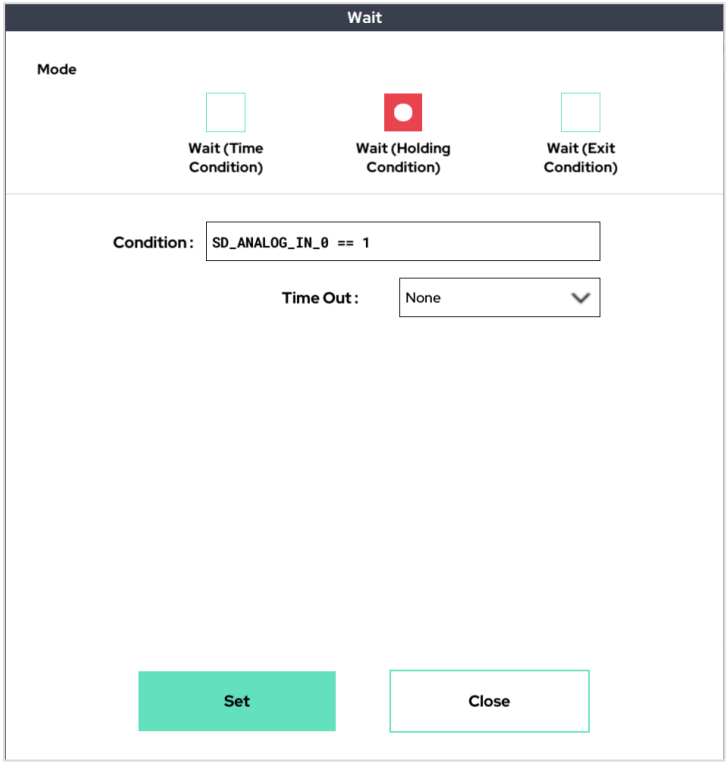


Bsp.) Warten Sie für die angegebene Zeit (3 Sekunden) und dann wird der nächste Befehl ausgeführt

Bei Verwendung der Synchronisationsfunktion von Geschwindigkeitsregler in ‚Sync' wird die Wartezeit umgekehrt proportional zum Wert des Geschwindigkeitsreglers angepasst.

© RAINBOW ROBOTICS Inc. All rights reserved.

2) Die Funktion zum Warten, während die angegebene Bedingung wahr ist (Holding Condition)

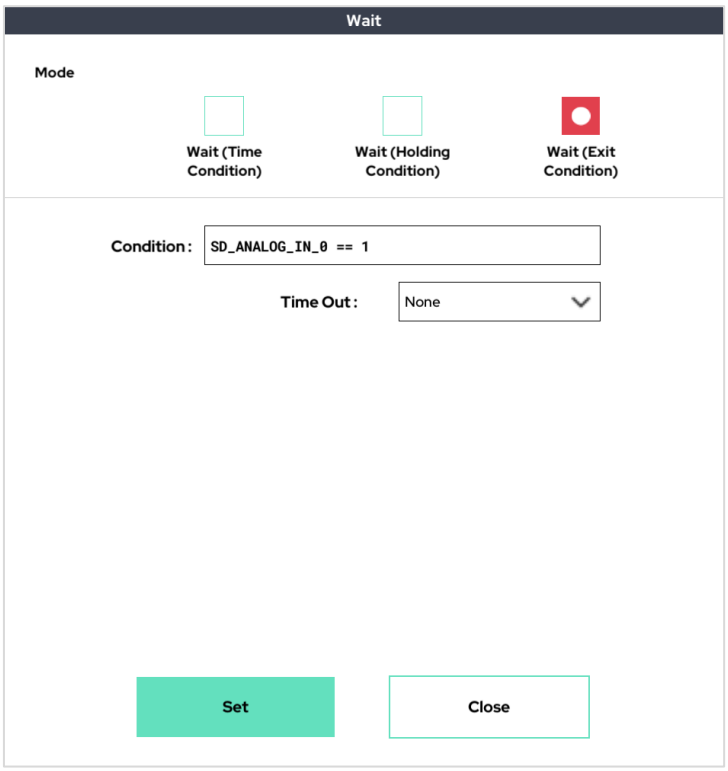


Bsp) Wenn die in ‘Condition‘ eingegebene Bedingung wahr ist, wird weiter gewartet.

Time Out-Funktion ist eine Funktion zum Verhindern eins kontinuierlichen Wartens, bis die bedingte Anweisung falsch wird, obwohl die Anweisung nicht falsch sein kann. Verlassen Sie die Wait-Funktion nach Ablauf der angegebenen Zeit.

© RAINBOW ROBOTICS Inc. All rights reserved.

3) Die Funktion zum Beenden des Wartens, wenn die angegebene Bedingung wahr ist (Exit Condition)



Bsp.) Wenn die in 'Condition' eingegebene bedingte Anweisung wahr ist, wird das Warten beendet und der nächste Befehl ausgeführt.

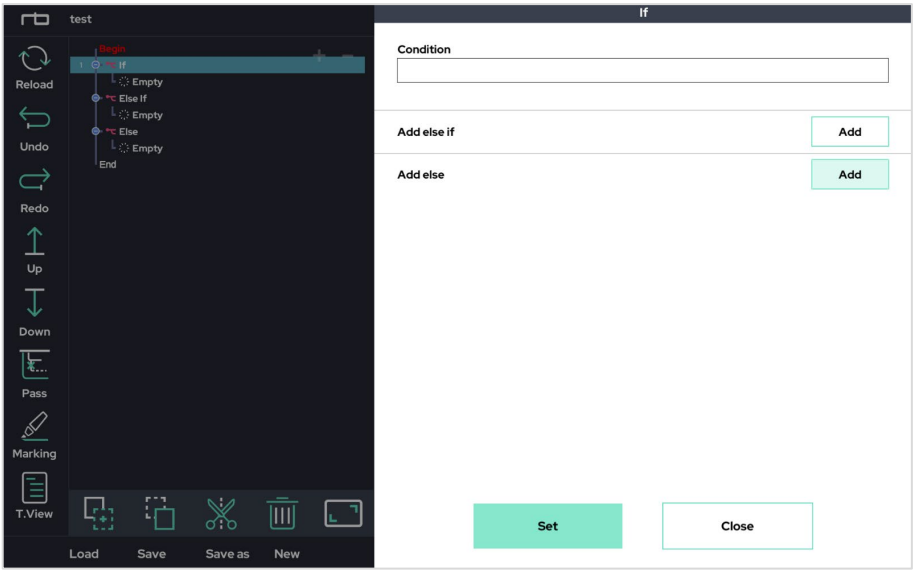
Time Out-Funktion ist eine Funktion zum Verhindern eins kontinuierlichen Wartens, bis die bedingte Anweisung falsch wird, obwohl die Anweisung nicht falsch sein kann. Verlassen Sie die Wait-Funktion nach Ablauf der angegebenen Zeit.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ If-Funktion:

If-Anweisung. Verzweigungen können eingerichtet werden, sodass der Roboter je nach Bedingungen verschiedene Befehle ausführen kann. If-/else if- /else-Anweisungen können verwendet werden.

Nachdem Sie die If-Funktion zum Programmbaum hinzufügt haben, klicken Sie auf die hinzugefügte If-Funktion. Dann wird das folgende Popup-Fenster angezeigt. Geben Sie hier direkt die konditionale Anweisung ein, die in der If-Anweisung verwendet werden soll.



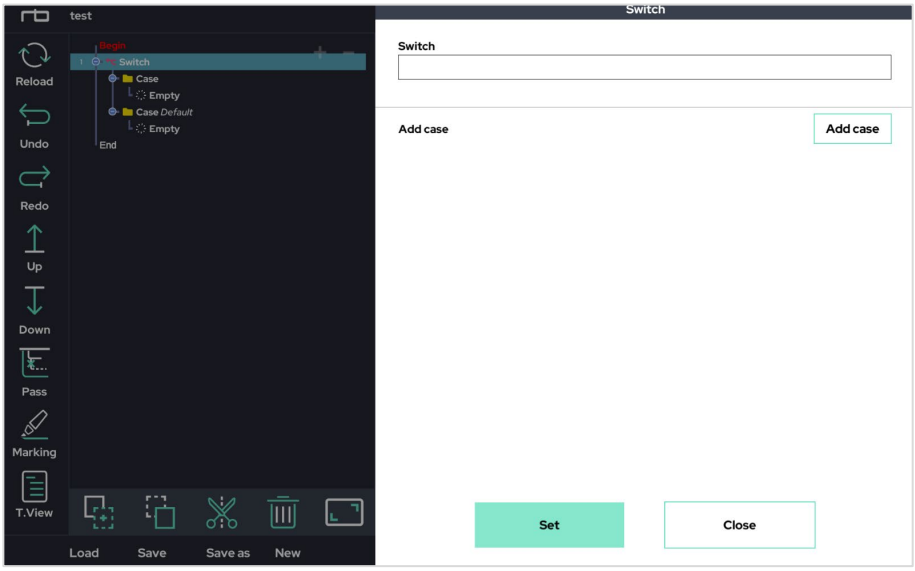
Wenn Sie für die Else if- oder Else-Funktion ‘Add else if’-Schaltfläche oder ‘Add else’-Schaltfläche drücken, können Sie eine Verzweigung einer konditionalen Anweisung erstellen.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Switch-Funktion:

Switch-Anweisung. Verzweigungen können eingerichtet werden, sodass der Roboter je nach Bedingungen verschiedene Befehle ausführen kann. Switch-/case-Anweisungen können verwendet werden.

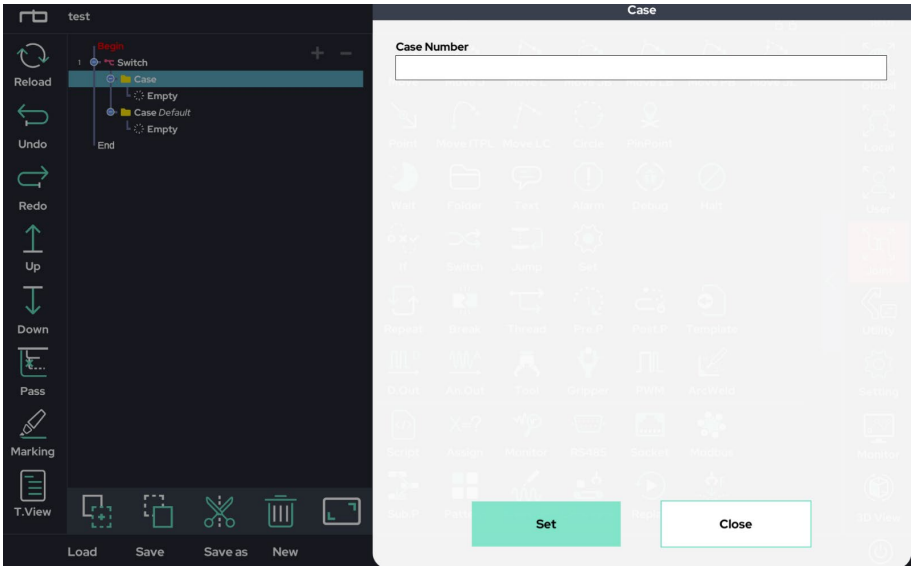
Nachdem Sie die Switch-Funktion zum Programmbaum hinzufügt haben, klicken Sie auf die hinzugefügte Switch-Funktion. Dann wird das folgende Popup-Fenster angezeigt. Geben Sie hier den Referenzfaktor für die auszuführende Switch-Anweisung ein.



Beim Erstellen der ersten Switch-Anweisung wird ‘default‘ automatisch erstellt. Wenn eine zusätzliche Case-Anweisung erforderlich ist, kann eine Case-Anweisung mit der ‚Add case‘-Schaltfläche hinzugefügt werden.

Wenn Sie auf die hinzugefügte Case-Anweisung klicken, erscheint das folgende Fenster: Sie können den Standardfaktor bestimmen, der in der Switch-Anweisung verwendet wird, und den Case-Anweisungsfaktor in dieses Fenster eingeben.

© RAINBOW ROBOTICS Inc. All rights reserved.



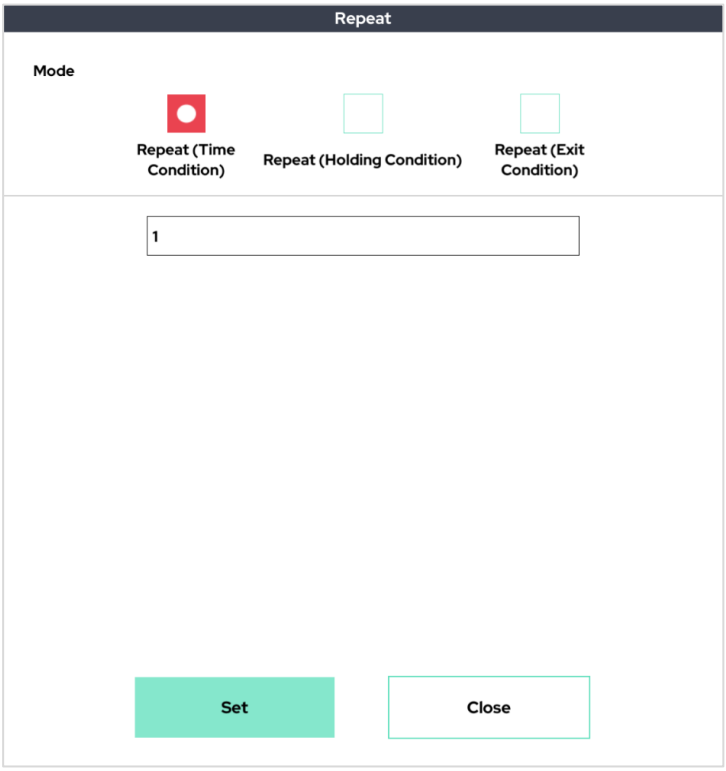
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Wiederholen(Repeat)-Funktion:

Wiederholungsanweisung. Das Unterprogramm wird für die angegebene Bedingung oder Anzahl wiederholt. Es gibt 3 Modi. Eine Funktion zum Wiederholen für eine bestimmte Anzahl, eine Funktion zum Wiederholen, solange eine bestimmte Bedingung wahr ist, und eine Funktion zum Beenden der Wiederholung, wenn die Bedingung wahr ist.

Wenn Sie die Wiederholungsfunktion zum Programm hinzufügen und auf ‚Repeat' klicken, finden Sie die folgenden 3 Optionen. Wählen und verwenden Sie die gewünschte Funktion.

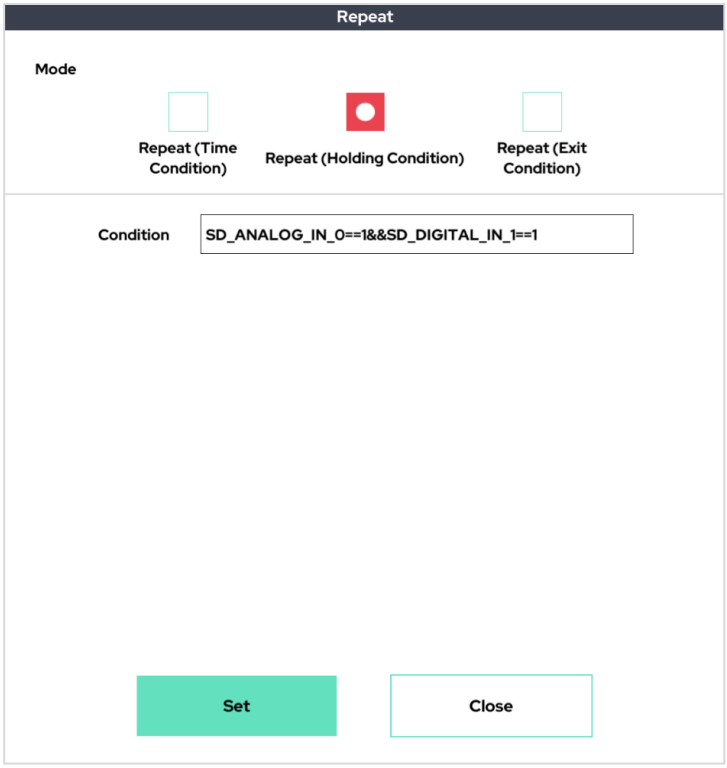
1) Die Funktion zur Wiederholung so oft wie angegeben (Time Condition)



Bsp.) Einmal wird das Unterprogramm wiederholt.

© RAINBOW ROBOTICS Inc. All rights reserved.

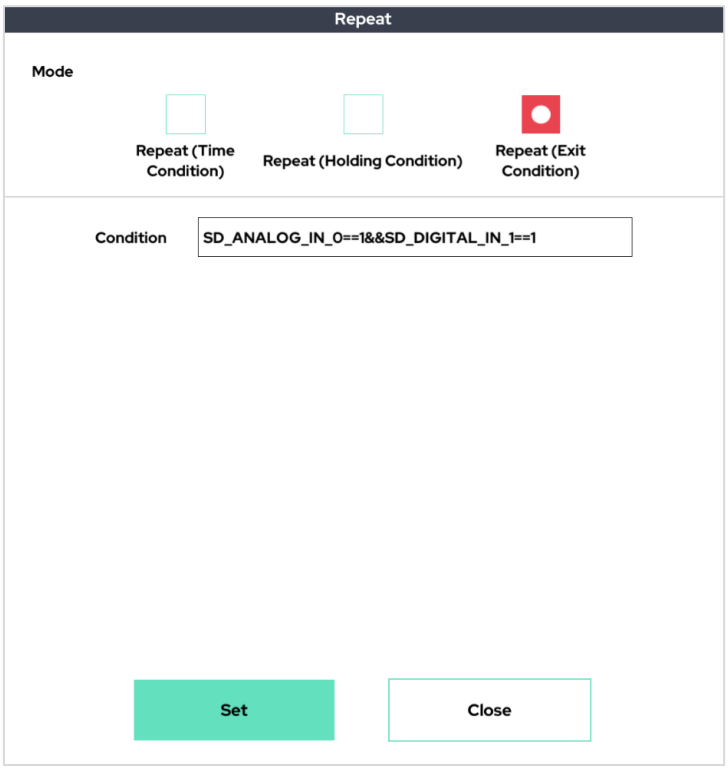
2) Eine Funktion zur Wiederholung, solange eine bestimmte Bedingung wahr ist (Holding Condition)



Bsp.) Solange die in 'Condition' eingegebene Bedingung wahr ist, wird es ständig wiederholt.

© RAINBOW ROBOTICS Inc. All rights reserved.

2) Die Funktion zum Beenden der Wiederholung, wenn die angegebene Bedingung wahr ist (Exit Condition)



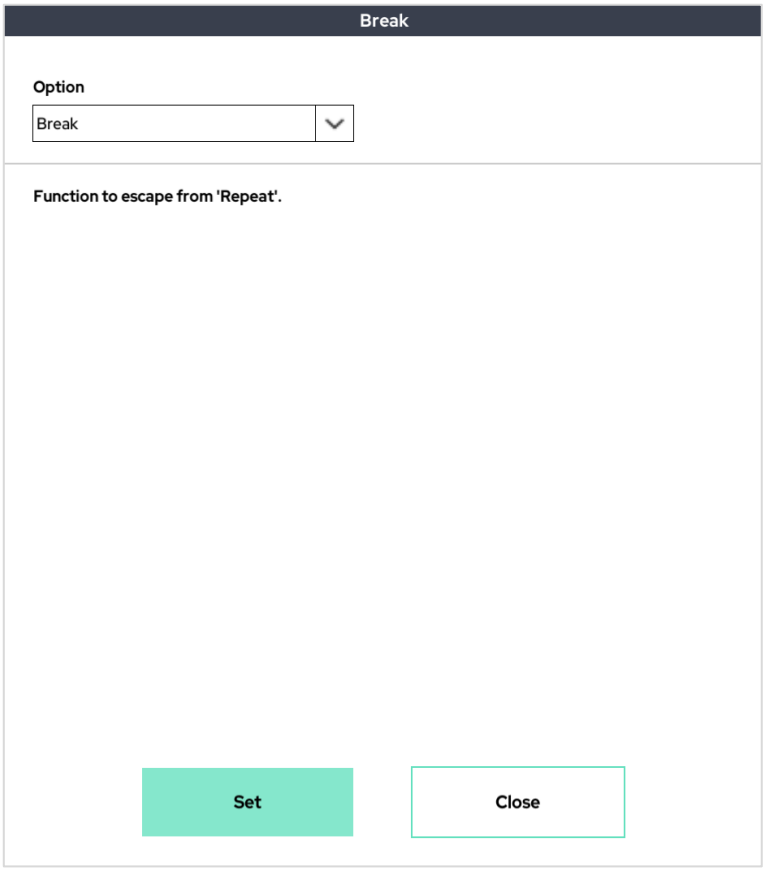
Bsp.) Wenn die in 'Condition' eingegebene konditionale Anweisung wahr ist, wird die Wiederholung beendet und der nächste Befehl ausgeführt.

© RAINBOW ROBOTICS Inc. All rights reserved.

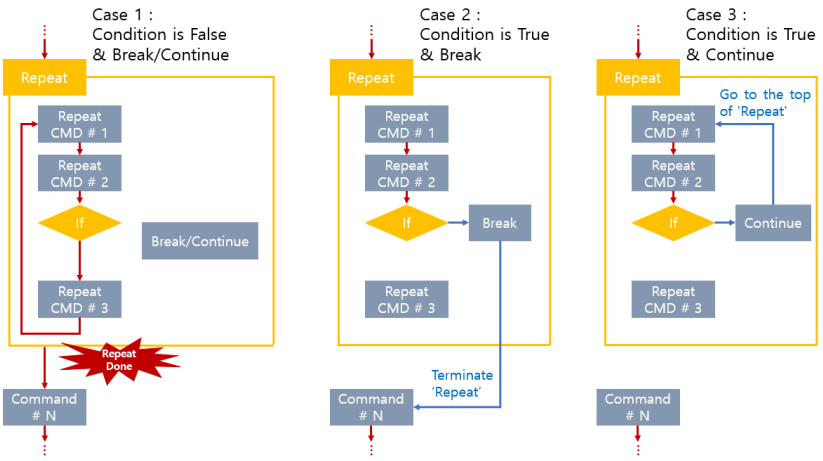
■ Break-Funktion:

Die Funktion zum zwangsweisen Beenden von Repeat-Anweisung (Wiederholung zu stoppen) oder zum Verschieben zu Anfang von Repeat-Anweisung (fortsetzen). Auch wenn die Repeat-Bedingung bestimmt, dass das Unterprogramm fortgesetzt werden soll, kann die ‚Break'-Funktion verwendet werden, um die Wiederholung zu umgehen. Die ‚Continue'-Funktion wird in der 'Repeat'-Funktion verwendet, und wenn sie verwendet wird, bewegt sie sich zum Angang der Wiederholung, ohne das Programm auszuführen.

Nur unter der 'Repeat'-Funktion kann funktionieren. Jede andere Syntax hat keine Auswirkung auf das Programm.



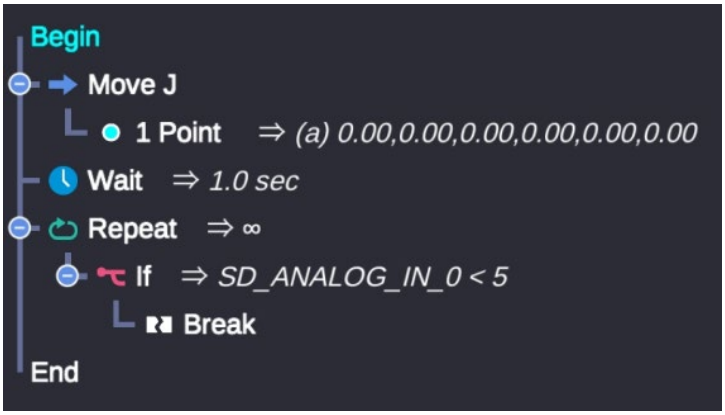
© RAINBOW ROBOTICS Inc. All rights reserved.



Wenn ‚Repeat', ‚Break' und ‚Continue' verwendet werden, wird das wie in der Abbildung oben gezeigt betrieben.

* Beispiel von ‚break'-Funktion

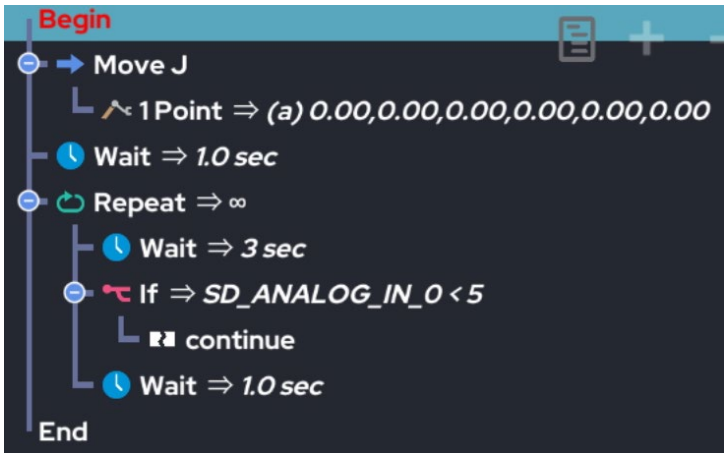
Im folgenden Beispiel gibt es eine Repeat-Anweisung für endlose Wiederholung. Und das Programm ist so konfiguriert, dass die Break-Funktion gemäß der If-Bedingung unten ausgeführt wird. Wenn die Bedingung nicht eintritt, wird die Repeat-Anweisung immer wieder wiederholt. Sobald jedoch die Bedingung eintritt, wird die Repeat-Anweisung verlassen und der nächste Befehl ausgeführt.



© RAINBOW ROBOTICS Inc. All rights reserved.

* Beispiel von ‘Continue’-Funktion

Das folgende Beispiel hat eine Repeat-Anweisung für endlose Wiederholung. Und das Programm ist so konfiguriert , dass die ‘Continue’-Funktion gemäß ‚If‘-Bedingung ausgeführt wird. Wenn die Bedingung nicht eintritt, wiederholt die Repeat-Anweisung den gesamten Inhalt. Sobald jedoch die Bedingung eintritt, wird der Befehl oben in der ‚Continue‘-Funktion ausgeführt, ohne dass der Inhalt unten in der ‚Continue‘-Funktion ausgeführt wird.

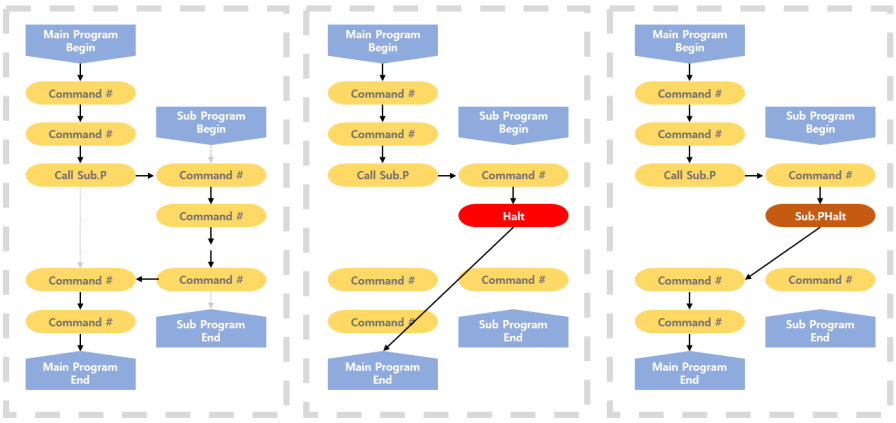


© RAINBOW ROBOTICS Inc. All rights reserved.

■ Halt-Funktion:

Halt-Funktion zum Beenden des Programms. Wenn sie beim Auftritt einer bestimmten Bedingung das Programm beenden müssen, können Sie diese Funktion verwenden.

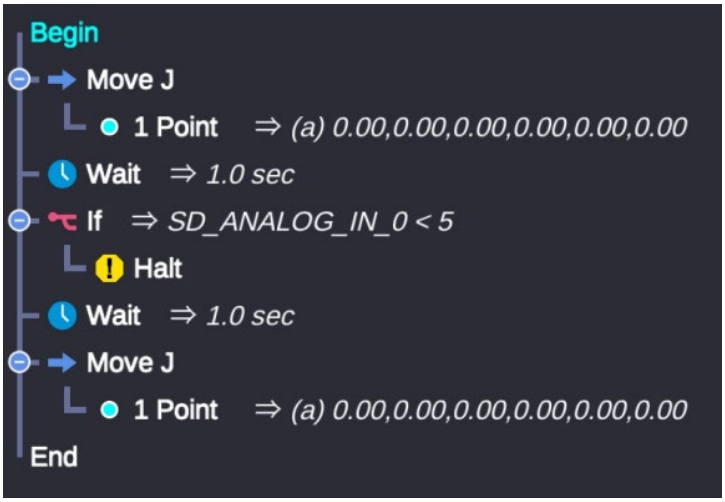
Halt-Funktion wird in ‘Halt’ und ‚Sub.P Halt’ unterteilt. Halt-Funktion zum Beenden des Hauptprogramms unabhängig davon, ob sie im Hauptprogramm oder im Unterprogramm ausgeführt wird. ‚Sub.P Halt’-Funktion muss innerhalb des Unterprogramms verwendet werden, und sobald sie ausgeführt wird, endet das Unterprogramm und kehrt zum Hauptprogramm zurück. Siehe bitte das Diagramm unten.



common case | during the subprogram When Halt is called | during the subprogram When Sub.P Halt is called

Im folgenden Beispiel ist das Programm so konfiguriert, dass ‘If‘-Anweisung überprüft wird und die Halt-Funktion aufgerufen wird, wenn die konditionale Anweisung wahr ist. Wenn diese konditionale Anweisung wahr ist, endet das Programm an dieser Stelle und führt die folgenden Anweisungen nicht aus.

© RAINBOW ROBOTICS Inc. All rights reserved.



Vorsicht

Vorsicht:

1) Wenn die Halt-Funktion ausgeführt wird, wird nicht nur das Hauptprogramm, sondern auch die Thread-Funktion beendet.

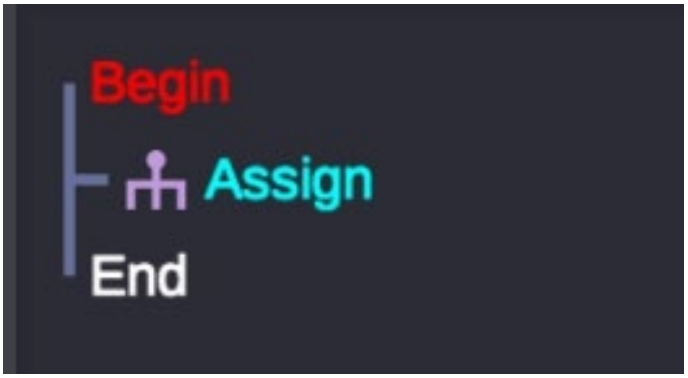
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Assign-Funktion:

Um die Variabelisierungsfunktion zu verwenden, ist dies die Assign-Funktion für die anfängliche Variablendeklaration. Variablentypen können in 4 Typen deklariert werden. Es gibt einen Variable-Typ für die Speicherung einer einzelnen Zahl, einen Array-Typ für die Speicherung einer Array-Zahl, Point-Typ für die Speicherung der Haltungsinformationen und String-Typ für die Speicherung von Zeichenfolge.

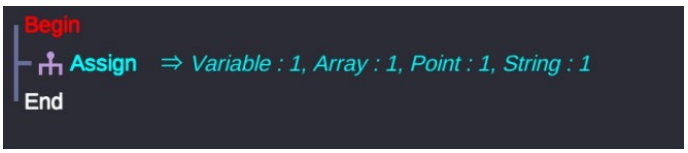
- Variable-Typ: Eine einzelne Zahl wird gespeichert.
- Array-Typ: Ein Array wird gespeichert (die maximale Länge eines Arrays beträgt 10).
- Point-Typ: Haltungsinformationen werden gespeichert (x, y, z, Rx, Ry, Rz werden gespeichert).
- String-Typ: Zeichenfolge wird gespeichert.

Wenn die ‘Assign’-Funktion zum Programmbaum hinzugefügt wird, wird sie wie folgt angezeigt.



Wenn Sie auf die hinzugefügte ‘Assign’ klicken, wird das folgende Popup angezeigt. Sie können hier alle Variablen deklarieren. Wenn Sie mehrere Variablen gleichzeitig deklarieren möchten, können Sie mehrere Variablentypen deklarieren, indem Sie auf die ‘Add’-Schaltfläche klicken.

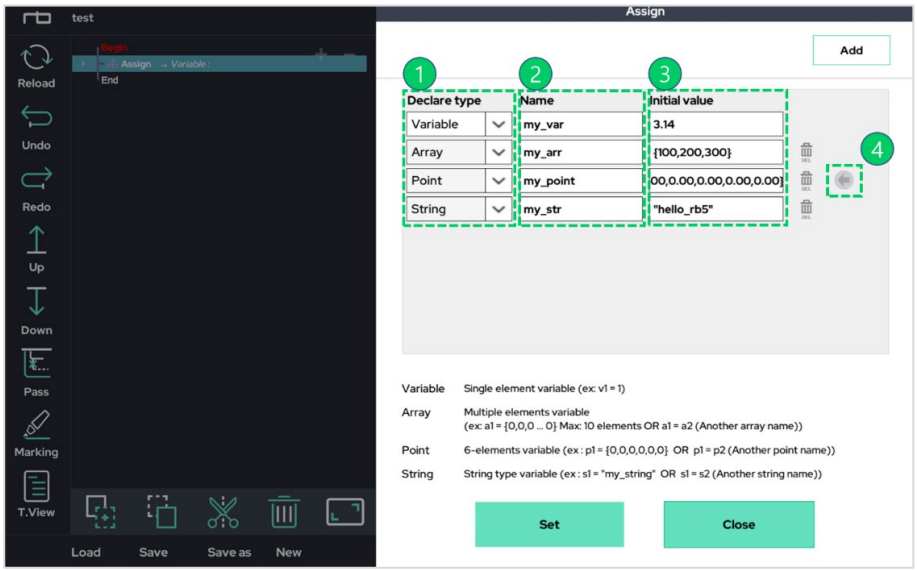
Wenn die Variablendeklaration abgeschlossen ist, wird die Anzahl der im Baum deklarierten Variablen wie unten angezeigt.



© RAINBOW ROBOTICS Inc. All rights reserved.

Wenn nur eine Deklaration vorgenommen wird, werden der Variablenname und der Anfangswert wie unten in der Baumstruktur angezeigt.

Der Aufbau des Popup-Fensters der ‘Assign‘-Funktion ist wie folgt.



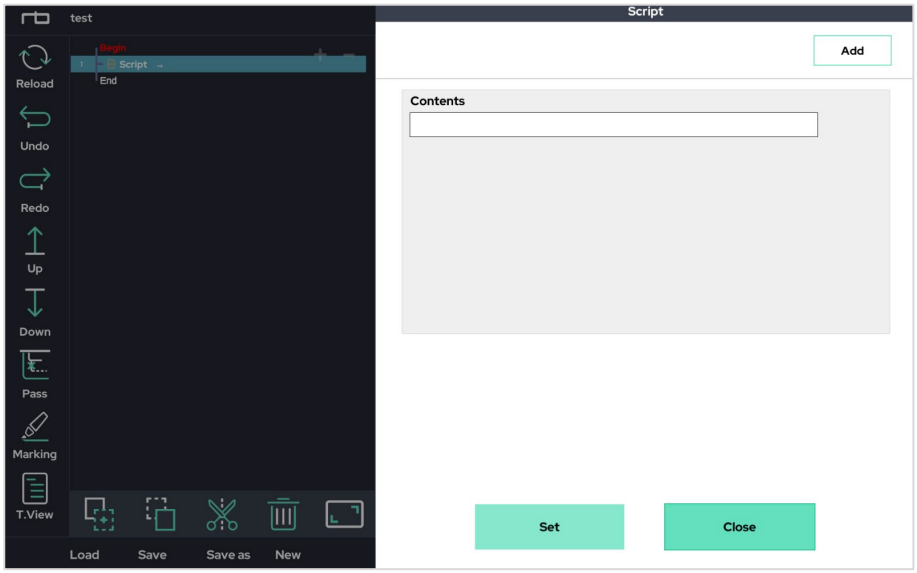
1) Wählen Sie den Variablentyp aus (einen von Variable-, Array-, Point-, String-Typ).
2) Benennen Sie die Variable.
3) Der Anfangswert bei der Deklaration wird festgelegt.
   - Beim Variable-Typ wird er Initialwert als einzelne Zahl eingegeben (Bsp.: 1).
   - Beim Array-Typ wird der Anfangswert mit geschweiften Klammern als Array-Typ eingefügt (Bsp. {100, 200, 300}).
   - Beim Point-Typ wird der Anfangswert mit geschweiften Klammern als Array-Typ mit 6 Längen eingefügt (Bsp.: {300, 300, 300, 0, 90, 0}).
   - Beim String-Typ wird der Anfangswert in Anführungszeichen als Zeichenfolge eingefügt (Bsp.: “hello_rb5”).
4) Diese Schaltfläche ist nur im Point-Typ verfügbar.
   - Die sechs Haltungswerte (x, y, z, Rx, Ry, Rz) des aktuellen Roboters werden automatisch in den Anfangswert gesetzt.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Skript(Script)-Funktion :

Script-Funktion, mit der Benutzer beliebige Skripte schreiben können. Es ist möglich, bestimmte Operation /Berechnungen separat von der Funktion zu implementieren oder Funktionen wie Variablensubstitution und -zuweisung unter Verwendung der Skriptfunktion zu implementieren.

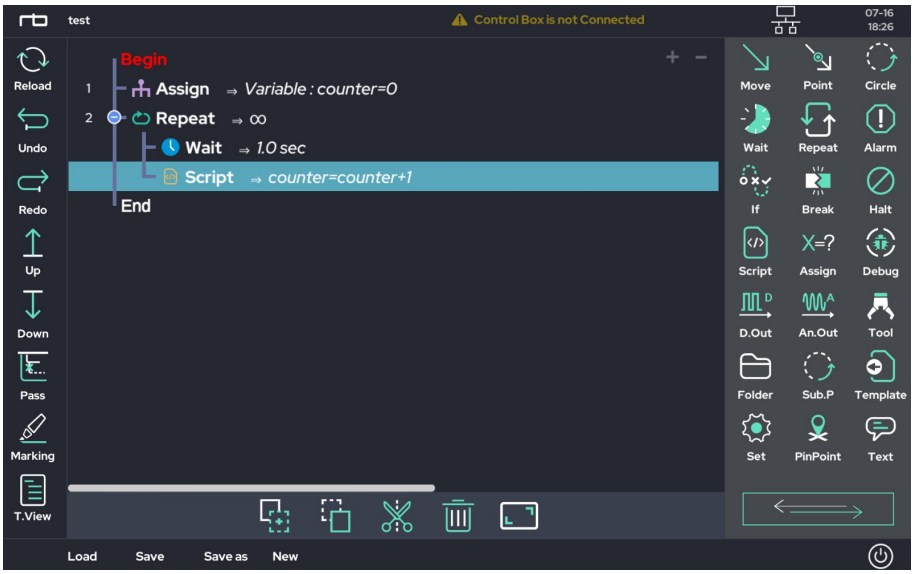
Fügen Sie die Script-Funktion zum Programmbaum hinzu und klicken Sie auf die hinzugefügte Script-Funktion, erscheint das folgende Popup-Fenster.



Hier können Sie ein beliebiges Skript frei eingeben. Über die Tastatur können verschiedene Funktionen wie Variablenbetrieb und Variablensubstitution frei eingegeben werden. Wenn Sie die Script-Funktion mehrerer Zeilen gleichzeitig verwenden möchten, können Sie dies über die ‚Add'-Schaltfläche am unteren Rand des Popup-Fensters vornehmen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Das folgende Beispiel ein Programm, das eine Repeat-Anweisung verwendet, um einmal pro Sekunde zu wiederholen. Nach jeder Seku8nde erhöht die ‚Script'-Funktion die Variable ‚***counter***' um 1. Dies wird kontinuierlich wiederholt, da die Anweisung so eingestellt ist, dass sie unendlich oft fortgesetzt wird.



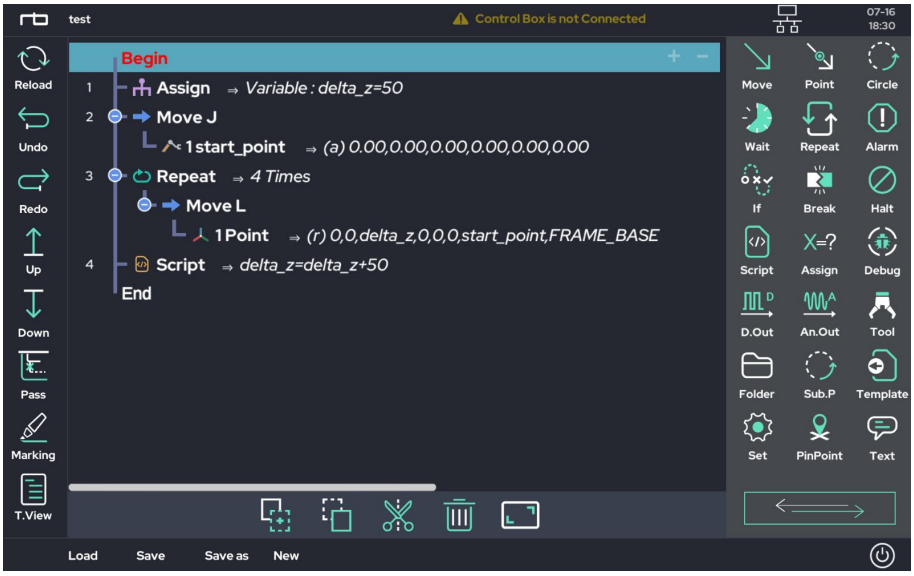
Im folgenden Beispiel unten wird der Roboter durch die MoveJ-Funktion zu einer bestimmten Haltung bewegt und die Haltung wird als 'start_point' benannt (unter Verwendung der Point-Benennungsfunktion).

Und mit einer Repeat-Anweisung (Repeat-Funktion) wird Bewegung zu MoveL viermal wiederholt.

- Dabei verwendet MoveL die ‚Relative Point'-Funktion, um sich in der z-Richtung um delta_z im Vergleich zum zuvor deklarierten ‚start_point' zu bewegen. (Siehe ‚relative point'-Funktion der ‚Linear Movement'-Serie der ‚Point'-Funktion)
- Am Ende der Repeat-Anweisung wird die Script-Funktion verwendet, um delta_z um 50 zu erhöhen. (delta_z wurde im Voraus durch die Assign-Funktion im entsprechenden Programm deklariert.)

Abschließend bewegt sich der Roboter mit MoveJ zur ersten Position, und speichert die Position als ‚start_point' und führt dann die Repeat-Funktion 4-mal aus und bewegt sich mit der MoveL-Funktion jede Repeat-Anweisung um 50 mm nach oben.

© RAINBOW ROBOTICS Inc. All rights reserved.



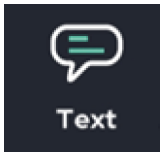
Vorsicht

Vorsicht:

1) Die Script-Funktion ist ein Bereich, in dem der Benutzer ein Skript freischreiben und ausführen kann.
2) Wenn Benutzer ein Skript schreibt, das nicht der Syntax entspricht, kann das Programm fehlerhaft funktionieren oder stoppen. Wenn sie diese Funktion verwenden, sollten sie die richtige Syntax verwenden.

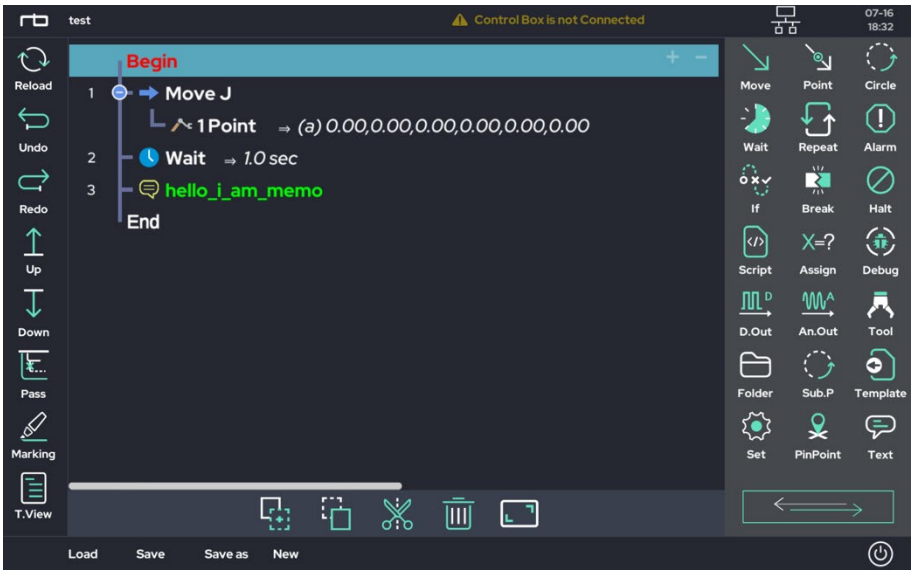
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Text-Funktion :

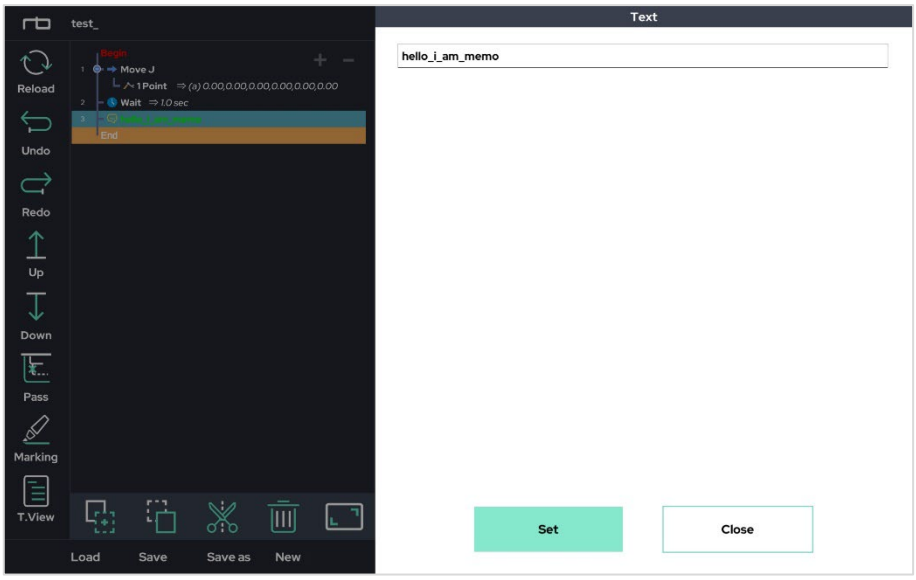


Sie können einfach Notizen zu notwendigen Teilen wie Kommentare im Programmlistenbaum machen.

Die Text-Funktion wird im Programmbaum grün dargestellt und beeinflusst die Programmfunktion in keiner Weise. Klicken Sie auf das Text-Symbol im Programmbaum, um die Text-Funktion einzufügen und sie wird wie unten gezeigt hinzugefügt.



Sie können eine gewünschte Notiz hier hinterlassen, indem sie hier auf Text-Schaltfläche klicken.

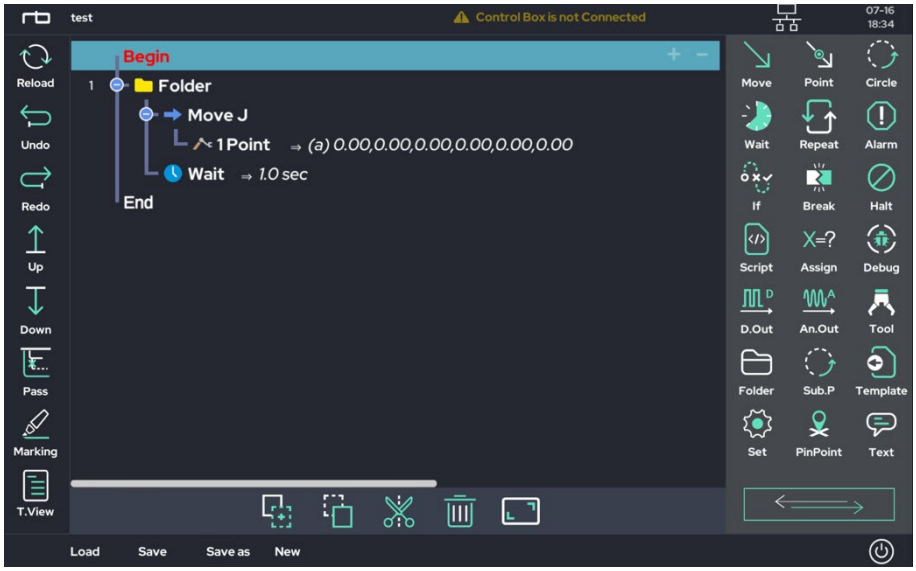


© RAINBOW ROBOTICS Inc. All rights reserved.

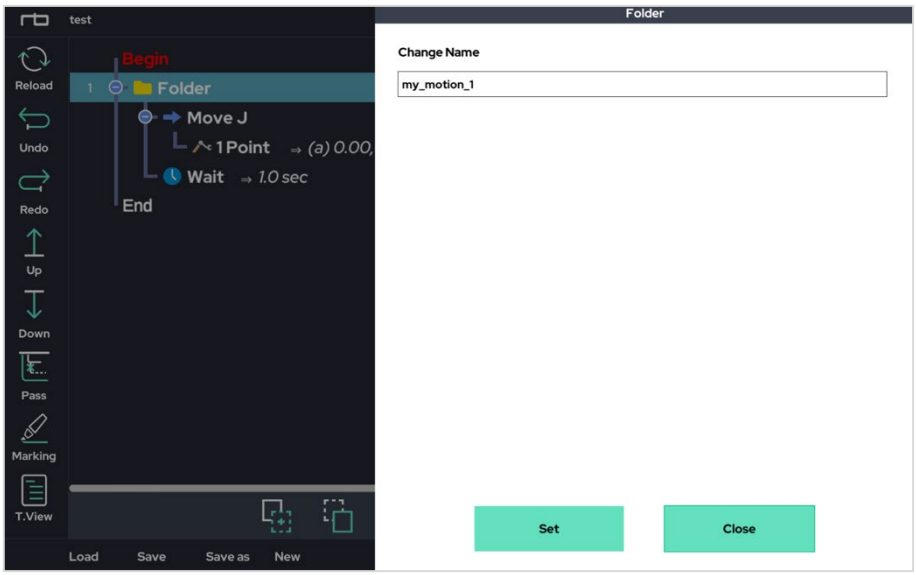
■ Ordner(Folder)-Funktion :

Die Ordnerfunktion, die dabei hilft, die geschriebenen Programmbefehle als Module zu verwalten, indem sie in Unterelementen des Ordners gruppiert werden. Es ist ein Werkzeug, das bei der Verwaltung hilft, indem es jeder Ordnerfunktion unterschiedliche Namen gibt.

Die Ordnerfunktion kann zum Programmbaum hinzugefügt werden, und verschiedene Funktionen könne wie unten gezeigt in Unterelemente eingefügt werden.

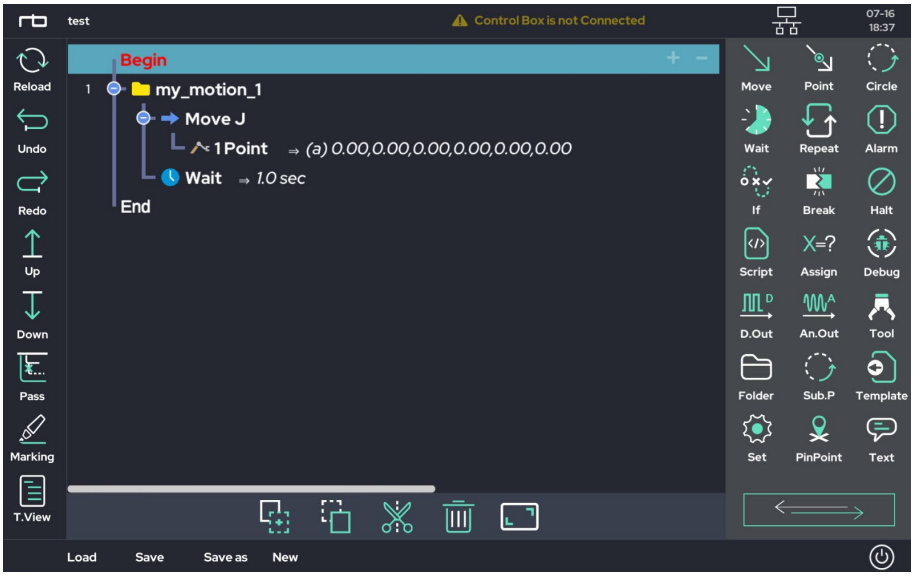


Wenn Sie den Ordner hier umbenennen möchten, klicken Sie auf das Ordnersymbol, um den Namen zu ändern.



© RAINBOW ROBOTICS Inc. All rights reserved.

Nach dem Umbenennen können Sie den Ordner mit dem geänderten Namen in der Baumstruktur sehen.



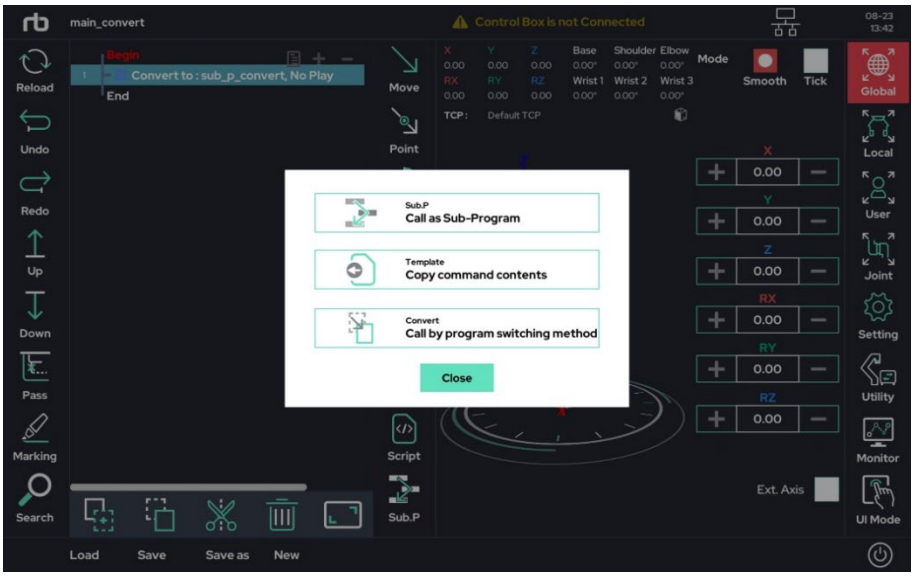
Wie bei der Text-Funktion hat dies keinen Einfluss auf die Funktionalität des Programms. Dies ist eine Funktion, die bei der modulweisen Verwaltung und der Gruppenverwaltung von Funktionen während des Unterrichts hilft.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Sub.P(Sub Program)-Funktion:

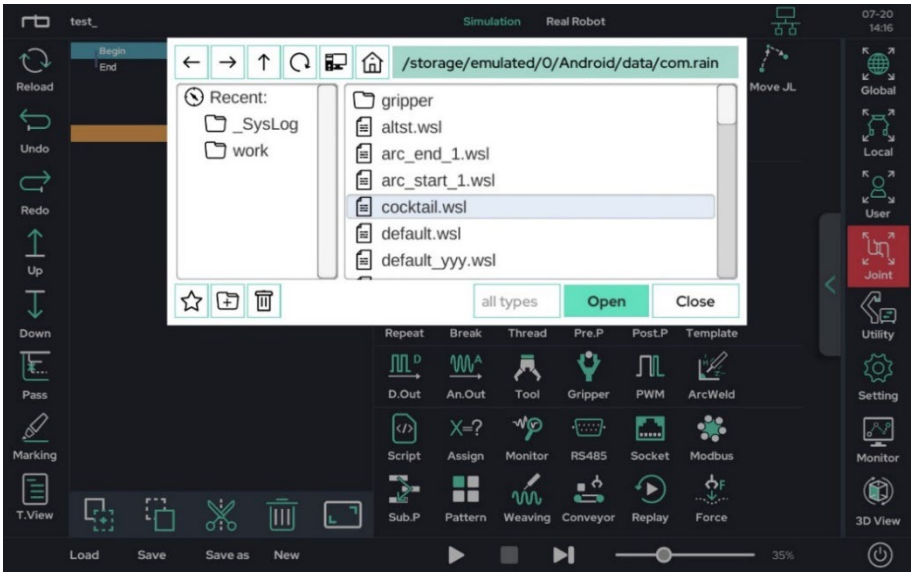
Mit dieser Funktion können sie andere vorgefertigte Programmdateien (Teaching File) in das aktuelle Projekt einfügen können (Unterprogrammfunktion).

Wenn Sie im Programm auf das 'Sub.P'-Symbol klicken, erscheint das folgende Popup-Fenster. Dann klicken sie auf ‚Sub.P'.



Wenn Sie auf ‚Sub.P' klicken, erscheint ein Popup-Fenster mit Datei-Explorer-Funktion wie unten gezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.



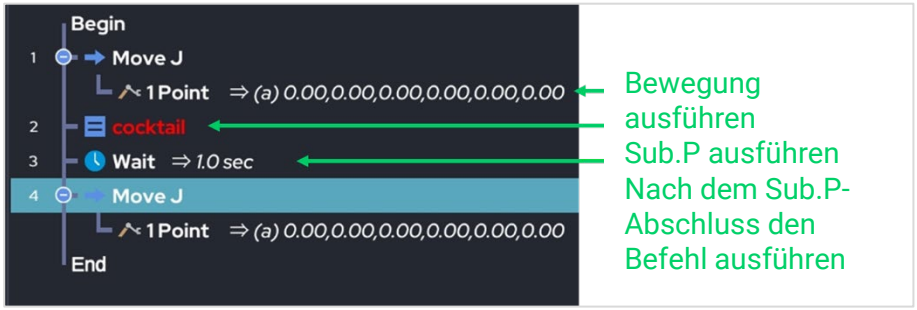
Hier sehen Sie andere Projekte, die auf dem Tablet-PC erstellt wurden. Wählen Sie hier das gewünschte Projekt aus und klicken Sie auf die Open-Schaltfläche.

Wie im obigen Bild zu sehen, wurde das zuvor erstellte Projekt in dieses Projekt eingefügt. Nachdem Sie den Programmbaum-Viewer in erweiterte Form erweitert haben, klicken sie auf das importierte Unterprojekt, um den Bildschirm zu teilen und den Inhalt des importierten Projektes zu überprüfen. Links wird das aktuelle Projekt und rechts werden die geladenen Projektinhalte angezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Durch die Sub.P-Funktion wird das Projekt wie andere Funktionen zeitlich sequenziell abgearbeitet.

Wenn andere Anweisungen nach der Sub.P-Funktion platziert werden, werden diese Anweisungen ausgeführt, nachdem die Ausführung der Sub.P beendet ist.



© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

Vorsicht:

1) Sie können den Inhalt des als Sub.P-Funktion bezeichneten Unterprojekts im aktuellen Projekt sehen, aber es ist nicht möglich, den Inhalt zu ändern. Sind Änderungen erforderlich, muss das jeweilige Projekt separate geöffnet werden.

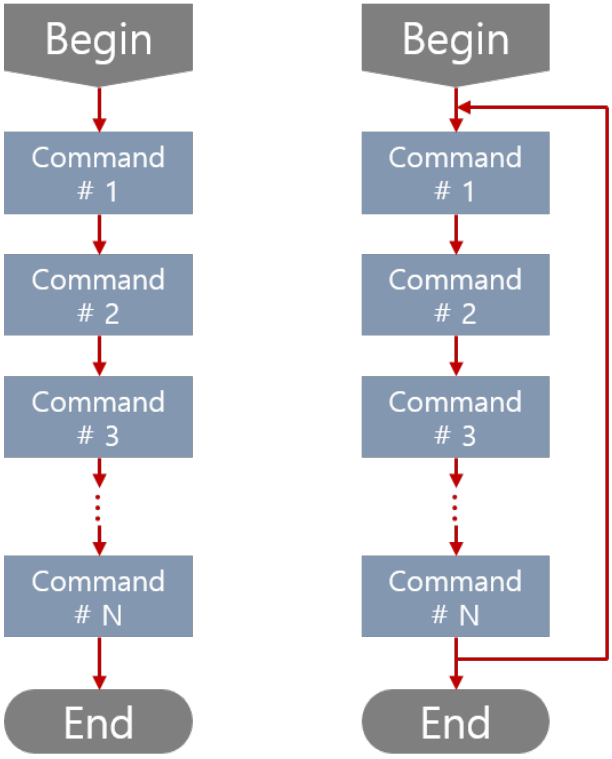
2) Die Sub.P-Funktion kann bis zu 10 Ebenen tief aufgerufen werden. Und es wird nicht empfohlen, den Teaching-Baum in einer rekursiven Aufrufmethode zu konfigurieren.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Pre.P(Pre Program)-Funktion:

Dies ist eine Pre-Program-Funktion(Vorprogrammfunktion), die den Inhalt des Untergeordneten nur einmal zum ersten Mal ausführt. Wie im Play-Modus werden beim Weiterlaufen des Programms die unter dem Pre-Program deklarierten Inhalte nur einmal zum ersten Mal ausgeführt. Bei Einzelbefehlen wie Variablendeklaration oder Kommunikationsverbindung können sie dies in der unteren Ebene des Vorprogramms verwalten.

Die folgende Abbildung zeigt den allgemeinen Ablauf der Befehlsausführung, wenn die Pre.P-funktion nicht verwendet wird.



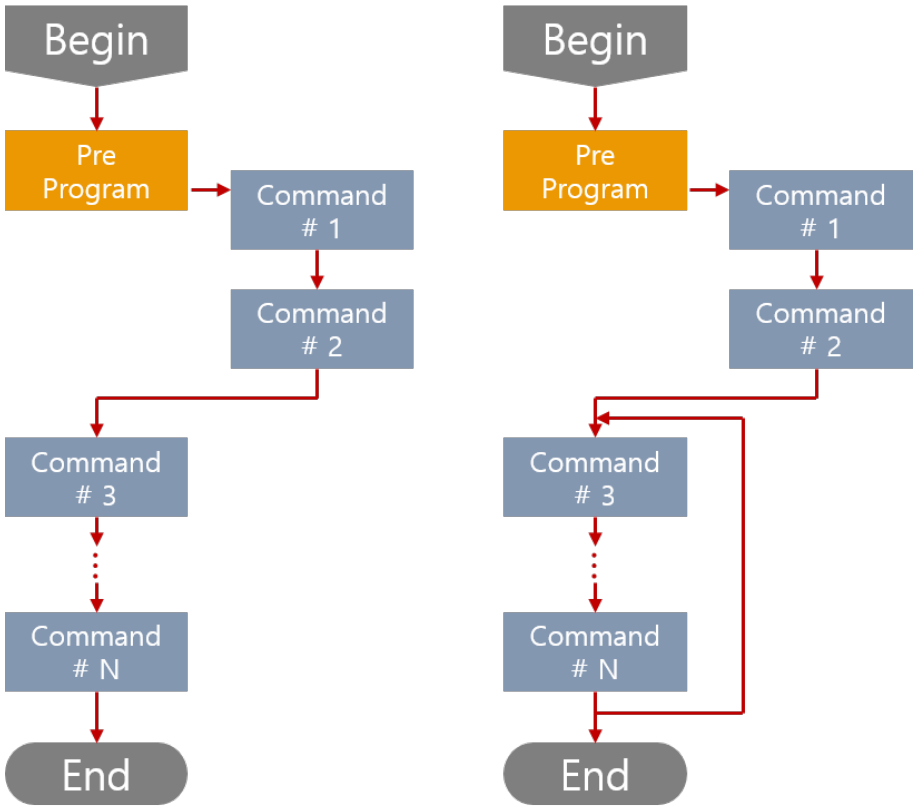
[Make-Bildschirm] [Play- Bildschirm]

Die linke Seite zeigt den Fall des Ausführens des Programms auf dem Make-Bildschirm, und die rechte Seite zeigt den Fall des Ausführens des Programms auf dem Play-

© RAINBOW ROBOTICS Inc. All rights reserved.

Bildschirm. In Make wird das Programm zwischen Begin und End einmal ausgeführt. Bei Play wird das Programm zwischen Begin und End wiederholt.

Die folgende Abbildung zeigt den Ablauf der Programmbefehlsausführung bei Verwendung der Pre.P-Funktion.



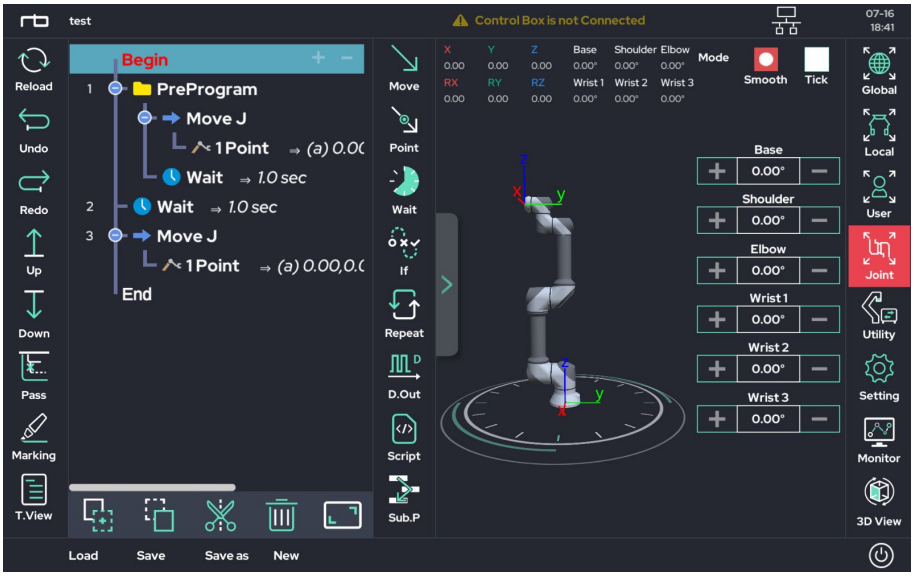
[Make-Bildschirm] [Play-Bildschrim]

Auf dem Make-Bildschirm werden Befehle zwischen Begin und End unabhängig von Pre.P sequenziell ausgeführt. Auf dem Play-Bildschirm wird das Programm zwischen Begin und End wiederholt, aber die unter Pre-Program(Pre.P) deklarierten Befehle werden nur einmal ausgeführt.

Funktionen und Abläufe, die nur einmal zum ersten Mal ausgeführt werden müssen, Variablendeklarationen und Kommunikationsverbindungen könne mit der Pre.P-Funktion konfiguriert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Die folgende Abbildung zeigt die in das aktuelle Projekt eingefügte Pre.P(Pre Program)-Funktion. Die Pre.P-Funktion wird aufgrund der Art der Funktion nur unter Begin eingefügt, und Pre.P kann nicht kopiert und an anderer Stelle eingefügt werden.



Vorsicht

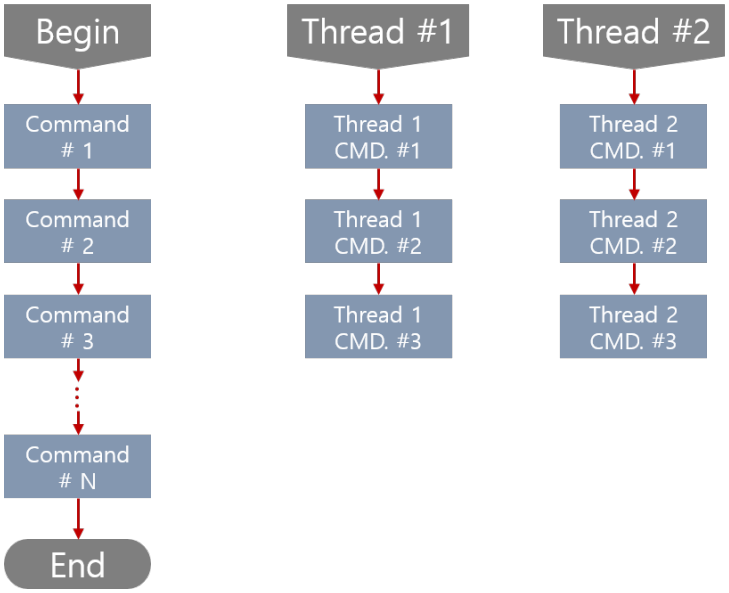
Vorsicht:

1) Wenn die Pre.P-Funktion in dem Projekt verwendet wird, das über die Sub.P-Funktion im Hauptprogramm aufgerufen wird, wird die Pre.P-Funktion nur auf das Hauptprogramm angewendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Thread-Funktion:

Die Unterelemente der Thread-Funktion laufen parallel zum Hauptprogramm. Befehle, die sich auf die Roboterbewegung beziehen, können jedoch nicht unter den Thread gestellt werden.



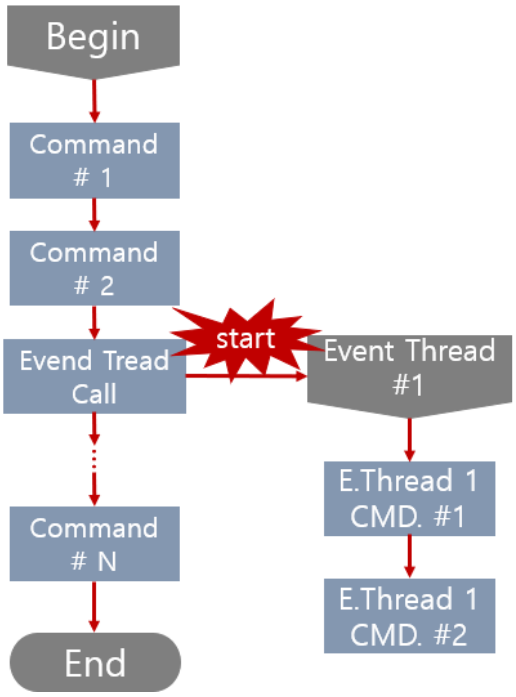
Wie in der obigen Abbildung dargestellt, wird die Thread-Funktion parallel zum Hauptprogramm konfiguriert.

- Thread-Funktion wird nicht automatisch wiederholt. Wenn Sie eine Repeat-Funktion implementieren möchten, müssen Sie die Repeat-Funktion unterhalb des Threads verwenden. Wenn Sie einen Thread implementieren möchten, der sich alle 1 Sekunde wiederholt, setzen Sie Repeat (infinity option) unter den Thread und Wait (1 Sekunde) unter Repeat.
- Bis zu 3 Threads werden unterstützt.
- Die Thread-Funktion funktioniert nur im Programm der obersten Ebene. Wenn ein durch die Sub.P-Funktion aufgerufenes Unterprogramm Threads verwendet, funktioniert es nicht normal.

© RAINBOW ROBOTICS Inc. All rights reserved.

Die Arten von Thread sind wie folgt.

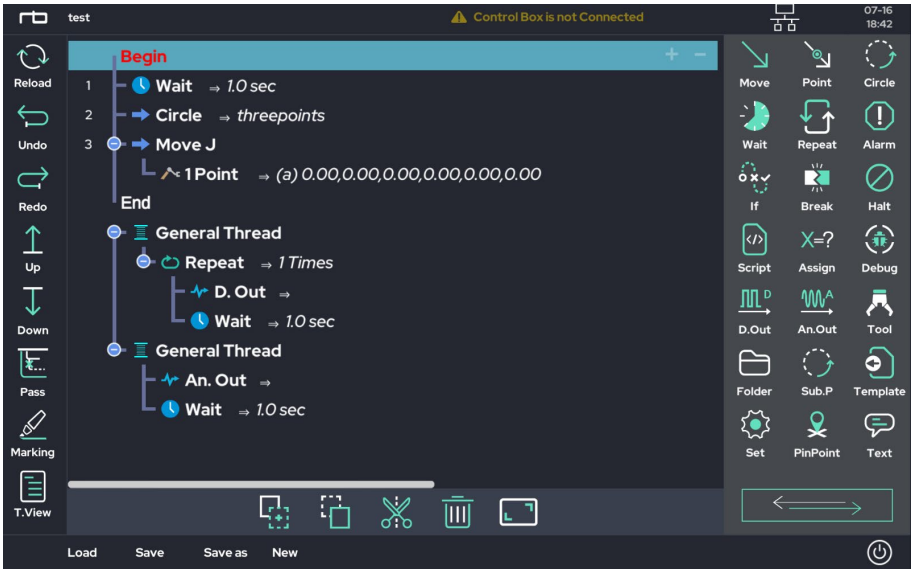
- Normaler Thread: bei der absichtlichen Pause, dem Alarm, der Kollisionserkennung usw. wird zusammen gestoppt.
- Non-Stop-Thread: außer bei der Kollisionserkennung wird das nicht gestoppt.
- Non-Stop-Thread 2 : Bis das Programm beendet ist, wird das nicht gestoppt.
- Allgemeiner Event-Thread: Ein allgemeiner Thread, der von der Event-Thread-Aufruffunktion im Hauptprogramm ausgeführt wird.
- Non-Stop-Event-Thread: Non-Stop-Thread, der von der Event-Thread-Aufruffunktion im Hauptprogramm ausgeführt wird.



Wie in der obigen Abbildung gezeigt, beginnt der Event-Thread zu laufen, wenn die Event-Aufruffunktion im Hauptprogramm verwendet werden.

Die folgende Abbildung zeigt, wie die Thread-Funktion in das eigentliche Projekt eingefügt wird. Im folgenden Beispiel werden zwei Threads eingefügt.

© RAINBOW ROBOTICS Inc. All rights reserved.



Vorsicht

Vorsicht:

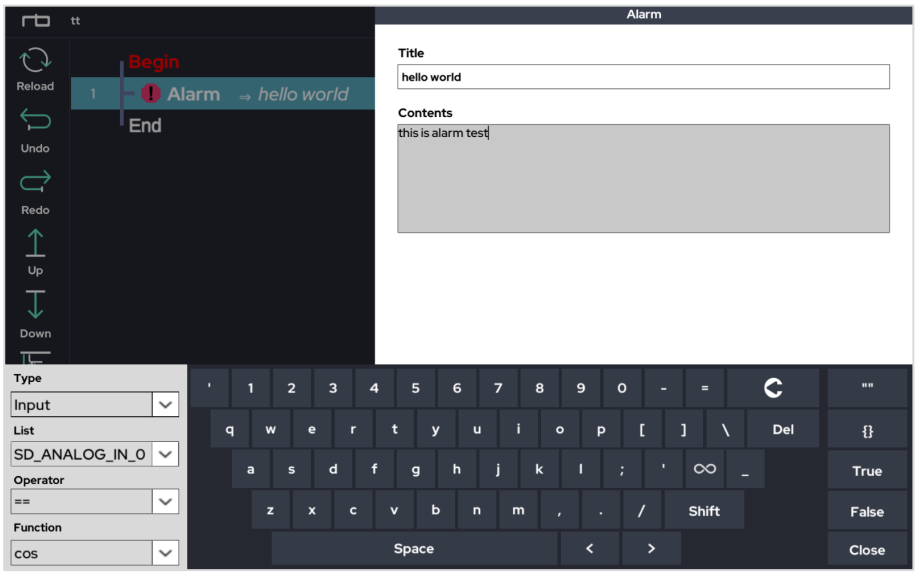
1) Aus Gründen der Programmstabilität wird es nicht empfohlen, Threads in Programmen mit dem Namen Sub.P außer dem Hauptprojekt zu verwenden.
2) Befehle, die sich auf die Roboterbewegung beziehen, wie Move oder Circle, können nicht unter den Thread gestellt werden.
3) Bei Verwendung Pause- oder Alarmfunktion werden sowohl das Hauptprogramm als auch der Thread angehalten. Wenn das Hauptprogramm beendet wird, wird auch der Thread beendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Alarm-Funktion:

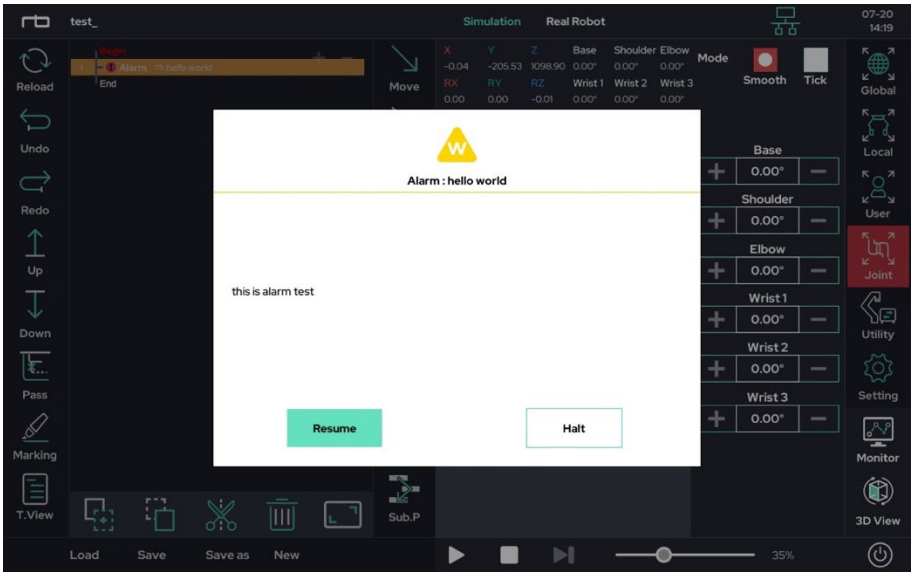
Mit der Funktion wird eine Alarm-Popup-Meldung angezeigt. Sie können Benutzer eine Bestätigung anfordern, indem Sie eine Alarmmeldung in einem bestimmten Zustand oder Abschnitt platzieren.

Nachdem Sie die Alarmfunktion in das Programm eingefügt und angeklickt haben, erscheint folgendes Einstellungsfenster.



Der Benutzer kann Titel und Inhalt des Alarmfensters direkt eingeben. Wenn Sie das Programm nach der obigen Einstellung ausführen, erscheint das folgende Popup-Fenster, sobald der Alarmbefehl ausgeführt wird.

© RAINBOW ROBOTICS Inc. All rights reserved.



- Resume (Fortsetzen): Mit dem nächsten Befehl fortfahren.
- Halt (Halten): An dieser Stelle Programm beenden.

Der Programmablauf wird angehalten, während die Alarmfunktion ausgeführt wird. Wenn Sie im Popup-Fenster auf die Resume (Fortsetzen)-Schaltfläche klicken, wird der nächste Befehl ausgeführt. Wenn Sie die Halt(Halten)-Schaltfläche klicken, wird der Programmablauf an dieser Stelle angehalten.

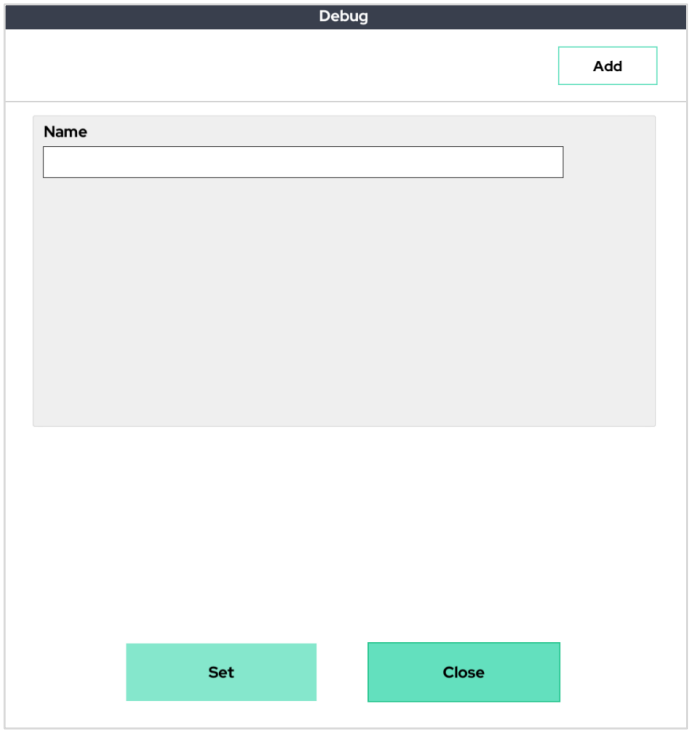
Wenn die Alarmfunktion verwendet wird, wird das Hauptprogramm angehalten, und der im Thread verwendete Ablauf wird ebenfalls gleichzeitig angehalten.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Debug-Funktion:

Debug-Funktion zum Debuggen interner Werte. Wenn Sie einen gewünschten Variablenwert oder internen Parameter anfordern, können Sie den Wert in einem Popup-Format überprüfen.

Die Debugging-Funktion ist eine Funktion zum Beobachten interner Variablen. Es wird hauptsächlich verwendet, um den Wert der Variablen zu überprüfen, die im Programm während des Programmierens / der Entwicklung des Programms verwendet werden.

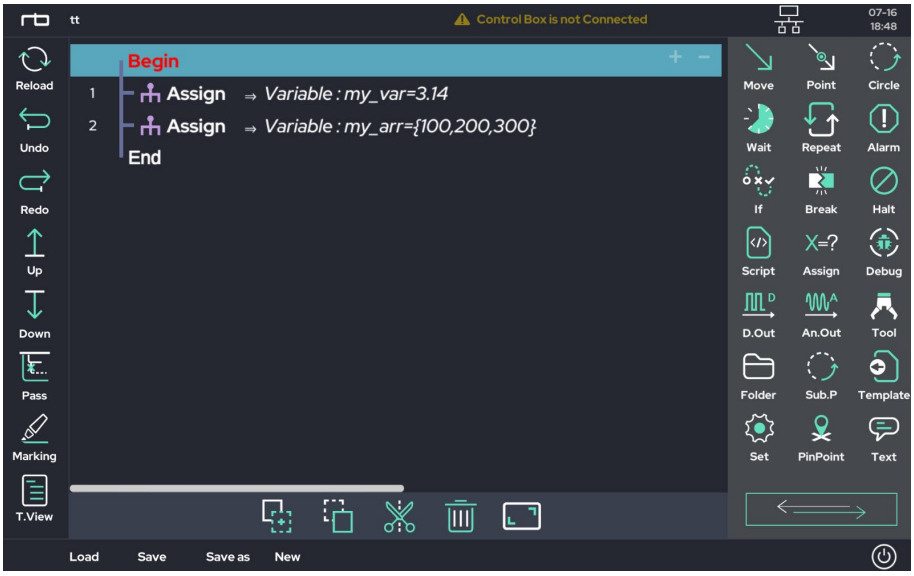


Nachdem Sie die Debug-Funktion zum Programmbaum hinzugefügt haben, klicken Sie auf Debug. Dann ist das obige Popup-Fenster anzuzeigen. Geben Sie einfach den gewünschten Variablennamen in das Name-Feld ein. Wenn Sie mehrere Variablen gleichzeitig beobachten möchten, klicken Sie auf die Add-Schaltfläche, um Elemente hinzuzufügen.

Das folgende Beispiel zeigt die Debug-Funktion.

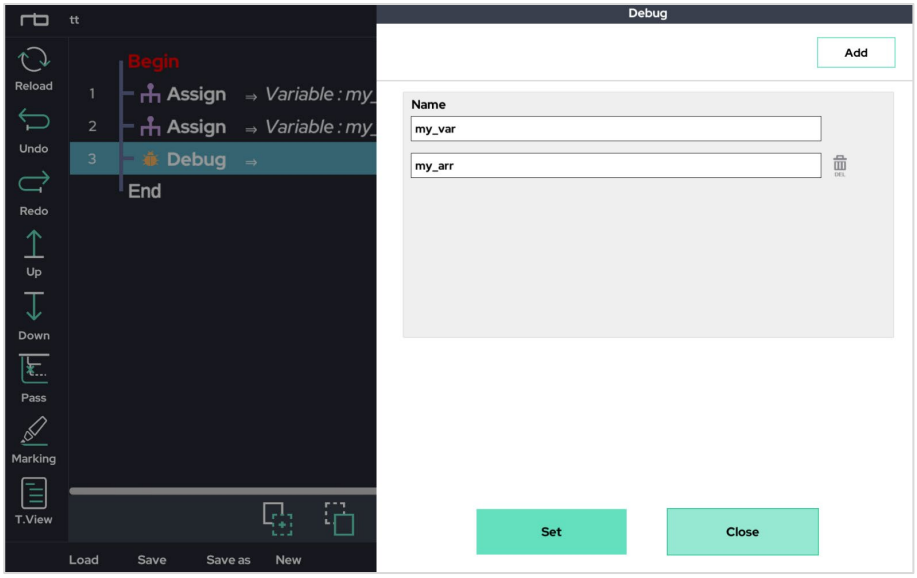
Deklarieren Sie eine Variable vom Variable-Typ (my_var = 3,14) und eine Variable von Array-Typ (my_arr = {100,200,300}) mit der Assign-Funktion wie unten gezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.



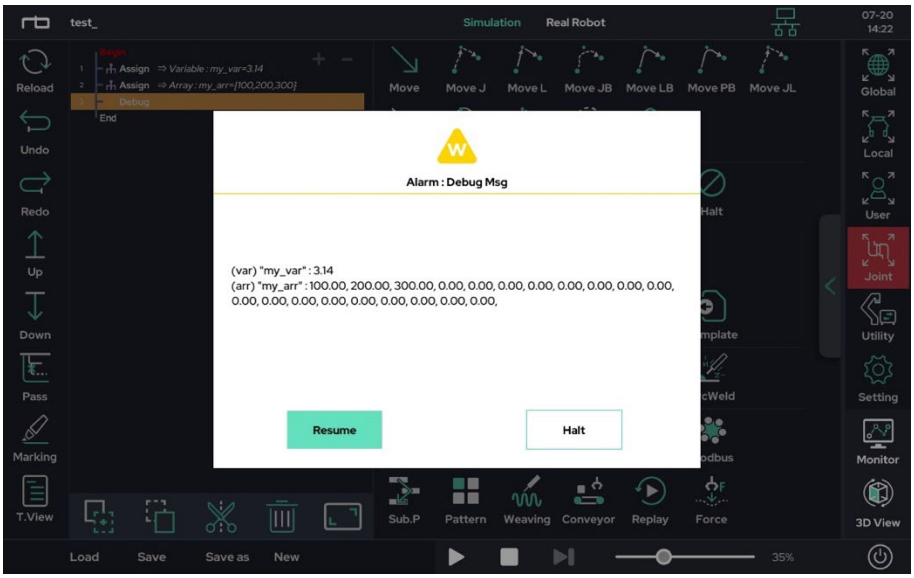
Fügen Sie darunter die Debug-Funktion hinzu.

Legen Sie im Debug-Eigenschaftenfenster fest, dass die beiden zuvor deklarierten Variablen wie unten gezeigt beobachtet werden.



© RAINBOW ROBOTICS Inc. All rights reserved.

Wenn Sie das Programm nach Abschluss der Einstellung ausführen (Tablet-PC und Steuerbox müssen vor der Ausführung verbunden sein), erscheint das folgende Popup-Fenster, sobald der Debug-Befehl ausgeführt wird. Dann können Sie die angegebenen Variablenwerte beobachten.



- Resume (Fortsetzen): Mit dem nächsten Befehl fortfahren.
- Halt (Halten): An dieser Stelle das Programm beenden.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Set-Funktion:

Verschiedene im Setup-Menü einstellbare Einstellparameter können innerhalb des Programms separate eingestellt werden. Das hat die Funktion, bestimmte Einstellparameter während der Programmausführung zu ändern.

Die verschiedenen im Setup-Einstellungsbildschirm geänderten Parameter werden als Standardwerte auf alle Projekte angewendet, die das entsprechende Steuerbox verwenden. Wenn Sie einen bestimmten Parameter separate für ein bestimmtes Projekt verwenden müssen, können Sie die Parametereinstellung für jedes Projekt verwalten, indem Sie die Set-Funktion oben im Projekt hinzufügen (z. B.: unter Pre.P).

Die Set-Funktion ist keine permanente, sondern eine temporäre Einstellung. Wenn für dieselbe Parametereinstellung eine neue Set-Funktion aufgerufen wird, werden die Parameter basierend auf der neuen Set-Funktion wiedergegeben.

Wenn das Programm endet, kehren die Parametereinstellungen zu dem im Setup festgelegten Standardwerten zurück.

Die von der Set-Funktion bereitgestellten Unterfunktionen sind wie folgt.

- Timer zurücksetzen
- Empfindlichkeit der Kollisionserkennung einstellen
- TCP-Last einstellen
- Bewegungsoffset
- Inbox-Funktion einstellen
- Temporärer TCP-Standort einstellen
- Werkzeugsicheren Bereich einstellen
- Betriebssicheren Bereich einstellen
- Inbox-Bereich einstellen
- Ein/Aus von Kollisionserkennung
- Geschwindigkeitsmultiplikator pauschal einstellen
- Beschleunigungsmultiplikator pauschal einstellen
- Serielle Kommunikation einstellen
- Betriebsgeschwindigkeit direkt einstellen
- Spiralkreisbewegung einstellen
- Geschwindigkeitsleiste kontrollieren

© RAINBOW ROBOTICS Inc. All rights reserved.

- Stoppmodus nach Kollisionserkennung
- Benutzerkoordinatensystem verschieben
- Programmablauf nach Kollisionserkennung
- Temporäre Steuerung von Steuerboxausgang
- XYZ-Positionswert beim Betrieb der L-Serie fixieren
- Rotationwerte beim Betrieb der L-Serie fixieren
- Benutzerkoordinatensystem temporär einstellen
- Positionswert des Zielpunkts verschieben
- Positionswert des Zielpunkts verschieben 2
- Ein/Aus von Vibrationserkennungssensor
- Simulation von digitalen Eingangssignalen
- Programmfluss kontrollieren
- Beschleunigungsmodus mit Hochgeschwindigkeit
- Betriebszeit beschränken
- Hochempfindliche Kollisionserkennung
- Offset-Feinsteuerung
- Benutzerkoordinatensystem verschieben 6D
- Benutzerkoordinatensystem automatisch ausrichten
- Timer einstellen
- No-Arc-Bewegungsgeschwindigkeit

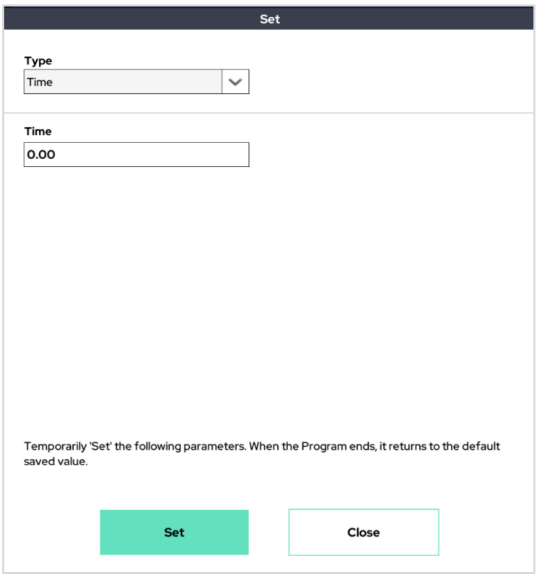
Vorsicht

Vorsicht:

1) Der in der Set-Funktion eingestellte Wert ist ein temporärer Wert. Wenn das Programm beendet wird, kehrt es automatisch zu den im Setup festgelegten Standardwerten zurück.
2) In der Set-Funktion bereitgestellte Funktionen können mitten im Programmablauf auf andere Werte geändert werden. Wenn Sie beispielsweise die Funktion „Collision On/Off“ von Set verwenden, können Sie die Kollisionserkennung mitten im Programmablauf selektiv ausschalten.

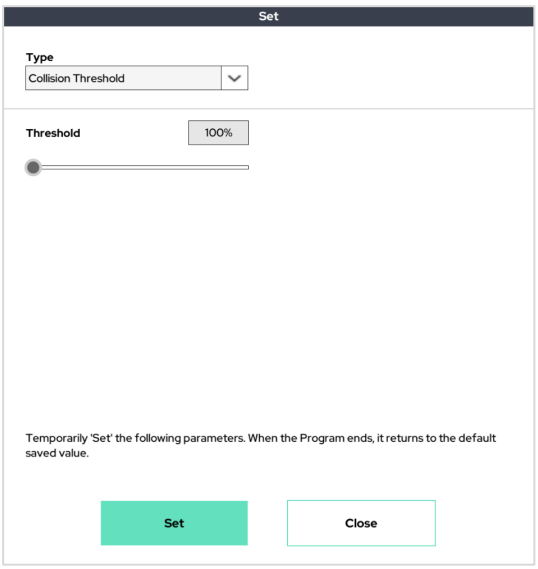
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Time (Timer zurücksetzen)



Der Timer startet und der anfängliche Startwert wird eingestellt. Der Timerwert erhöht sich vom eingegebenen Wert.

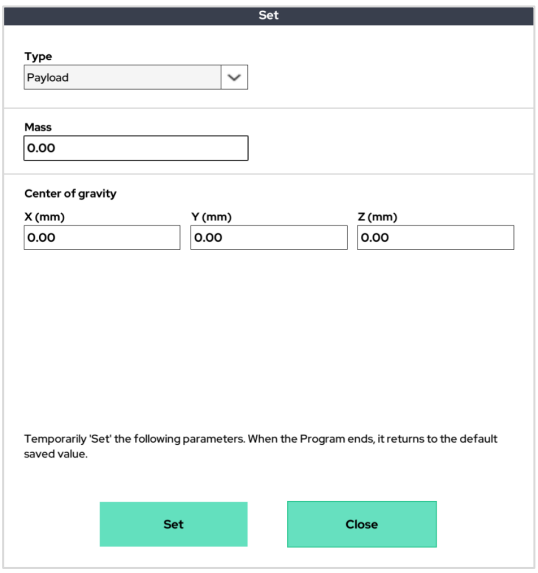
Set-Funktion: Collision Threshold (Empfindlichkeit der Kollisionserkennung einstellen)



Die Empfindlichkeit der Kollisionserkennung wird vorübergehend eingestellt. Je niedriger der Wert ist, desto empfindlicher ist die Erkennung eingestellt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Cobot.

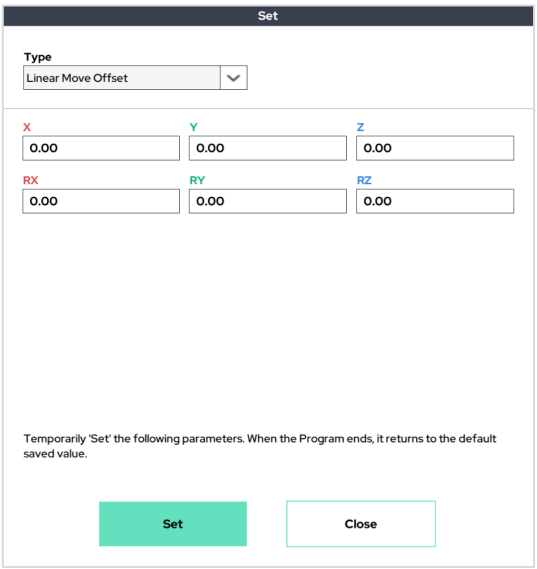
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Tool Payload (TCP-Last einstellen)



Das Gewicht und der Schwerpunkt des Werkzeugs werden vorübergehend festgelegt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Tool.

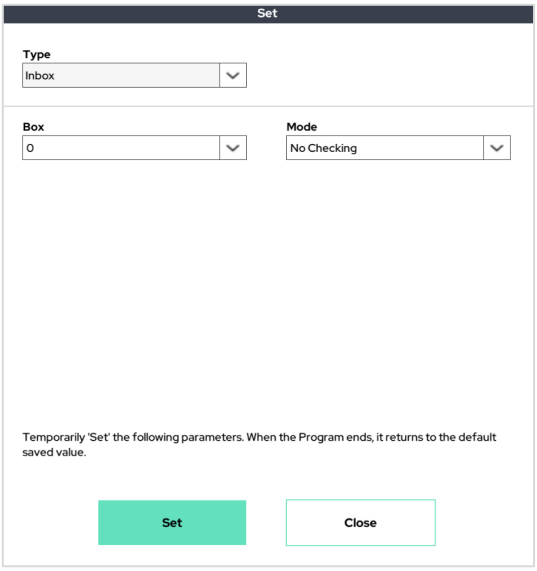
Set-Funktion: Linear Move Offset (Bewegungsoffset)



Sie können einen gewissen Versatz (Offset) zum Basiskoordinatensystem geben. Mit dieser Funktion ist die temporäre Einstellung von Offsets bis zu 20 mm möglich.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Inbox (Inbox-Funktion einstellen)

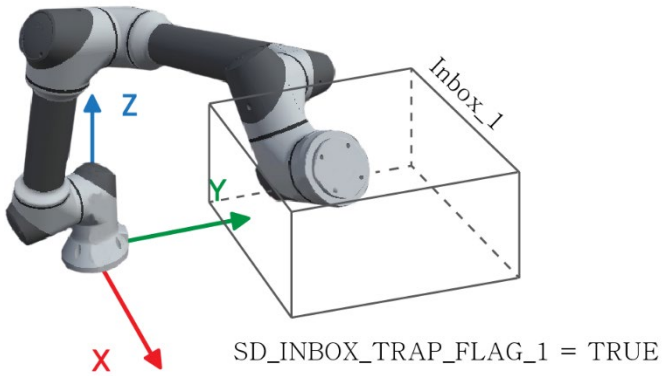


Optionen für die Inbox-Funktion können eingestellt werden. Mit der Inbox-Funktion wird automatisch überprüft, ob sich ein bestimmter Teil des Roboters in einem bestimmten Bereich befindet (Worldzone Trap-Funktion). Folgende Bereiche könne überprüft werden.

Überprüfen Sie, ob sich die Mitte des Werkzeugflansches am Ende des Roboterarms in einem bestimmten Bereich befindet.

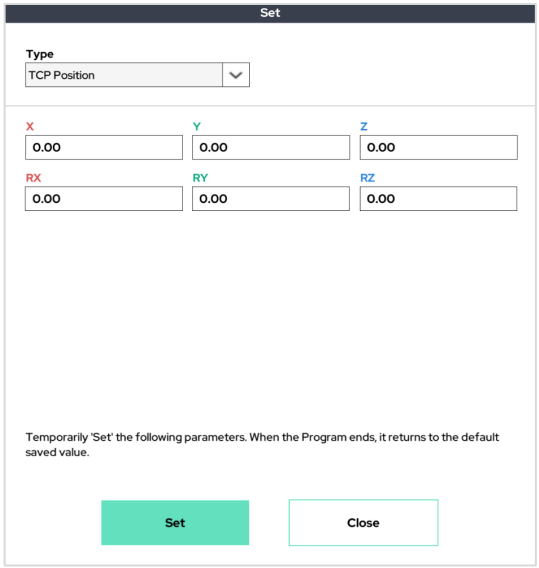
- Überprüfen Sie, ob der am Ende des Roboterarms eingestellte TCP in einen bestimmten Bereich eingetreten ist.
- Überprüfen Sie, ob das virtuelle Box am Ende des Roboterarms in einen bestimmten Bereich eingetreten ist.
- Überprüfen Sie, ob einer der oben genannten Fälle in einen bestimmten Bereich eingetreten ist.

Die Größe und Position von Inbox eines bestimmten Bereichs kann im Setup-Inbox eingestellt werden. Ob es sich in einem bestimmten Bereich befindet oder nicht, kann basierend auf der obigen Einstellung durch SD_INBOX_TRAP_FLAG_0 oder SD_INBOX_TRAP_FLAG_1 unter gemeinsam genutzten Datentypen verwendet werden.



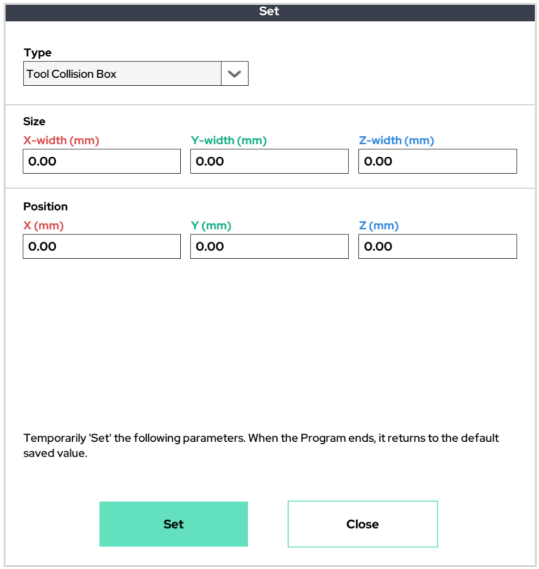
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: TCP Position (Temporärer TCP-Standort einstellen)



Die TCP-Position des Werkzeugs und das Werkzeugkoordinatensystem werden vorübergehend festgelegt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Tool.

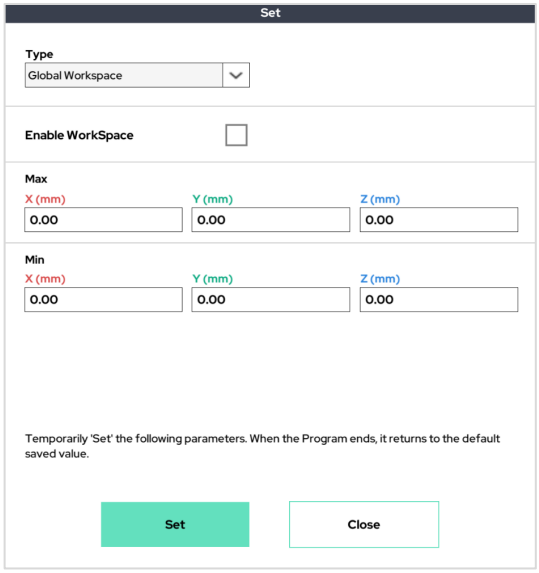
Set-Funktion: Tool Collision Box (Werkzeugsicheren Bereich einstellen)



Die Größe und Position der virtuellen Werkzeugbox zur Selbstkollisionsvermeidung vorübergehend festgelegt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Tool.

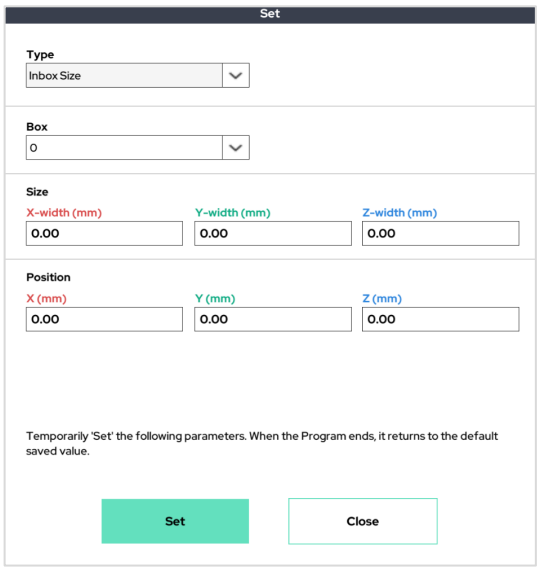
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Global Workspace (Betriebssicheren Bereich einstellen)



Die Kategorie des Betriebsbereichs (Workspace) für die Selbstkollisionsvermeidung wird festgelegt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Cobot.

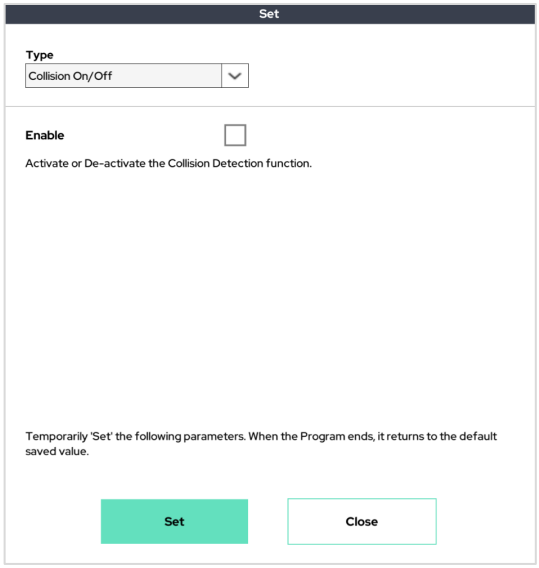
Set-Funktion: Inbox Size (Inbox-Bereich einstellen)



Die Position und Größe von Inbox werden festgelegt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Inbox.

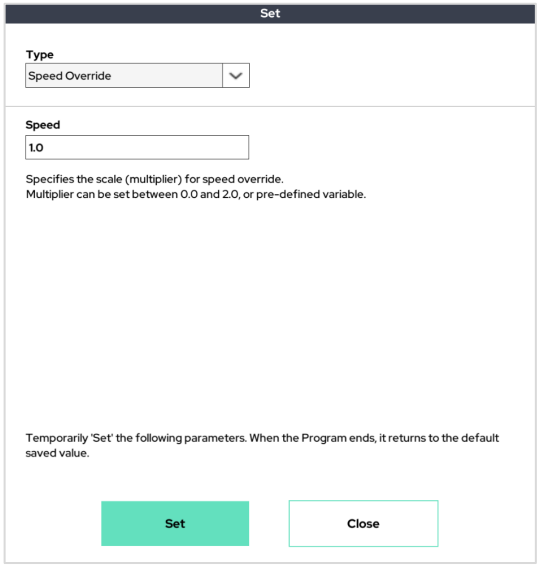
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Collision On/Off (Ein/Aus von Kollisionserkennung)



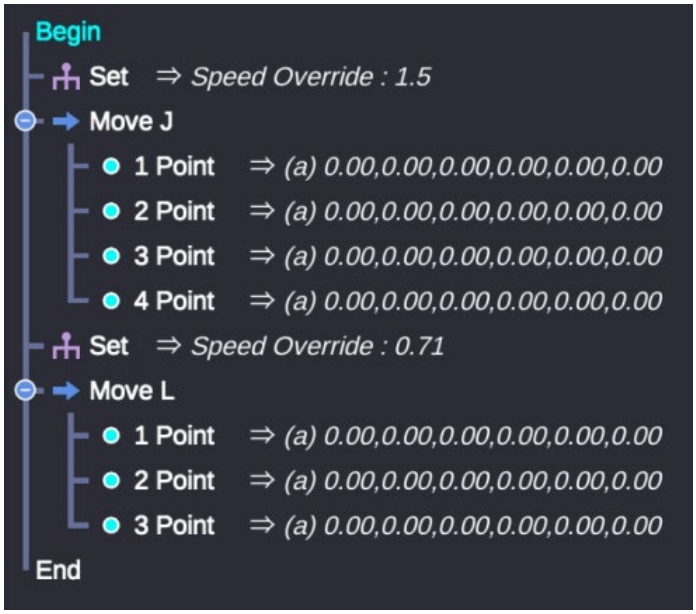
Ob der externe Kollisionserkennungsmodus verwendet werden soll, wird vorübergehend festgelegt. Es hat dieselbe physikalische Bedeutung wie die Einstellung in Setup-Cobot.

Set-Funktion: Speed Override (Geschwindigkeitsmultiplikator pauschal einstellen)



Bei Teaching werden die durch die Move-Funktion und Point-Funktion eingestellten Geschwindigkeitswerte gleichzeitig geändert. Sie können einen Wert zwischen 0 und 2,0 eingeben und auch parametrisierte Informationen hinzufügen.

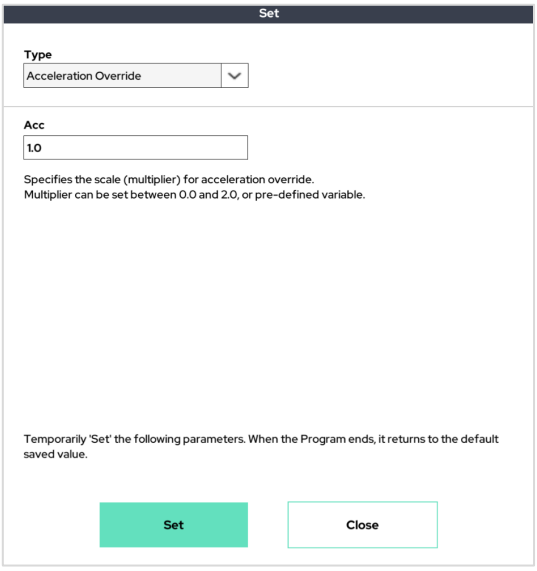
© RAINBOW ROBOTICS Inc. All rights reserved.



Im obigen Beispiel wird Set-Speed-Override vor MoveJ als 1,5 eingestellt. Daher wird bei der MoveJ-Bewegung das 1,5-fach der eingestellten Geschwindigkeit mit der Gleichen Wirkung wie die eingestellte Geschwindigkeit. Zwischen MoveJ und MoveL wird Set-Speed-Override als 0,71 verwendet. Daher wird bei der MoveL-Bewegung das 0,71-fach der eingestellten Geschwindigkeit mit der Gleichen Wirkung wie die eingestellte Geschwindigkeit.

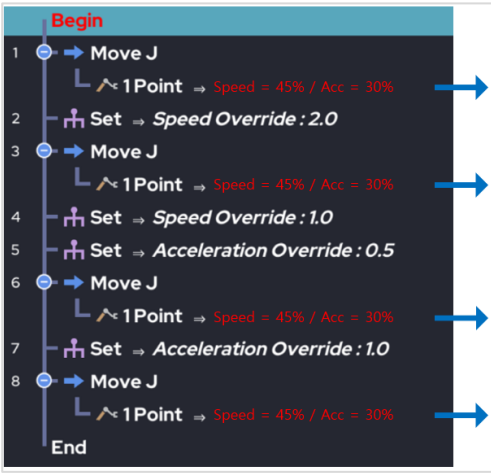
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Acceleration Override (Beschleunigungsmultiplikator pauschal einstellen)



Bei Teaching werden die durch die Move-Funktion und Point-Funktion eingestellten Geschwindigkeitswerte gleichzeitig geändert. Sie können einen Wert zwischen 0 und 2,0 eingeben und auch parametrisierte Informationen hinzufügen.

Durch den folgenden Code können Sie sehen, wie sich Geschwindigkeit und Beschleunigung ändern, wenn Speed Override und Acceleration Override verwendet werden.



Der Roboter bewegt sich mit der folgenden Geschwindigkeit und Beschleunigung. = 45%/30%

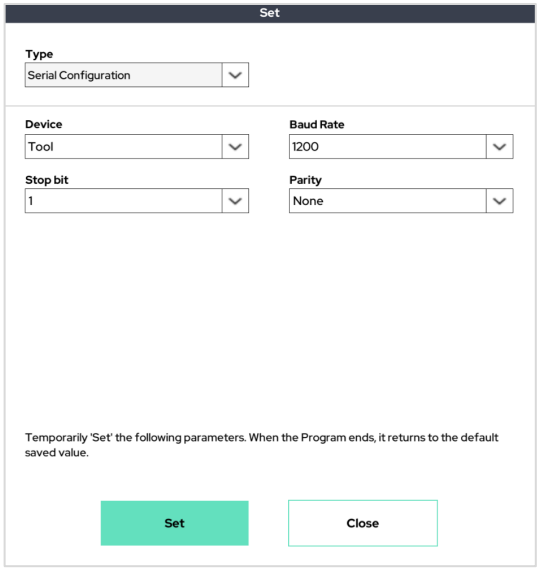
Der Roboter bewegt sich mit der folgenden Geschwindigkeit und Beschleunigung. = 90%/30%

Der Roboter bewegt sich mit der folgenden Geschwindigkeit und Beschleunigung. = 45%/15%

Der Roboter bewegt sich mit der folgenden Geschwindigkeit und Beschleunigung. = 45%/30%

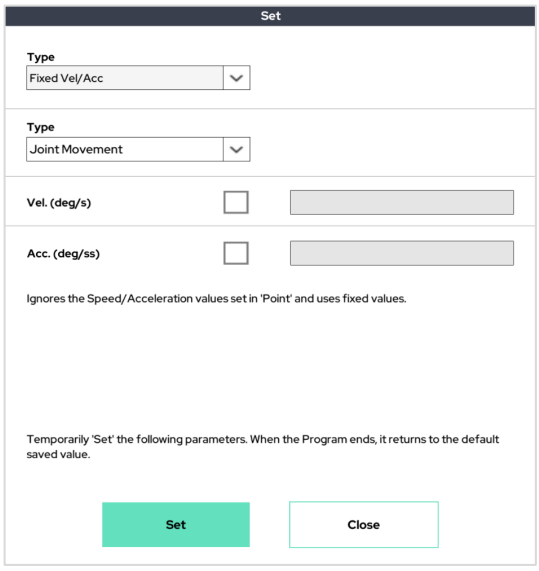
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Serial Configuration (Serielle Kommunikation einstellen)



Die Baudrate und das Protokoll (Stop bit / Parity) der seriellen Kommunikation werden vorübergehend eingestellt. Es hat dieselbe Bedeutung wie die Einstellung in Setup-Serial.

Set-Funktion: Fixed Vel/Acc (Betriebsgeschwindigkeit direkt einstellen)



Wenn Sie einen festen Wert verwenden möchten, können Sie mit dieser Funktion die für jeden Bewegungspunkt eingestellte Geschwindigkeit/ Beschleunigung ignorieren. Es gibt zwei Unteroptionen: Joint Movement und Linear Movement.

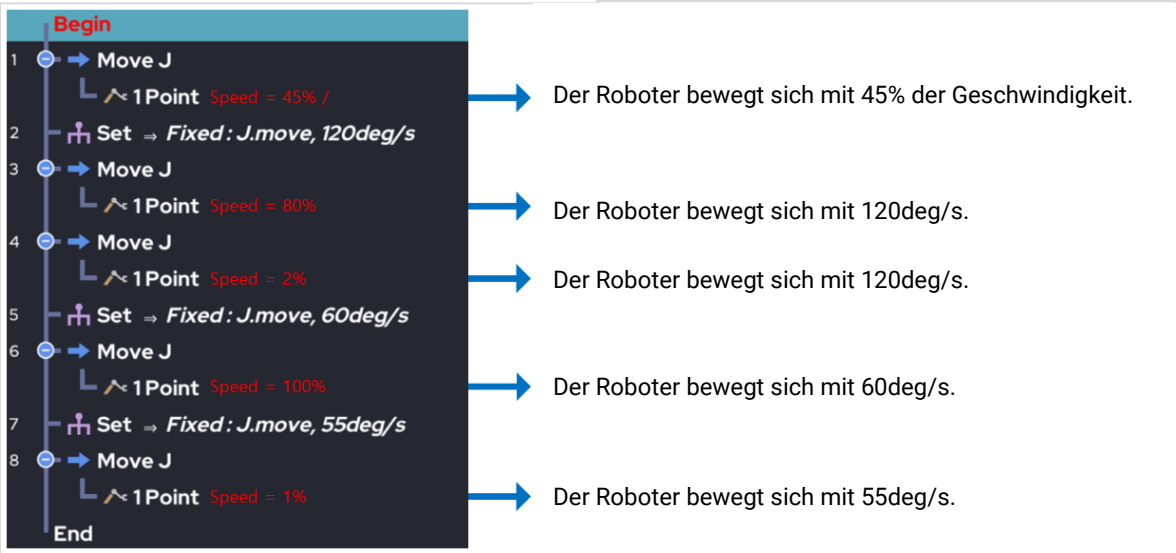
Die in Joint Movement eingestellte Geschwindigkeit (deg/s) und Beschleunigung (deg/s^2) wirken sich auf die Bewegungsgeschwindigkeit und Bewegungsbeschleunigung der Gelenkbewegungstypen ‚MoveJ' und ‚MoveJB' aus.

© RAINBOW ROBOTICS Inc. All rights reserved.

Die in Linear Movement eingestellte Geschwindigkeit (mm/s) und Beschleunigung (mm/s^2) wirken sich auf die Bewegungsgeschwindigkeit und Beschleunigung von ‚Linear movement type' MoveL, MoveLB, MovePB, MoveJL, MoveITPL und Circle aus.

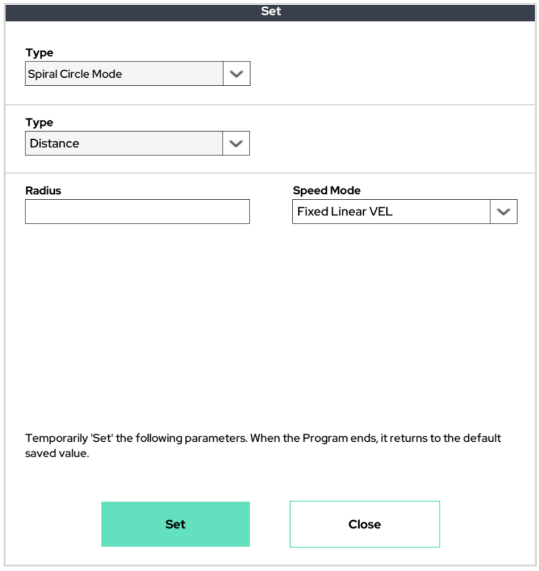
Wenn Sie die Geschwindigkeit/Beschleunigung nicht durch diese Funktion erzwingen möchten, können Sie die Checkbox deaktivieren. In diesem Fall wird während des Betriebs der für jeden Punkt eingestellte Geschwindigkeits-/Beschleunigungswert befolgt.

Bsp.) Wenn Sie während des Betriebs eine bestimmte Geschwindigkeit und Beschleunigung beibehalten müssen, können Sie diese Set-Funktion wie im folgenden Code verwenden.



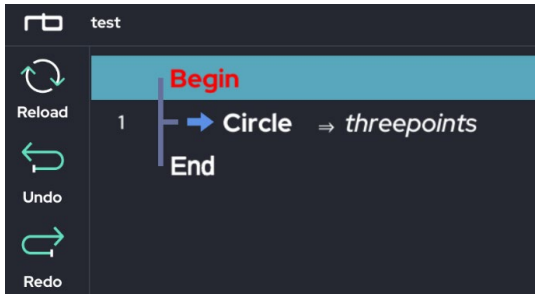
© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Spiral Circle Mode (Spiralkreisbewegung einstellen)

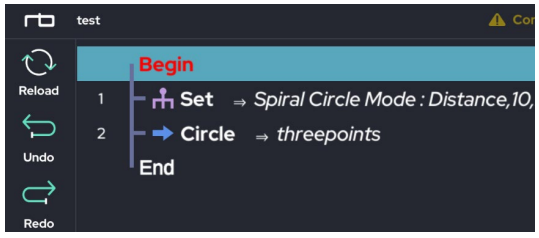


Die kreisförmige Bewegung wird mit dieser Funktion in eine spiralförmige Bewegung umzuwandeln. Wenn Sie nur die Circle-Funktion verwenden, zeichnen Sie einen Kreis/Bogen. Wenn der Set-Spiral-Modus über der Circle-Funktion verwendet wird, wird der vorhandene Kreis/Bogen in eine Spiralbewegung geändert. Um die Spiralbewegung zu implementieren, muss diese Funktion daher über der Circle-Funktion eingefügt werden.

Bsp.1) nur mit Circle-Funktion: Eine normale Kreis-/Bogenbahn wird erstellt.

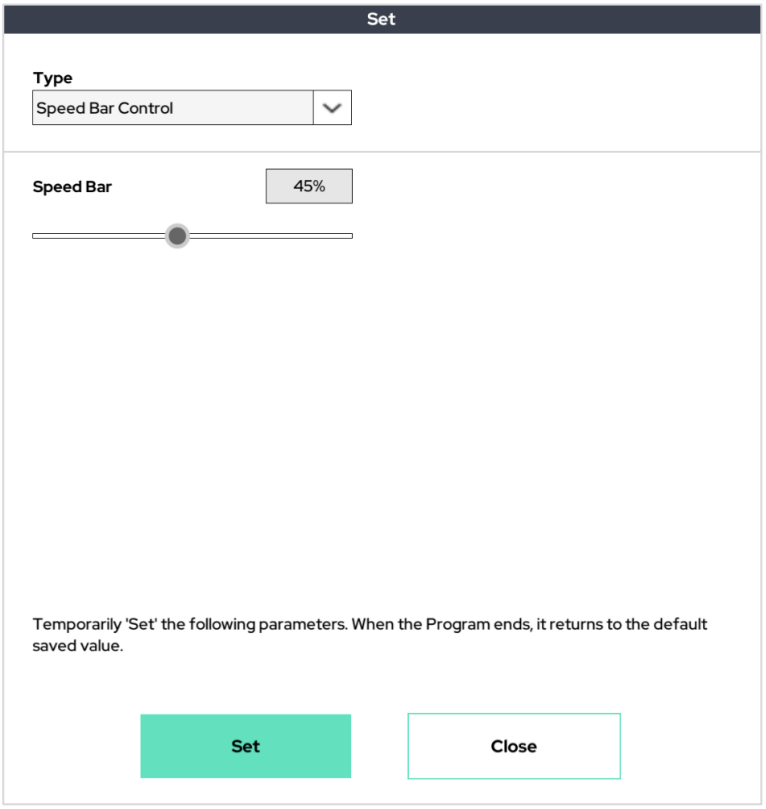


Bsp.2) Set-Spiral-Mode + Circle-Funktion: spiralförmige Trajektorie wird erstellt.



© RAINBOW ROBOTICS Inc. All rights reserved.

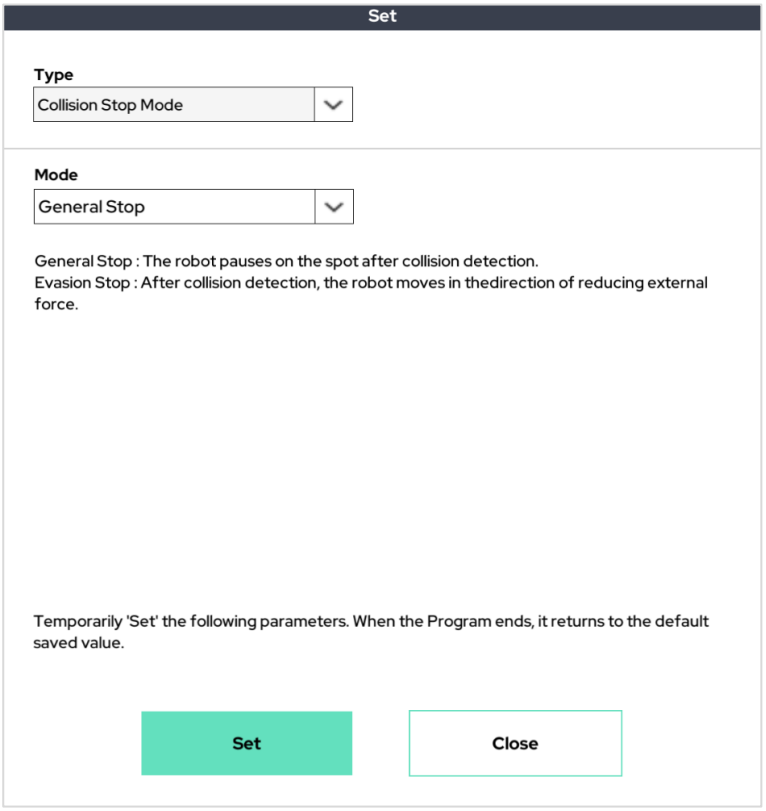
Set-Funktion: Speed Bar Control (Geschwindigkeitsleiste kontrollieren)



Mit dieser Funktion wird die Geschwindigkeitssteuerleiste der Benutzeroberfläche (unten rechts auf dem Bildschirm) mit Programm gesteuert. Sie können die Geschwindigkeitsleiste der Benutzeroberfläche ändern, indem Sie diese Funktion im gewünschten Abschnitt verwenden.

© RAINBOW ROBOTICS Inc. All rights reserved.

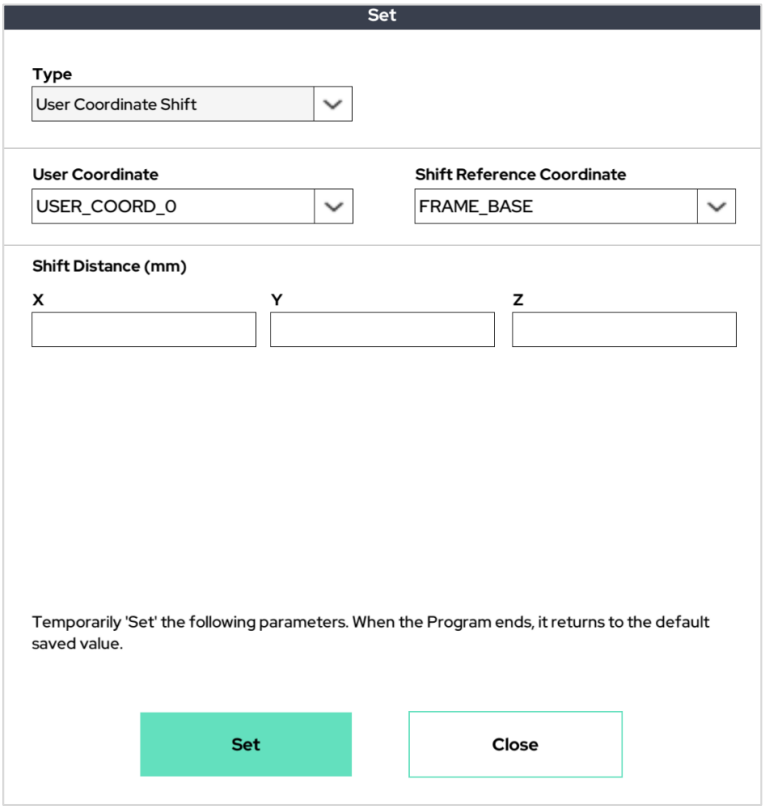
Set-Funktion: Collision Stop Mode (Stoppmodus nach Kollisionserkennung)



Der Bewegungstyp des Roboters wird gewählt, wenn er eine externe Kollision erkennt. Es gibt 2 Optionen.
General Stop: Nach dem Erkennen einer Kollision wird die Bewegung auf der Trajektorie angehalten.
Evasion Stop: Nach dem Erkennen einer Kollision bewegt sich der Roboter ein wenig in die Richtung, um einer äußeren Kraft auszuweichen, und hält dann die Bewegung auf der Trajektorie an.
Es hat dieselbe physikalische Bedeutung wie ‚Collision Detection Protection Mode'(Action after Collision) die Einstellung in Setup-Cobot.

© RAINBOW ROBOTICS Inc. All rights reserved.

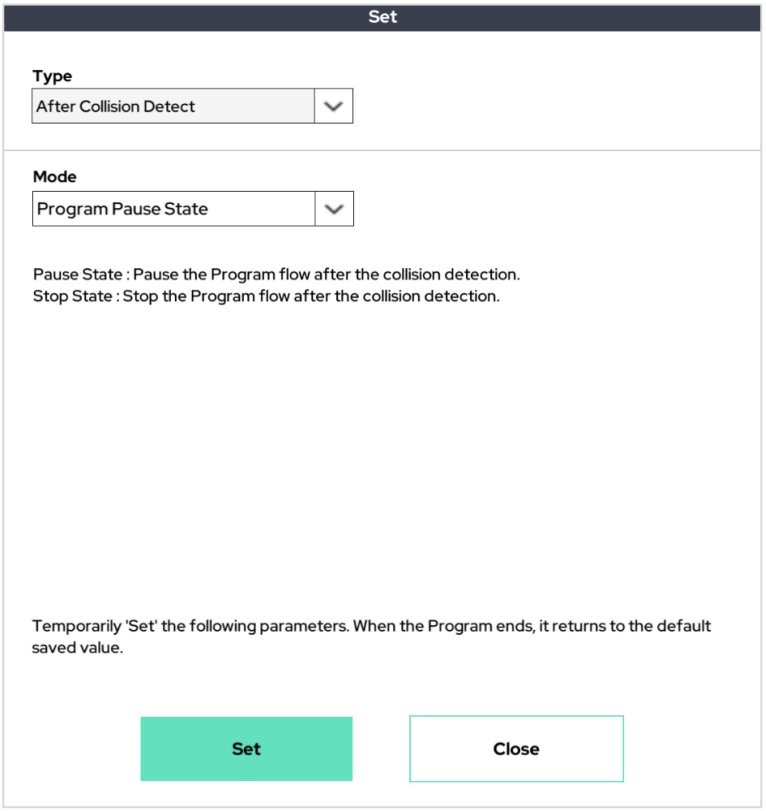
Set-Funktion: User Coordinate Shift (Benutzerkoordinatensystem verschieben)



Mit dieser Funktion wird der Ursprung des Benutzerkoordinatensystems vorübergehend verschoben (Shift). Sie können die Nummer des zu verschiebenden Benutzerkoordinatensystems und den Verschiebungsabstand festlegen und auswählen, auf welchem Koordinatensystem der Verschiebungsabstand basiert.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: After Collision Detect (Programmablauf nach Kollisionserkennung)



Mit dieser Funktion wird der Programmablauf nach externer Kollisionserkennung ausgewählt.

Die Standardeinstellung ist, das Programm anzuhalten, nachdem eine externe Kollision erkannt wurde. Nach der Kollisionserkennung wird ein Benachrichtigungsfenster zur Kollisionserkennung angezeigt und Programme und Threads werden in ihrer Grundform angehalten.

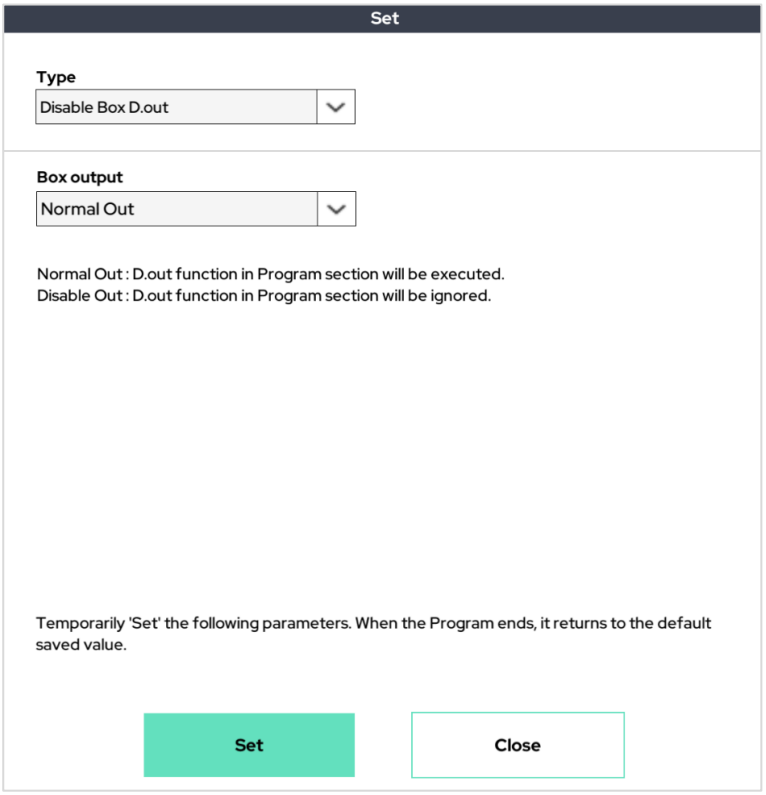
Wenn Sie das Programm nach Erkennen einer Kollision beenden möchten, können Sie diese Funktion verwenden und die Stop State-Option auswählen.

Pause State: Der Programmablauf wird angehalten, nachdem eine externe Kollision erkannt wurde.

Stop State: Der Programmablauf wird angehalten, nachdem eine externe Kollision erkannt wurde.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Disable Box D.out (Temporäre Steuerung von Steuerboxausgang)



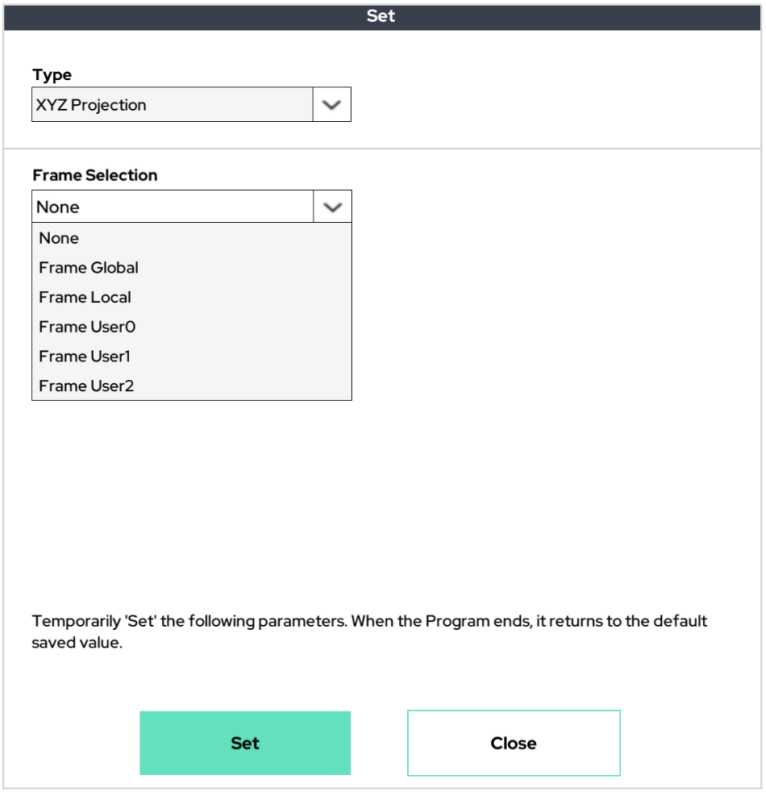
Mit dieser Funktion wird der digitale Ausgang der Steuerbox vorübergehend deaktiviert.

Auch wenn Sie den im Programm eingefügten digitalen Ausgangsbefehl nicht löschen, können Sie den digitalen Ausgangsbefehl in einem bestimmten Abschnitt ignorieren, indem Sie diesen Set-Befehl verwenden.

Zum Zweck von Entwicklungstest usw. kann diese Funktion verwendet werden. Durch Optionsauswahl kann die Ausgabe je nach Programmabschnitt deaktiviert/aktiviert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: XYZ Projection (XYZ-Position beim Betrieb der L-Serie fixieren)



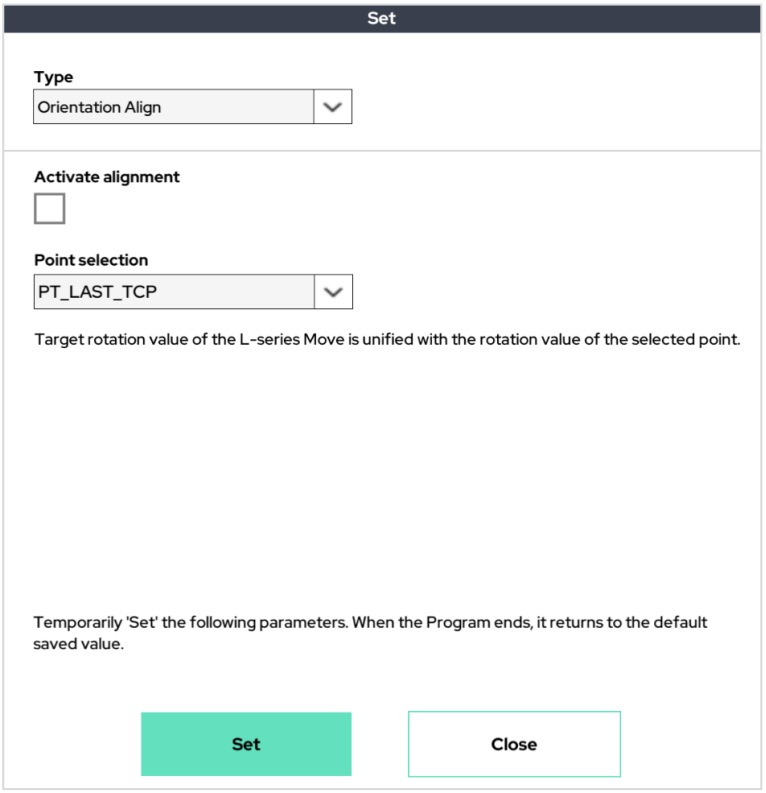
Eine Funktion zum Fixieren des Zielpositionskoordinatenwerts von Bewegungen der L-Serie (Bsp.: MoveL. MovePB, Circle usw.). Wenn Sie den zu fixierenden Wert und das Referenzkoordinatensystem auswählen, wird der Positionskoordinatenwert des Zielpunkts oder Sollwerts als Wert der ausgewählten Achse des ausgewählten Koordinatensystems festgelegt.

Wenn beispielsweise das Basiskoordinatensystem (Global) als Koordinatensystem ausgewählt und Z-Projection 100 mm ausgewählt/geschrieben wird, wird die Z-Höhe aller Bewegungssollwerte /eingestellten Koordinatenwerte auf einmal als 100 mm übernommen.

Mit dieser Funktion ist ebenfalls eine Funktion von Set-Funktion für jeden Programmabschnitt zu aktivieren/deaktivieren. Wenn Sie die Funktion deaktivieren möchten, wählen Sie einfach ‚None‘ im Koordinatensystem aus.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Orientation Align (Rotationwerte beim Betrieb der L-Serie fixieren)



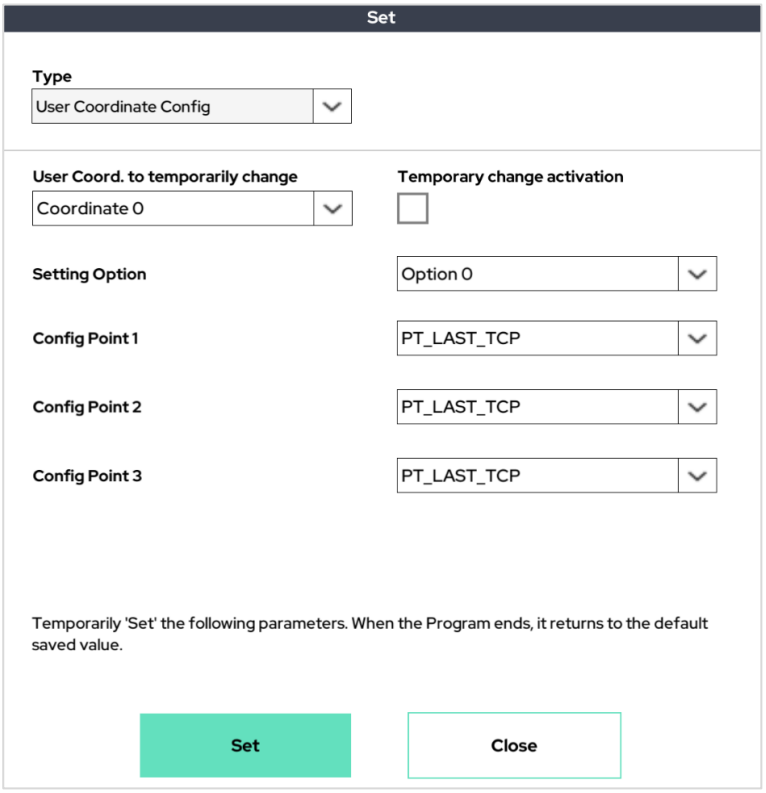
Mit dieser Funktion sind Zielrotationskoordinatenwerte von Bewegungen der L-Serie (Bsp.: MoveL. MovePB, Circle usw.) zu fixieren.

Die Rotation der Bewegungen der L-Serie wird mit dem Rotationswert des ausgewählten ‚Point' fixiert.

Als Unterfunktion der Set-Funktion kann diese Funktion je nach Programmabschnitt ein- oder ausgeschaltet werden, wenn Sei die TCP-Rotation auf einmal vereinheitlichen möchten.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: User Coordinate Config (Benutzerkoordinatensystem temporär einstellen)



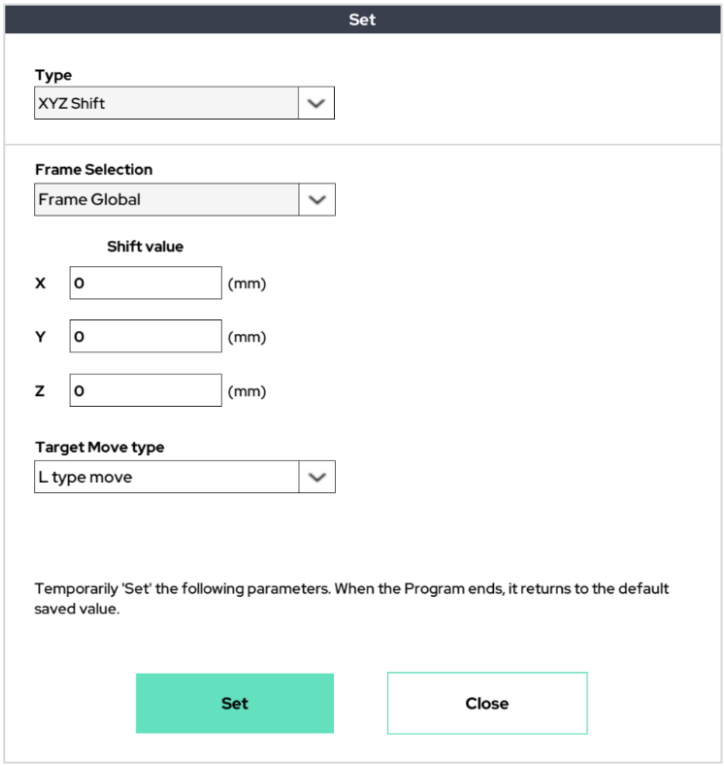
Mit dieser Funktion können die Einstellungen des Benutzerkoordinatensystems vorübergehend geändert werden.

Durch die Auswahl von 3 Punkten in der Mitte des Programmablaufs kann die Einstellung des Benutzerkoordinatensystems im Laufe des Programms beliebig geändert werden.

Da es sich um eine Unterfunktion von Set-Funktion handelt, werden die Einstellungen des Benutzerkoordinatensystems nach Beendigung des Programms auf ihre Standardwerte zurückgesetzt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: XYZ Shift (Positionswert des Zielpunkts verschieben)



Mit dieser Funktion kann der Zielpunkt vorübergehend geändert werden.

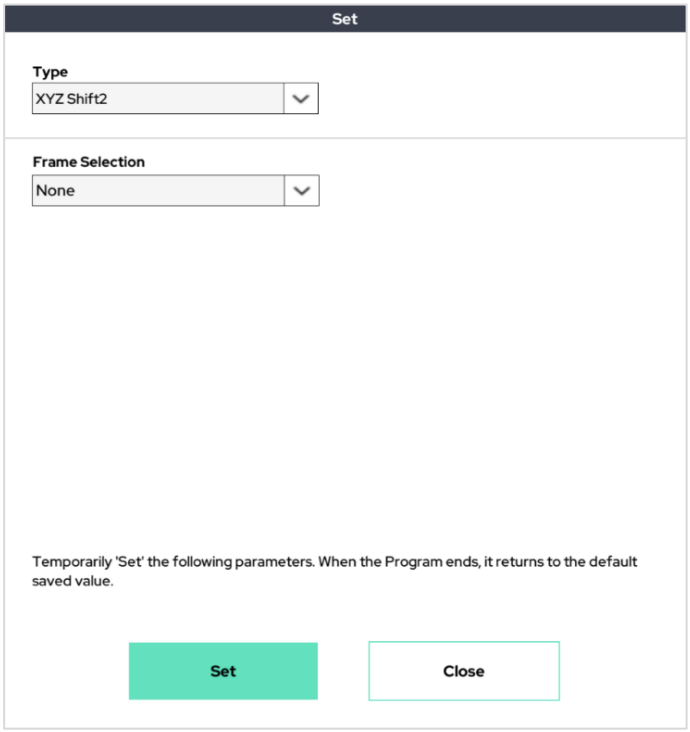
Sie können das Basis-/Werkzeug-/Benutzerkoordinatensystem auswählen und einen Verschiebungswert vom Zielpunkt eingeben.

Wählen Sie dabei aus, ob diese Verschiebung nur auf die Bewegung der L-Serie oder auf die Bewegung der L-Serie und der J-Serie angewendet werden soll.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: XYZ Shift2 (Positionswert des Zielpunkts verschieben 2)



Mit dieser Funktion kann der Zielpunkt vorübergehend geändert werden.

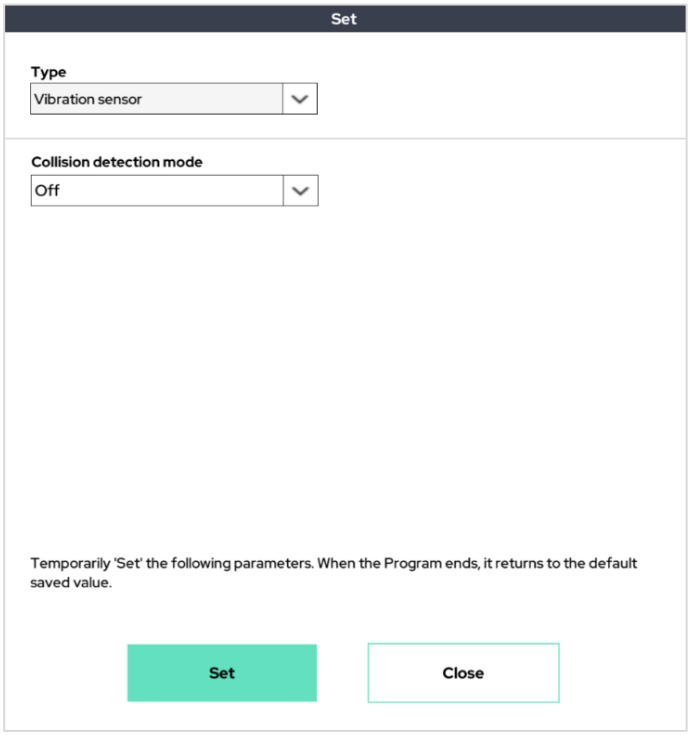
Sie können ein Basis-/Werkzeug-/Benutzerkoordinatensystem auswählen und eine Verschiebungswert vom Zielpunkt aus eingeben.

Dabei gilt diese Verschiebung nur für die Bewegung der L-Serie, und Sie können Verschiebungswerte sowohl für XYZ-Positionswerte als auch für Rotationswerte eingeben.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Vibration sensor (Ein/Aus von Vibrationserkennungssensor)

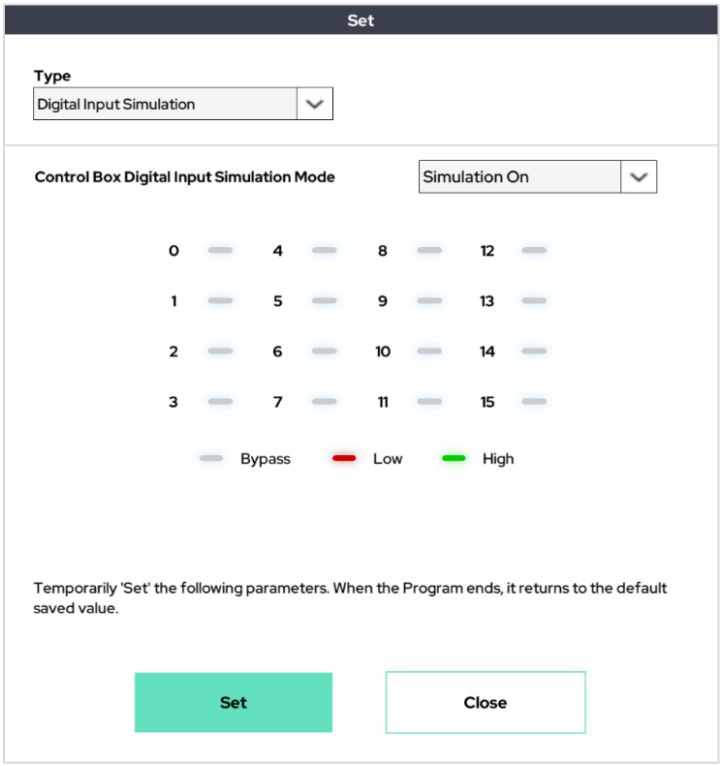


Mit dieser Funktion wird die Kollision aufgrund Vibrationen bei der Kollisionserkennung ausgeschlossen.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Digital Input Simulation (Simulation von digitalen Eingangssignalen)



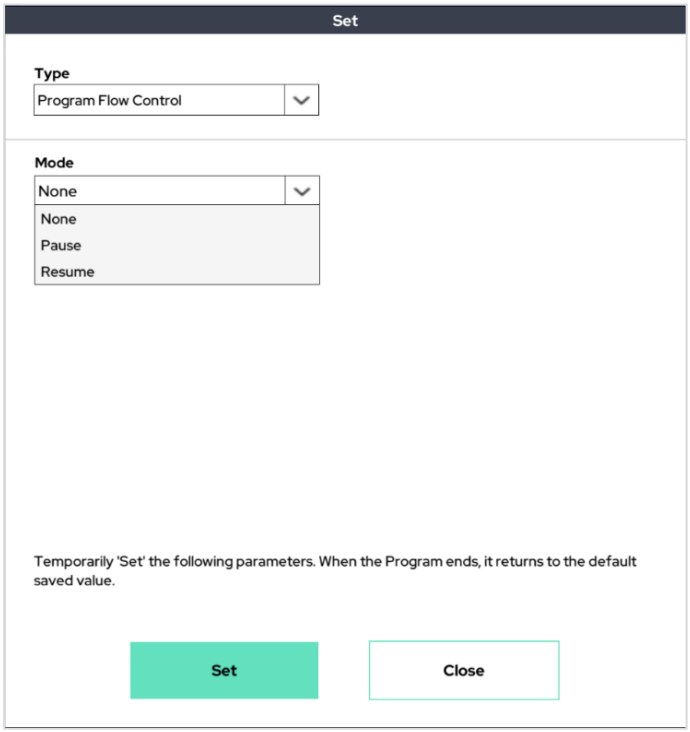
Mit dieser Funktion wird ein virtuelles Signal durch Simulation eingegeben.

Erstellen Sie die gewünschte Eingabe, indem sie den Status des Anschlusses festlegen, an dem Sei Eingaben vornehmen möchten.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Program Flow Control (Programmfluss kontrollieren)

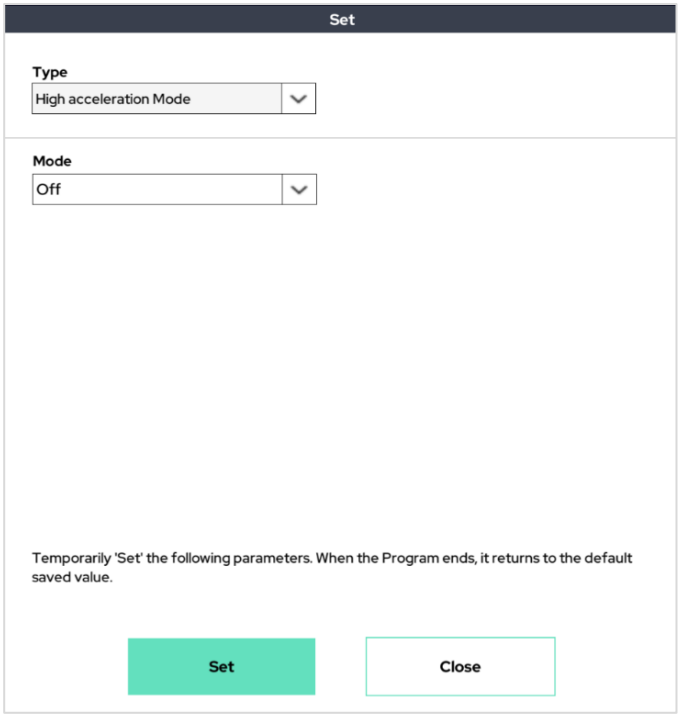


Wenn ein Programm läuft, ist es eine Funktion, die es ermöglicht, ohne Alarm und Ein/Aus anzuhalten und fortsetzen.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

Set-Funktion: High Acceleration Mode (Beschleunigungsmodus mit Hochgeschwindigkeit)

© RAINBOW ROBOTICS Inc. All rights reserved.

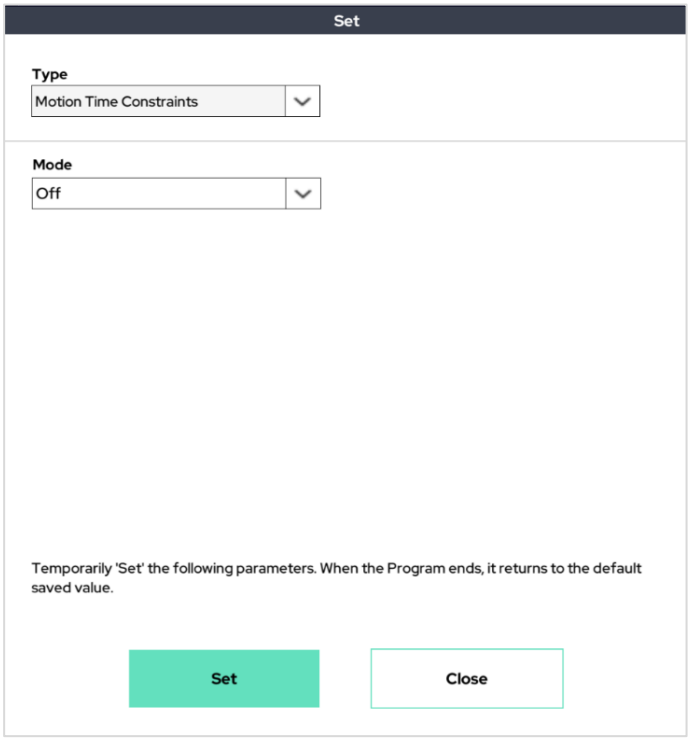


Der Hochgeschwindigkeits-Beschleunigungsmodus ist eine Funktion, die die Zeit verkürzt, die die Bewegungsgeschwindigkeit des Roboters benötigt, um die gewünschte Bewegungsgeschwindigkeit zu erreichen, indem das Verzögerungs-/Beschleunigungsprofil geändert wird.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Motion Time Constraints (Betriebszeit beschränken)

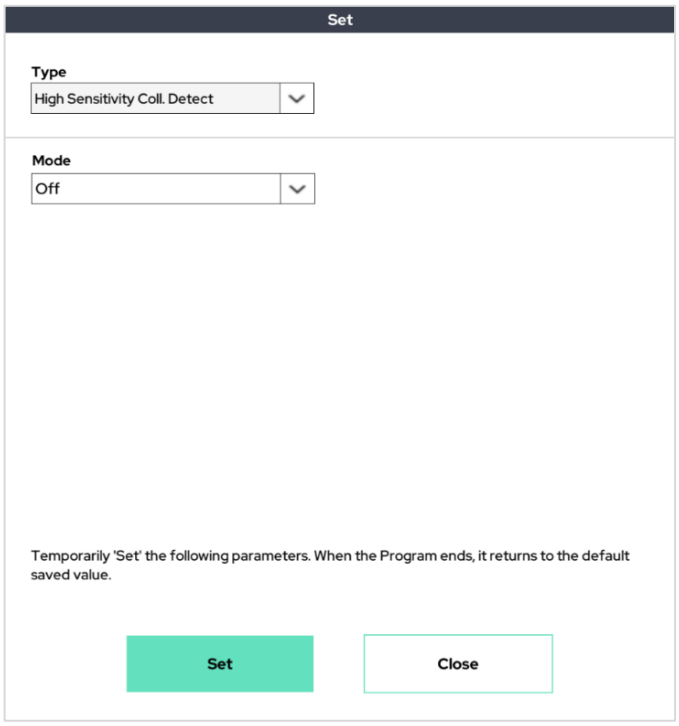


Die Bewegungszeitbeschränkung ist eine Funktion, die die Zeit zum Bewegen zwischen Punkten auf die Eingabezeit beschränkt. Dabei ist es nur möglich, die Zeit zu verlängern.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: High Sensitivity Coll. Detect (Hochempfindliche Kollisionserkennung)

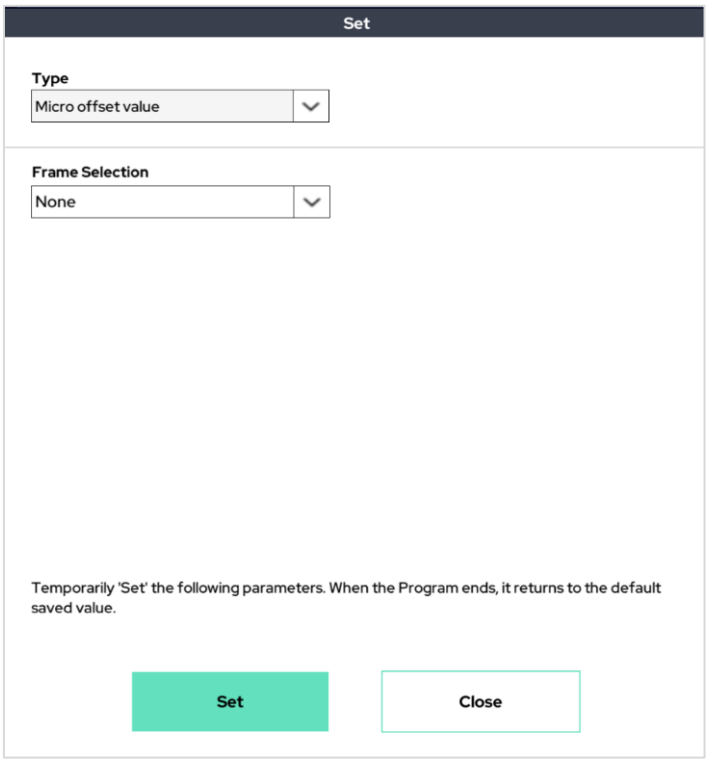


Die hochempfindliche Kollisionserkennung ist eine Funktion, die eine Kollisionserkennung mit 30 % höherer Empfindlichkeit als die herkömmliche ermöglicht. Im Setup ist die Empfindlichkeit, die die Kollisionserkennung am empfindlichsten gemacht hat, auch 30 % empfindlicher als 0 %.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Micro offset value (Offset-Feinsteuerung)

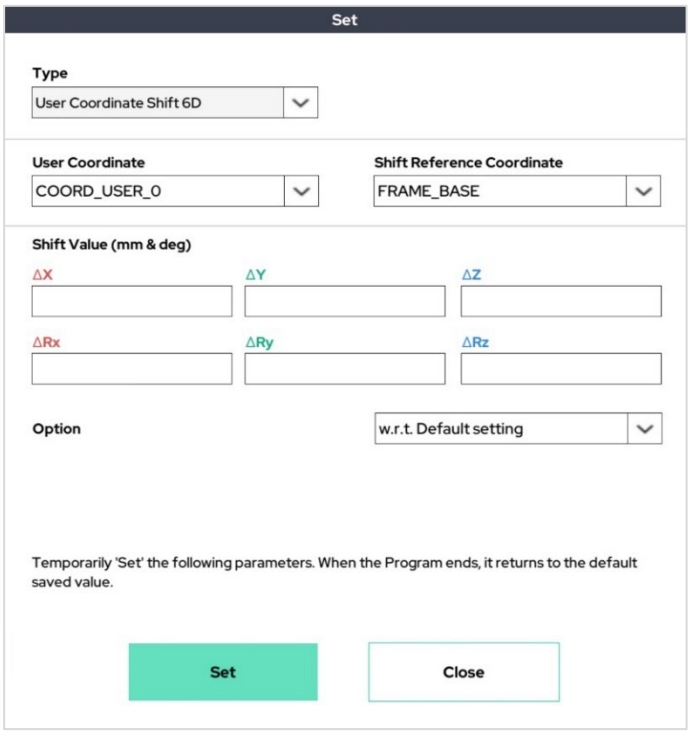


Mit dieser Funktion kann ein gewisse Versatz (Offset) zum Koordinatensystem gegeben werden. Diese Funktion ermöglicht das temporäre Einstellen von Offsets bis zu 20 mm.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: User Coordinate Shift 6D (Benutzerkoordinatensystem verschieben 6D)

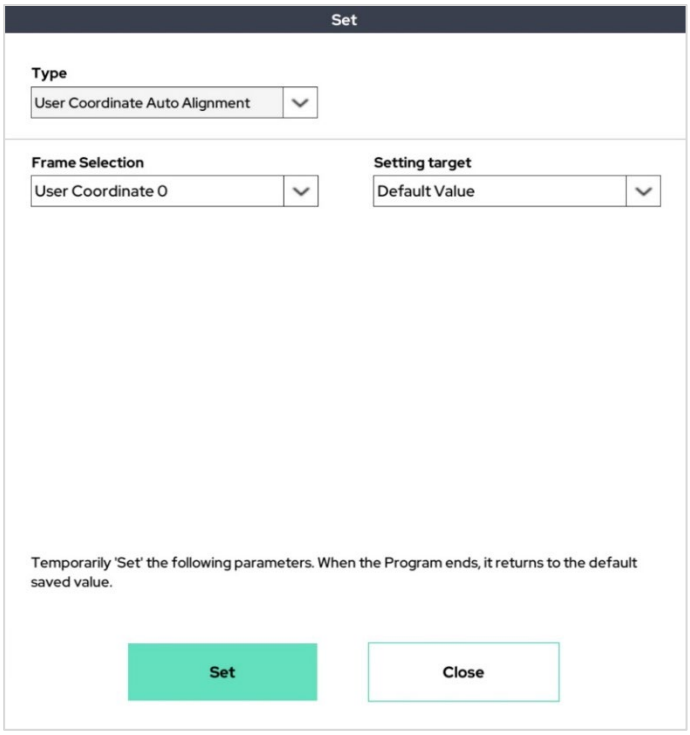


Mit dieser Funktion wird Benutzerkoordinatensystem vorübergehend verschoben und können Sie die Position und Rotation des Benutzerkoordinatensystems ändern.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: User Coordinate Auto Alignment (Benutzerkoordinatensystem automatisch ausrichten)

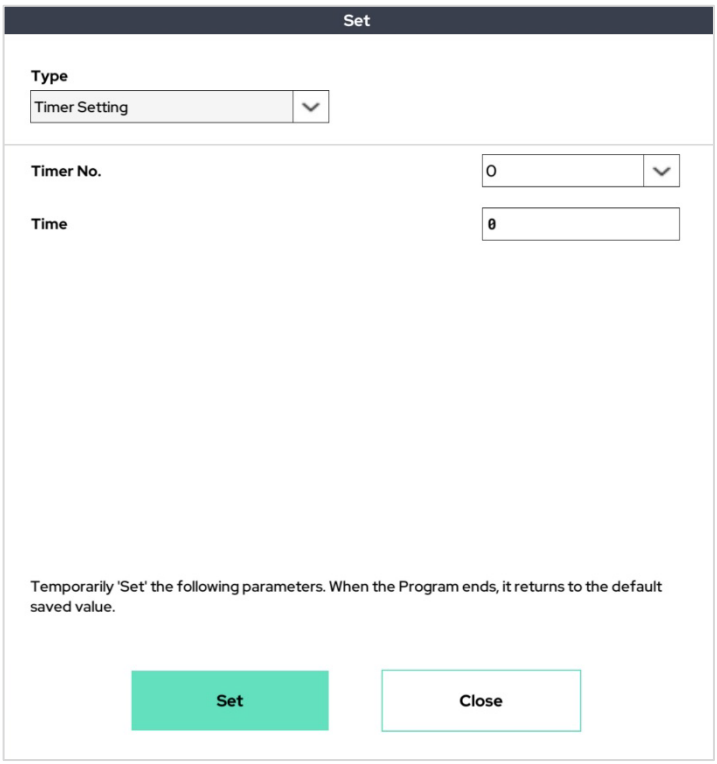


Mit dieser Funktion können Sie das ausgewählten Benutzerkoordinatensystem vom vorhandenen Koordinatensystem in das TCP-Koordinatensystem des aktuellen Roboters ändern, Es ist auch möglich, auf das zuvor eingestellte Benutzerkoordinatensystem zurückzuschalten.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: Timer Setting (Timer einstellen)

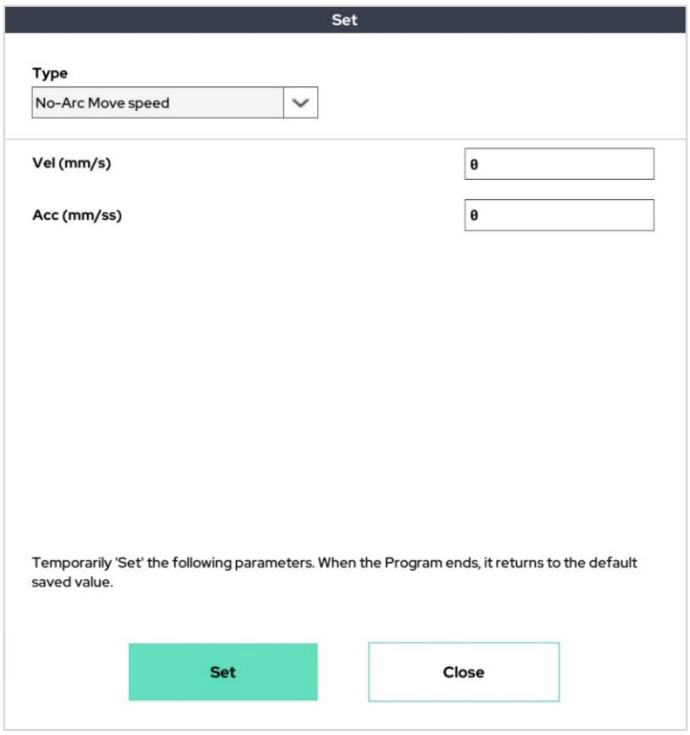


Mit dieser Funktion können Sie den Timer auswählen, den Sie verwenden möchten, und den Anfangswert des Timers einstellen. Der Timer startet ab dem von Ihnen eingestellten Anfangswert.

Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Set-Funktion: No-Arc Move speed (No-Arc-Bewegungsgeschwindigkeit)



Stellen Sie die Bewegungsgeschwindigkeit des Roboters im No-Arc-Zustand ein, in dem nicht geschweißt wird.

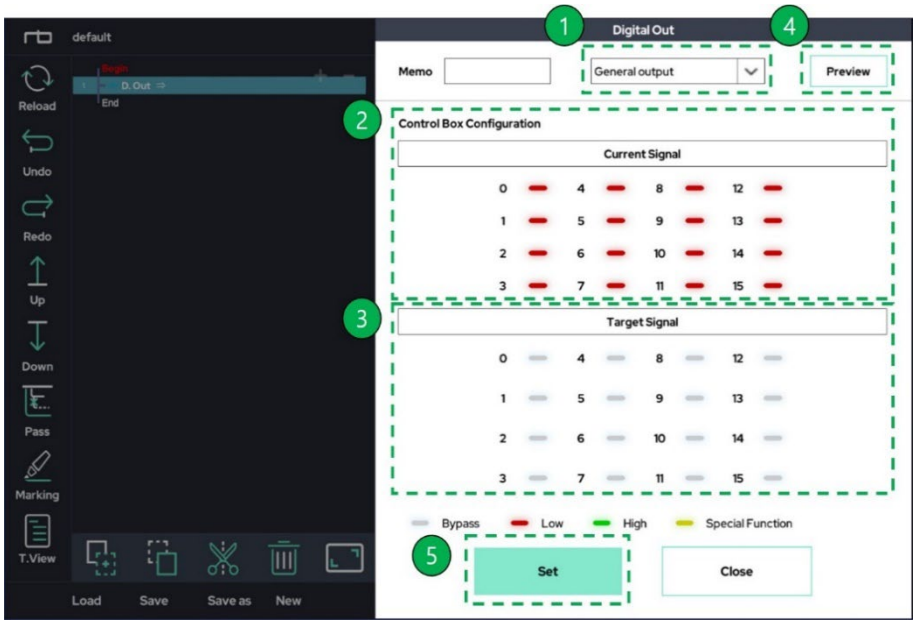
Da es sich um eine Unterfunktion von Set-Funktion handelt, kehrt sie zu ihrem Standardwert zurück, wenn das Programm endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ D.Out (Digital out)-Funktion :

Dies ist ein Befehl zur Steuerung des digitalen Ausgangs der Steuerbox. Diese Funktion gibt das ausgewählte Signal über den ausgewählten Port unter 16 Ports (Nr. 0-15) aus. Jeder Port kann 3 Einstellungen haben.: High-Signal, Low-Signal und Bypass.

Nachdem Sie die D.out-Funktion zum Programm hinzugefügt und daraufgeklickt haben, erscheint das folgende Popup-Fenster.



① Wählen Sie die detaillierten Funktionen aus, die in der D.out-Funktion verfügbar sind.

② Zeigt den Status des Digitalausgangs an, der derzeit von der Steuerbox ausgegeben wird.

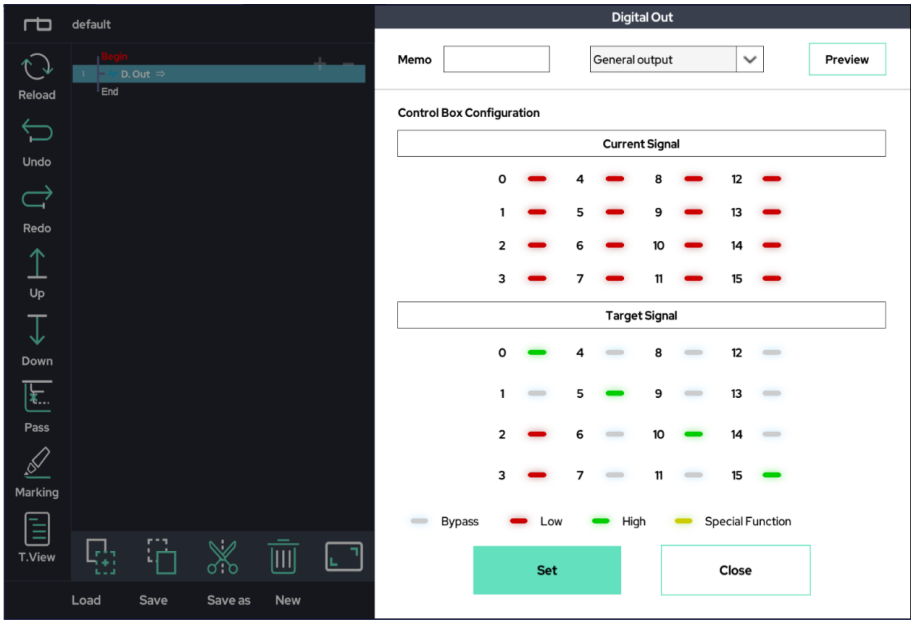
③ Die gewünschte Einstellung befindet sich dort, wo Sie sie auswählen. Bypass, Low, High können über die 3-stufige Umschalttaste ausgewählt werden.
Bypass: Beibehalten des vorherigen Ausgangssignalzustands (grau).
Low: Setzt das Ausgangssignal auf einen niedrigen Pegel (rot).
High: Setzt das Ausgangssignal auf High-Pegel (grün).

④ Dies ist eine Funktion, die die in Punkt 3 ausgewählten Einstellungen als Vorschau gedruckt.

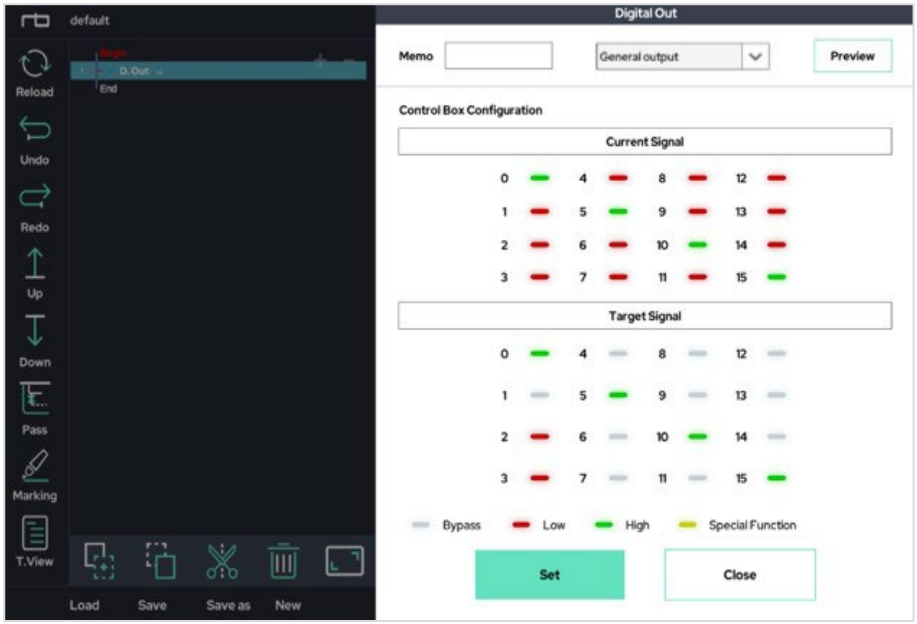
⑤ Stellen Sie die D.out-Funktion auf die in Punkt 3 ausgewählte Einstellung ein.

© RAINBOW ROBOTICS Inc. All rights reserved.

D.Out : Einzelausgang



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein und drücken Sie die Vorschau-Taste, um ein digitales Ausgangssignal von der Steuerbox auszugeben, wie unten gezeigt.

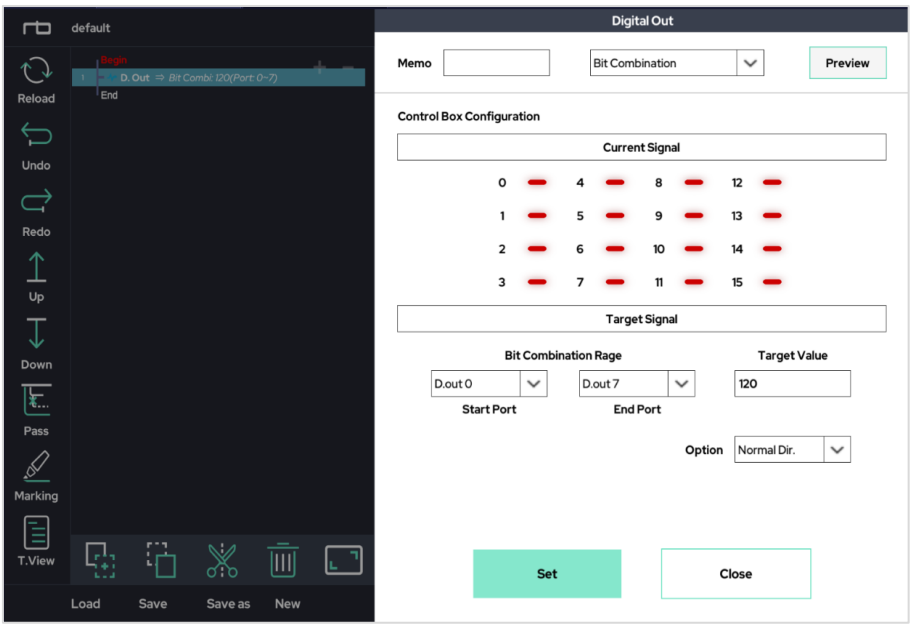


Sie können sehen, dass das Feld ‘Current Signal’ auf der rechten Seite des Bildes geändert wurde.

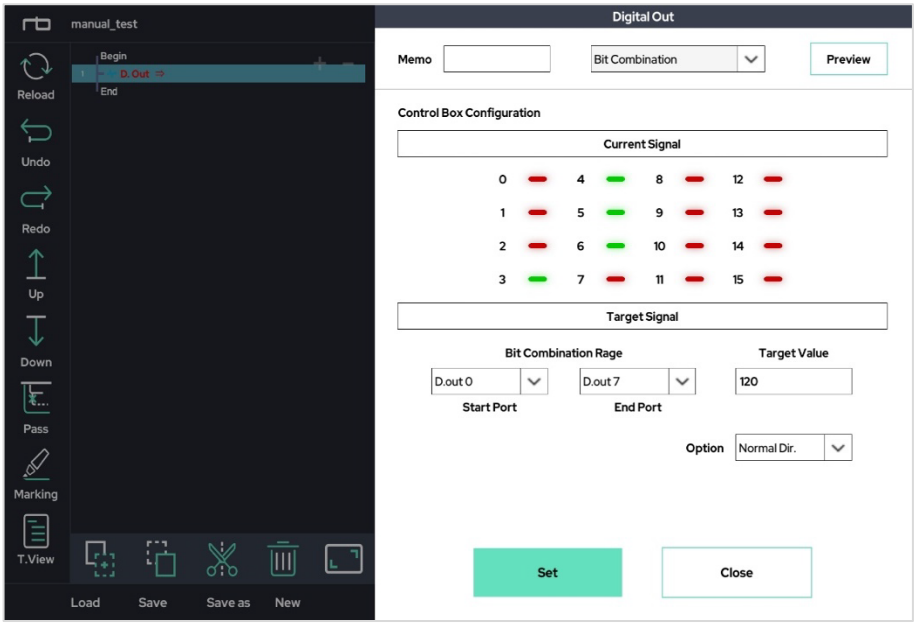
© RAINBOW ROBOTICS Inc. All rights reserved.

D.Out : Bit-Kombination

Geben Sie nach Auswahl des zu verwendenden Start- und Endanschlusses den gewünschten Wert in den Zielwert ein, um den digitalen Ausgang als Bit-Kommunikation auszugeben.



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein und drücken Sie die Vorschau-Taste, um ein digitales Ausgangssignal von der Steuerbox auszugeben, wie unten gezeigt.

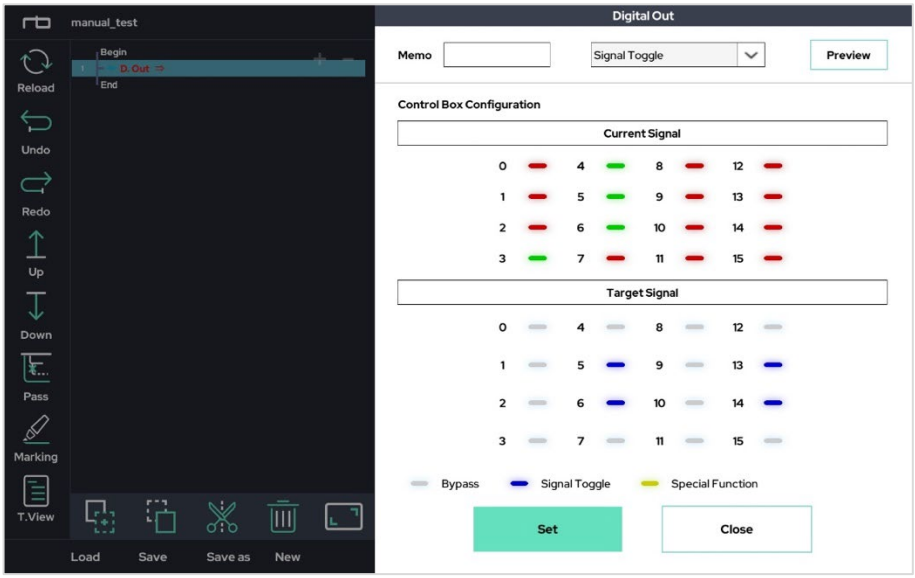


Das Feld ‘Current Signal’ ist auf der rechten Seite des Bildes geändert.

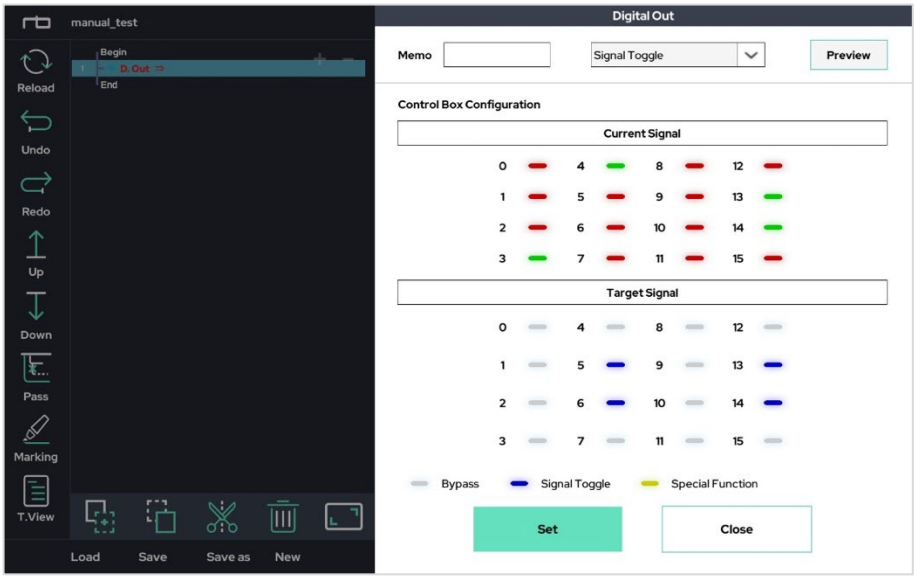
© RAINBOW ROBOTICS Inc. All rights reserved.

D.Out : Signal Toggle

Das High-Signal ändert den Ausgangswert auf Low und das Low-Signal ändert den Ausgangswert auf High.



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein (Toggle-Ausgangssignal wird blau dargestellt) und drücken Sie die Vorschau-Taste, um ein digitales Ausgangssignal von der Steuerbox auszugeben, wie unten gezeigt.

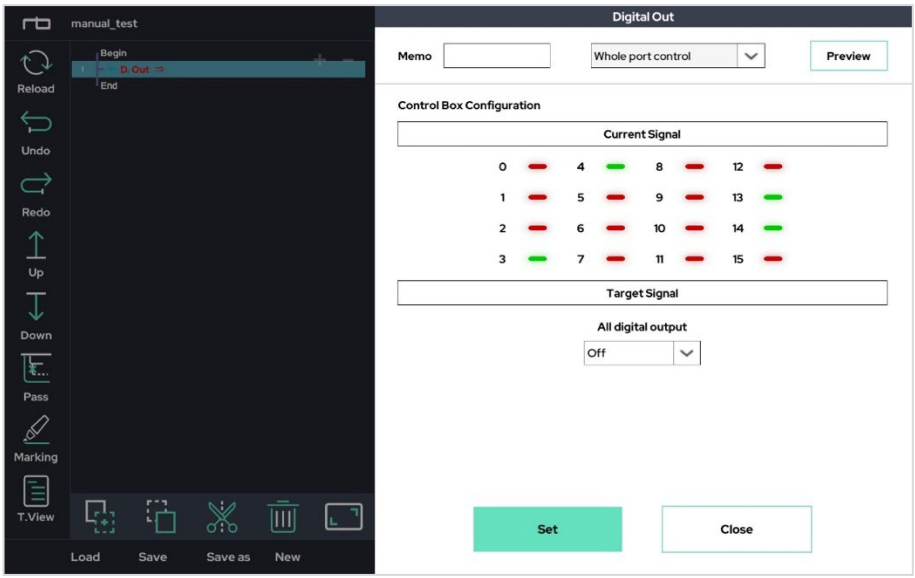


Das Feld ‘Current Signal’ ist auf der rechten Seite des Bildes geändert

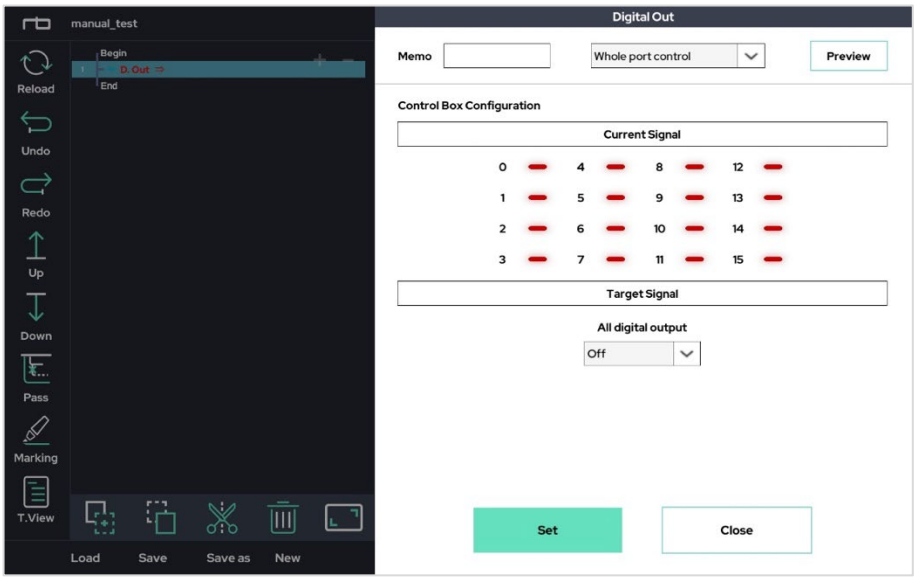
© RAINBOW ROBOTICS Inc. All rights reserved.

D.Out : Gesamte Signalausgabe

Signale von allen Ausgangsports können gleichzeitig auf High/Low geschaltet werden.



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein und drücken Sie die Vorschau-Taste, um ein digitales Ausgangssignal von der Steuerbox auszugeben, wie unten gezeigt.

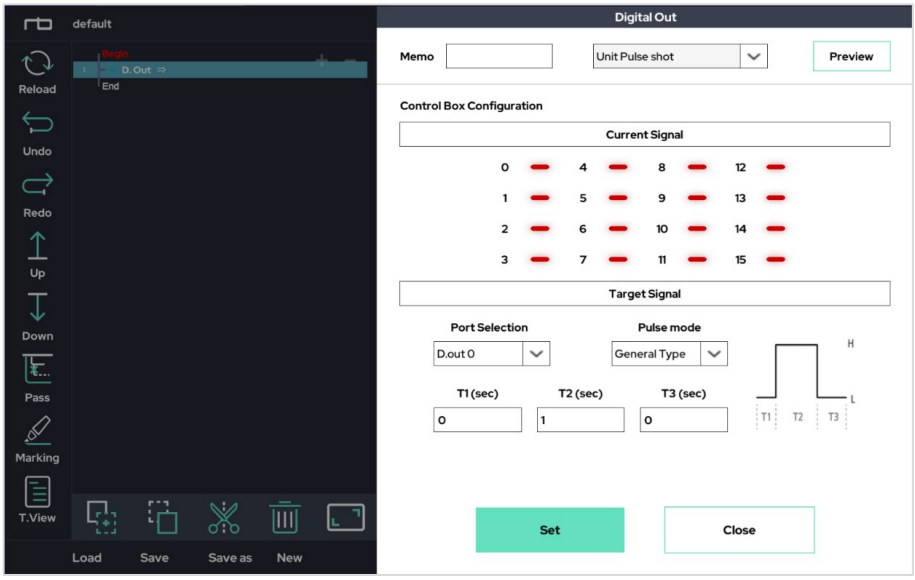


Das Feld ‘Current Signal’ ist auf der rechten Seite des Bildes geändert.

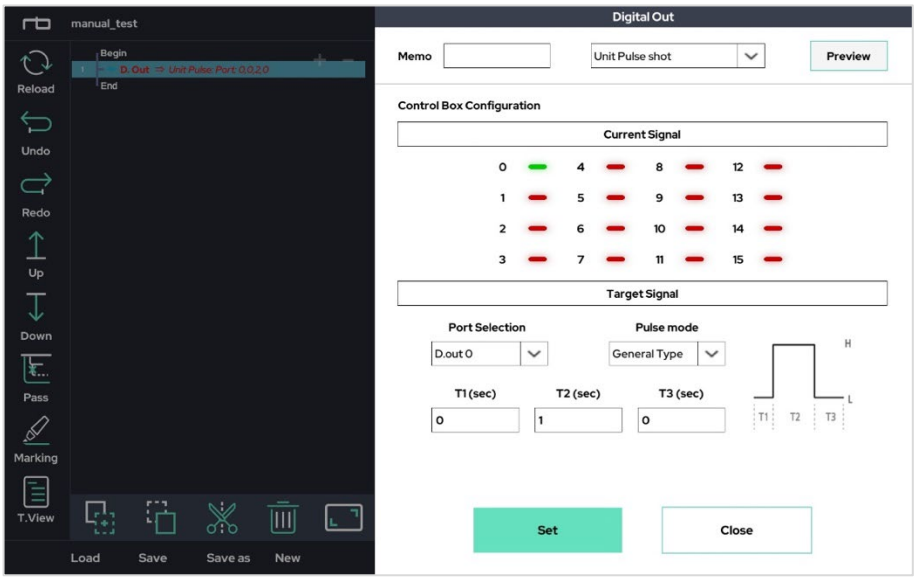
© RAINBOW ROBOTICS Inc. All rights reserved.

D.Out : Ausgabe von Einheitsimpuls

Wenn Sie den zu verwendenden Port auswählen und eine Zeit zwischen 0 und 2 Sekunden in T1 bis T3 eingeben, wird das Einheitsimpulssignal entsprechend der eingegebenen Zeit ausgegeben.



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein und drücken Sie die Vorschau-Taste, um ein digitales Ausgangssignal von der Steuerbox auszugeben, wie unten gezeigt.



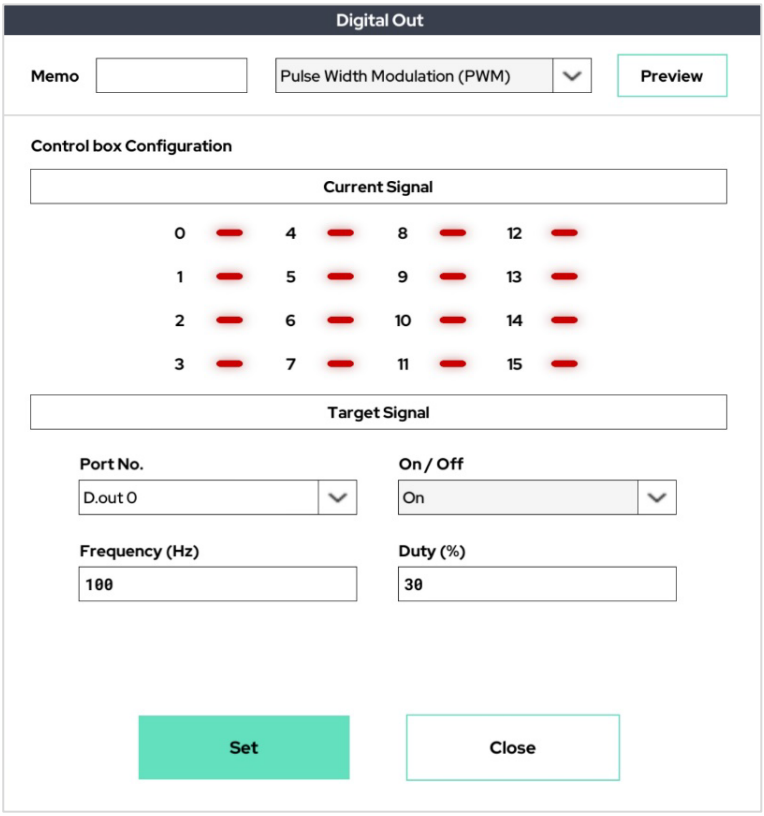
Das Feld ‘Current Signal’ ist auf der rechten Seite des Bildes geändert.

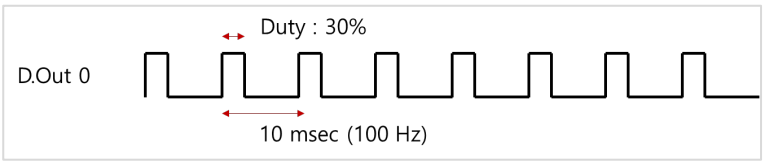
© RAINBOW ROBOTICS Inc. All rights reserved.

D.Out : Pulsweitenmodulation (PWM)

PWM (Pulse Width Modulation)-Ausgangsfunktion. Es ist eine Funktion, ein PWM-Signal durch den digitalen Ausgangsanschluss zu übertragen, wenn die Frequenz und das Tastverhältnis des PWM-Impulses eingegeben werden.

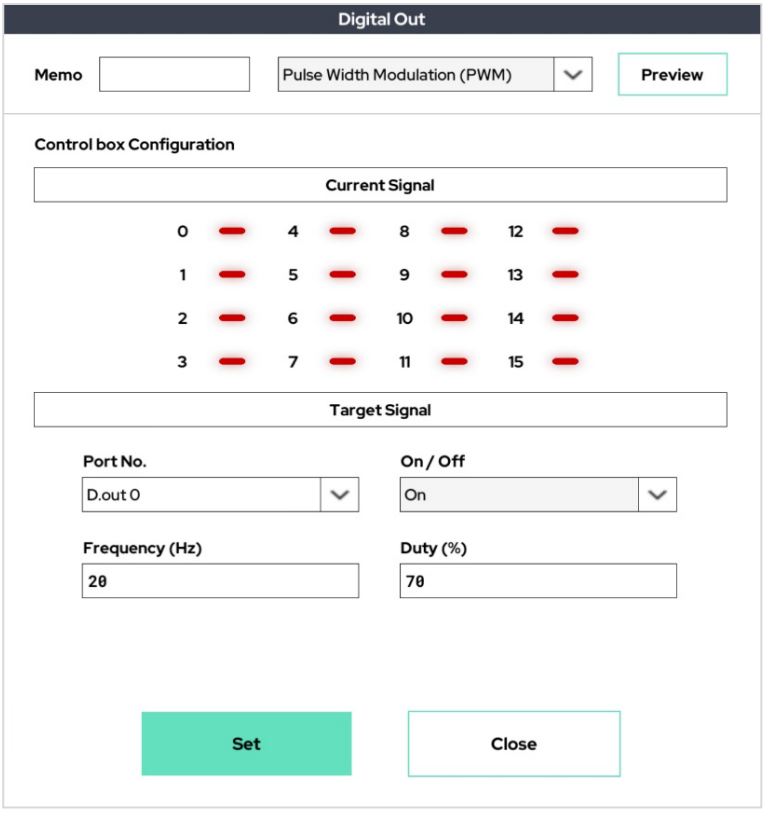
PWM Bsp. 1)

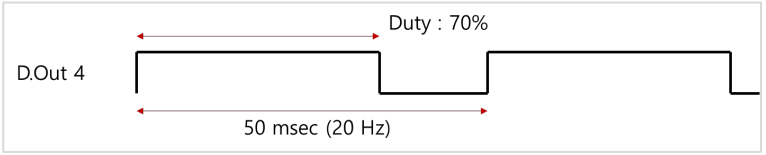




© RAINBOW ROBOTICS Inc. All rights reserved.

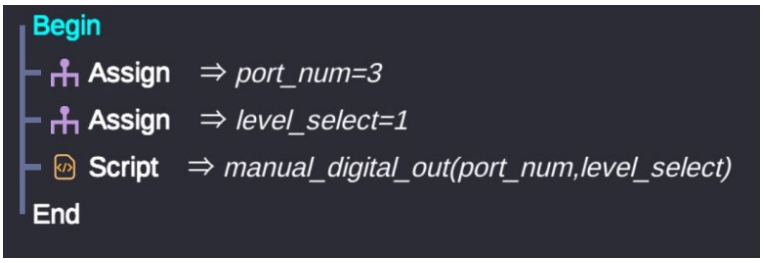
PWM Bsp. 2)





Zusätzlich zur Verwendung der D.out-Funktion können Sie einen Befehl zur Ausgabe digitaler Ausgänge mit der Skriptfunktion erstellen, wie unten gezeigt.

※ Script-Funktion: manual_digital_out(Portnummer, Ausgangspegel)



© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

Vorsicht:

1) Wenn einem bestimmten digitalen Ausgangsport in Setup-I/O eine Sonderfunktion zugewiesen wird, kann der entsprechende Port nicht über die D.out-Funktion verwendet werden.
2) Als Sonderfunktionen belegte Ports werden im D.out-Fenster gelb dargestellt.
3) Sie können die Memo-Funktion in der oberen rechten Ecke des D.out-Fensters verwenden, um eine Notiz über die Einstellung zu hinterlassen.
4) Bevor sie den Digitalausgang verwenden, machen Sie sich bitte mit den vom Hersteller bereitgestellten elektrischen Eigenschaften des Digitalausgangsports vertraut.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ An.out(Analog out)-Funktion:

Dieser Befehl steuert den analogen Ausgang der Steuerbox. Diese Funktion gibt die ausgewählte Spannung über den ausgewählten Port unter den 4 (0-3) analogen Ports aus. Jeder Port kann mit einem Spannungsbereich von 0 bis 10 V ausgeben.

Wenn Sie die An.out-Funktion zum Programm hinzufügen und draufklicken, erscheint das folgende Popup-Fenster.

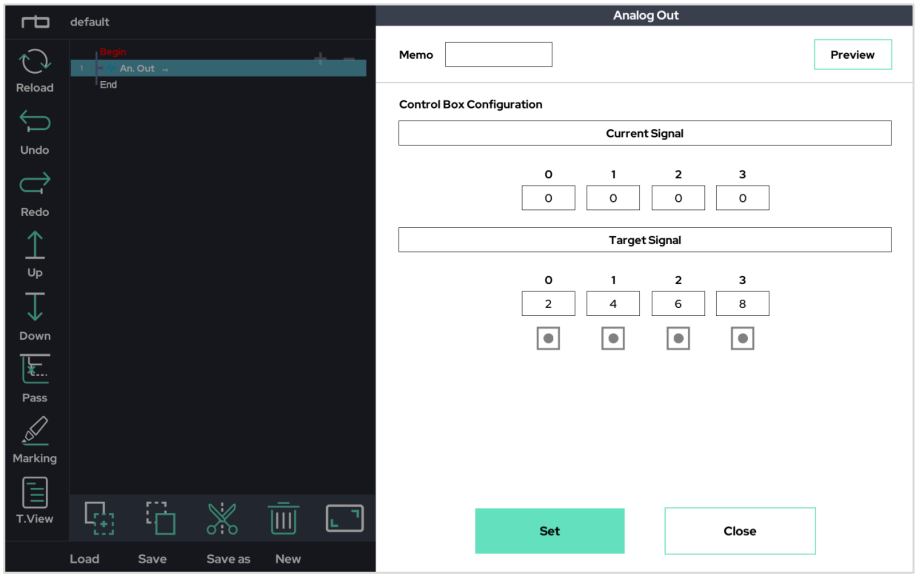
① Es zeigt den Status des Analogausgangs an, der derzeit von der Steuerbox ausgegeben wird. Dies ist das Feld zur Eingabe des gewünschten Spannungseinstellung. Wenn die Checkbox deaktiviert ist, handelt es sich um die Bypass-Funktion.

② Diese Einstellung behält die vorhandene Ausgangsspannung bei. Aktivieren Sie die Checkbox und geben Sie die gewünschte Spannung als 0 bis 10 V ein, um die Spannung einzustellen.

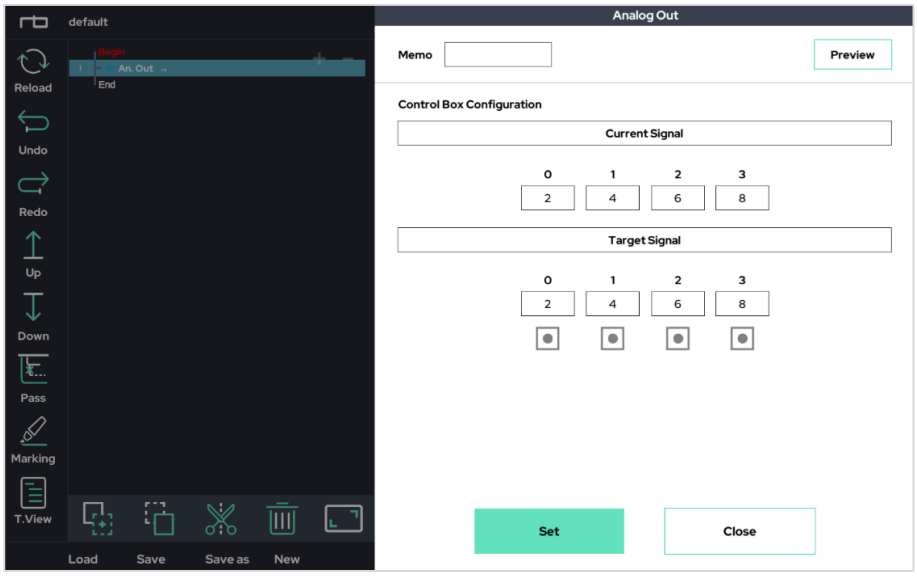
③ Dies ist eine Funktion zum Drucken der unter Punkt 2 ausgewählten Einstellungen als Vorschau.

④ Stellen Sie die An.out-Funktion mit der in Punkt 2 ausgewählten Einstellung ein.

© RAINBOW ROBOTICS Inc. All rights reserved.



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein und drücken Sie die Vorschau-Taste, um ein analoges Ausgangssignal von der Steuerbox auszugeben, wie unten gezeigt.

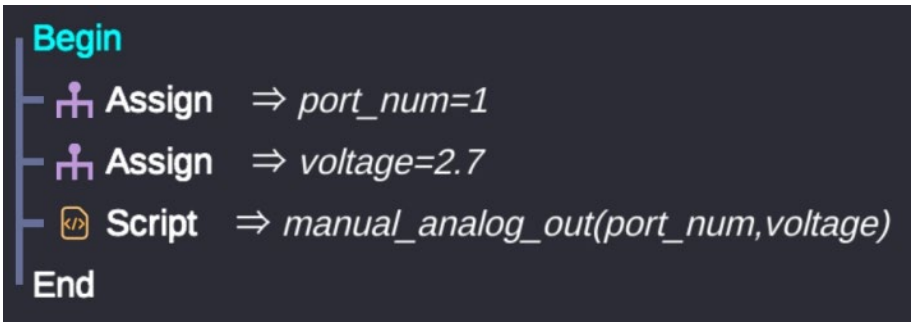


Das Feld ‘Current Signal’ ist auf der linken Seite des Bildes geändert.

© RAINBOW ROBOTICS Inc. All rights reserved.

Zusätzlich zur Verwendung der An.out-Funktion können Sie einen Befehl erstellen, der eine analoge Ausgabe mit der Script-Funktion ausgibt, wie unten gezeigt.

※ **Script-Funktion**: manual_analog_out (Portnummer, Ausgangsspannung)



Vorsicht

Vorsicht:

1) Sie können eine Notiz über die Einstellung hinterlassen, indem Sie die Memo-Funktion in der oberen rechten Ecke des An.out-Fensters verwenden.
2) Bevor Sie den Analogausgang verwenden, machen Sie sich bitte mit den vom Hersteller bereitgestellten elektrischen Eigenschaften des Analogausgangsports vertraut.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Tool Out-Funktion:

Der Werkzeugflansch hat zwei digitale Ausgänge. Signale von zwei digitalen Ausgängen können zugeordnet werden. Außerdem kann die Höhe der vom Werkzeugflansch auszugebenden Spannung (0V oder 12V oder 24V) gemeinsam eingestellt werden.

Nachdem Sie die Tool-Funktion zum Programm hinzugefügt und darauf geklickt haben, erscheint das folgende Popup-Fenster.

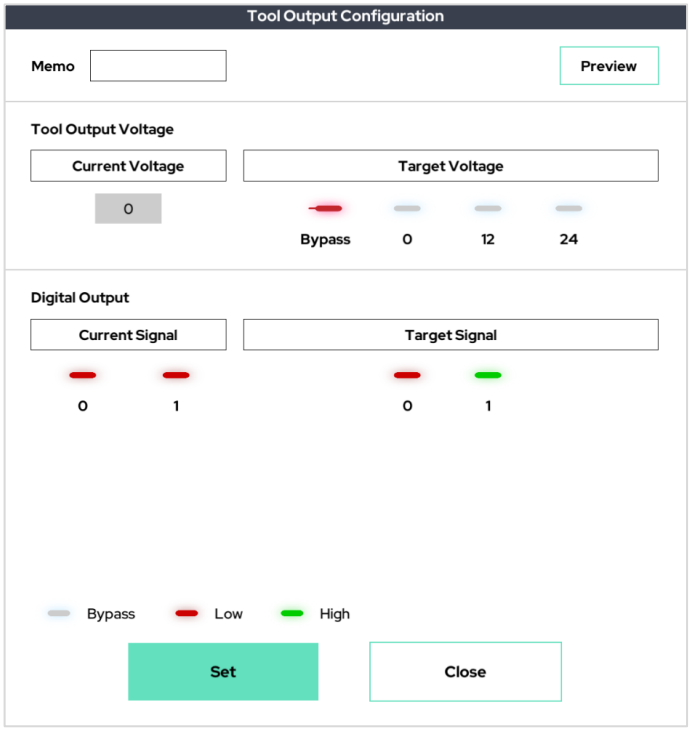
① Es zeigt den Status des Werkzeugflansches an, der gerade vom Ende des Roboters ausgegeben wird.

② Dies ist das Feld zum Einstellen der gewünschten Ausgangsspannung und des Digitalausgangs. Die Ausgangsspannung ist wählbar zwischen Bypass, 0V, 12V und 24V. Der digitale Ausgang wird mit einer 3-stufigen Umschalttaste zwischen Bypass, Low und High ausgewählt.

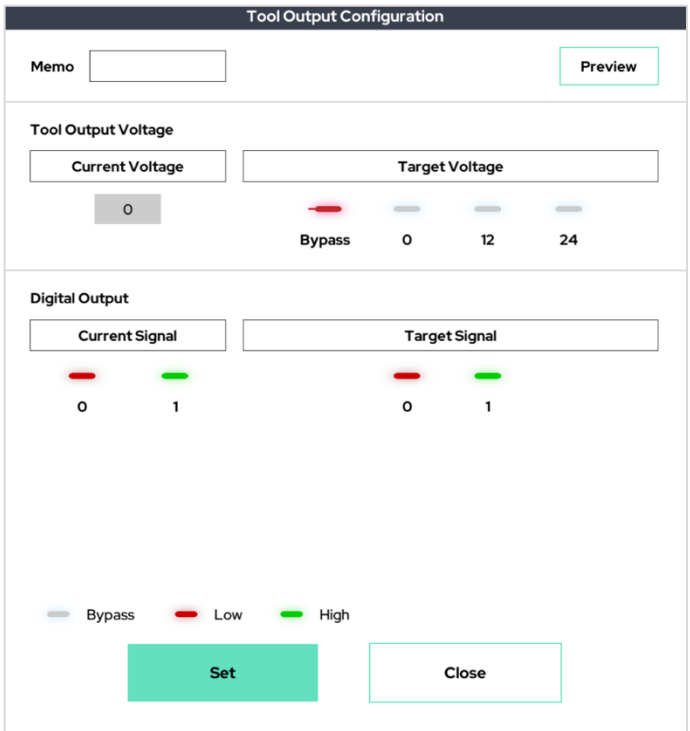
③ Dies ist eine Funktion zum Drucken der unter Punkt 2 ausgewählten Einstellungen als Vorschau.

④ Stellen Sie die Werkzeugfunktion mit der in Punkt 2 ausgewählten Einstellung ein.

© RAINBOW ROBOTICS Inc. All rights reserved.



Wenn die Steuerbox und der Tablet-PC verbunden sind, stellen Sie das Zielsignalfeld wie in der Abbildung oben gezeigt ein und drücken Sie die Vorschau-Taste, um ein Ausgangssignal von dem Werkzeugflansch auszugeben, wie unten gezeigt.



Das Feld ‘Current Signal’ ist auf der linken Seite des Bildes geändert.

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

Vorsicht:

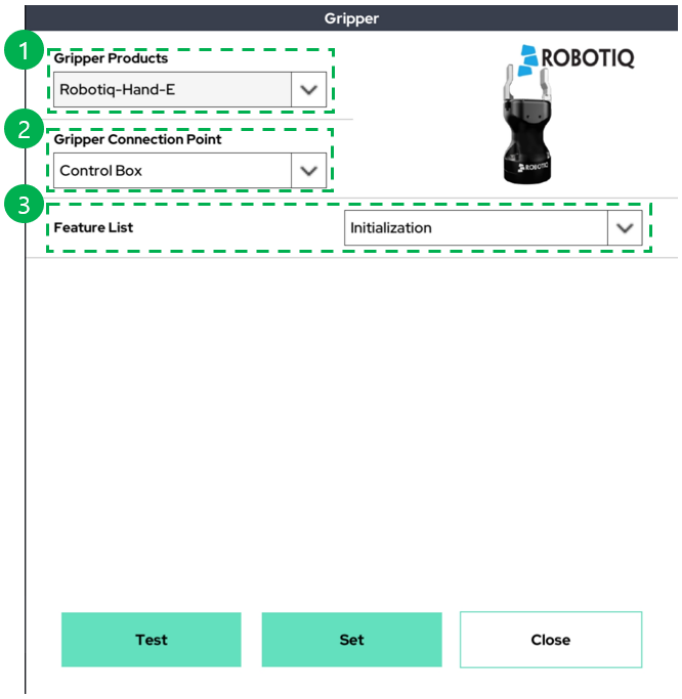
1) Sie können eine Notiz über die Einstellung hinterlassen, indem Sie die Memo—Funktion in der oberen rechten Ecke des Tool.out-Fensters verwenden.
2) Bevor Sie den Werkzeugflanschausgang verwenden, machen Sie sich bitte mit den elektrischen Eigenschaften des vom Hersteller bereitgestellten Anschlusses vertraut.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Greifer(Gripper)-Funktion :

Dedizierte Funktion für Greifer des kollaborativen Roboters. Kollaborative Robotergreifer verschiedener Firmen, wie beispielsweise von Robotiq, können bequem getestet, genutzt und in das Programm eingefügt werden. Es ist keine einfache Ein/Aus-Weise, sondern eine Funktion, die dem Benutzer bequem hilft, den Greifer zu verwenden, der umständlich zu programmieren und zu schreiben ist, wie die Verwendung einer seriellen Kommunikation wie RS485 oder die Verwendung von CRC.

Fügen Sie die Greiferfunktion dem Programmbaum hinzu. Wenn Sie auf den hinzugefügten Greifer klicken, erscheint das Fenstert wie folgt.



① Wählen Sie aus dem Greifer-Sortiment des Herstellers Ihrer Wahl.

② Wählen Sie das Teil aus, an dem der Greifer angeschlossen ist (Steuerbox, Werkzeugflansch).

③ Wählen Sie die Funktion, die als Greifer verwendet werden sollte.

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht:

1) Die Liste der von der Greiferfunktion angebotenen Produkte wird durch Benutzerfeedback aktualisiert.

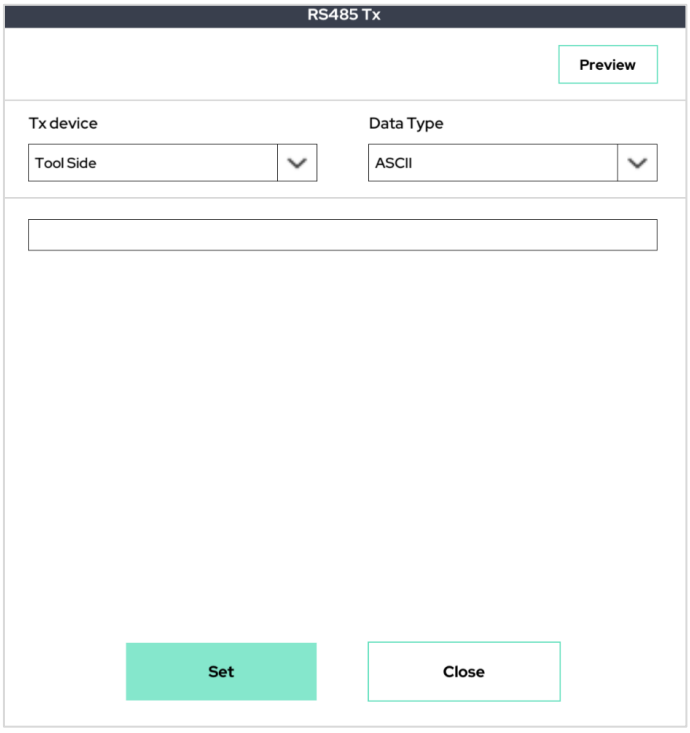
© RAINBOW ROBOTICS Inc. All rights reserved.

■ RS485-Funktion:

Funktion für RS485-Ausgang über Werkzeugflansch. Sie können mit der Funktion die Ausgabe im ASCII-Modus oder im Hex-Modus einstellen.

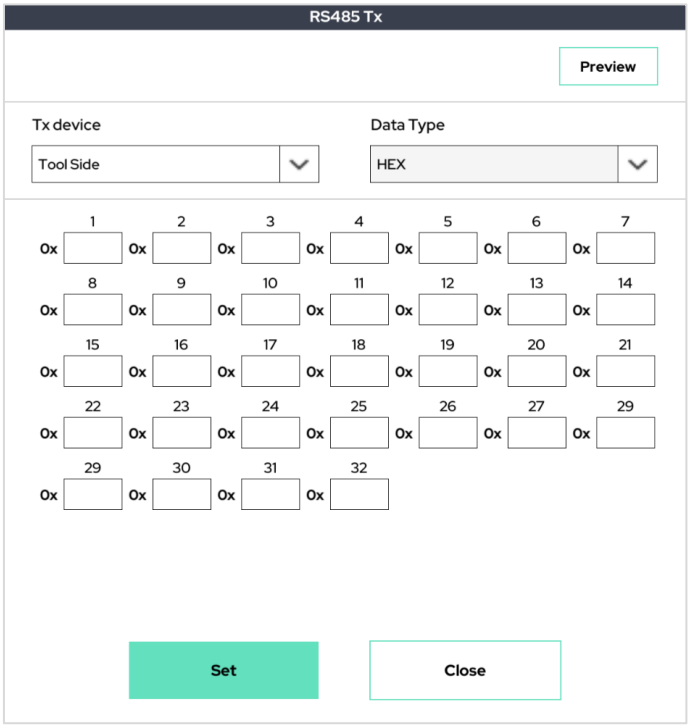
Nur RS485 Tx wird über das UI-Tablet unterstützt.

Die konfigurierte Kommunikation kann vorab über die Vorschau-Schaltfläche auf der rechten Seite des Popup-Fensters übertragen werden.



[ASCII-Modus]

© RAINBOW ROBOTICS Inc. All rights reserved.



[HEX-Modus]

Die Baudrate und andere Regeln (Parity bit, Stop bit), die bei RS485 verwendet werden sollen, werden im Setup-serial-Menü eingestellt. Alternativ kann es mit der Option Set-Serial_Configuration oben im Projekt festgelegt werden.

Um die serielle Kommunikation von der Boxseite (Box side) zu verwenden, können Sie ein handelsübliches USB-Serial-Gerät (RS232/422/485) verwenden, indem Sie es an den USB-Anschluss stecken.

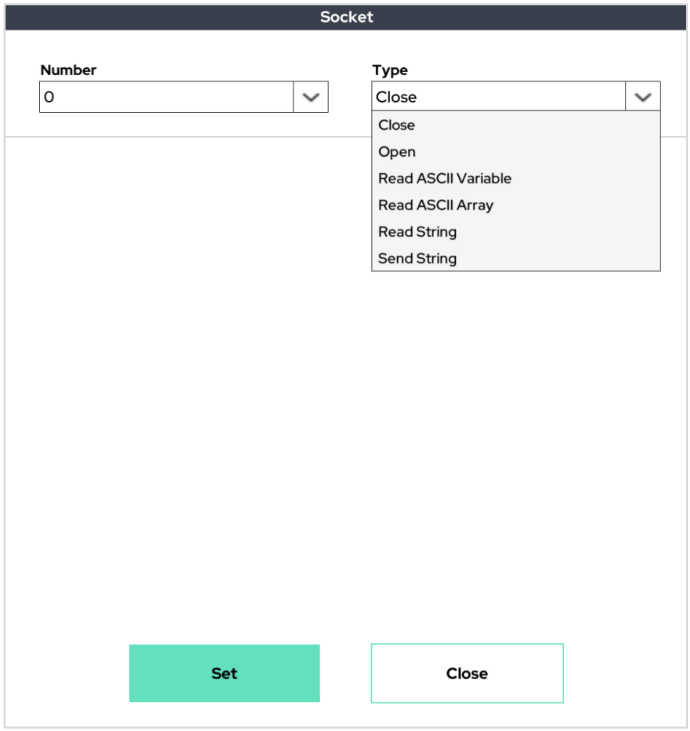
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Socket-Funktion:

Diese Funktion dient der Socket-Kommunikation. Bietet Funktionen zum Herstellen einer Verbindung zu einem bestimmten Server durch Öffnen eines Sockets, zum Senden einer Anforderungsnachricht an den Server und zum Empfangen bestimmter Daten vom Server. Die Socket-Kommunikation kann bis zu 5 separate Server verbinden.

Die IP-Einstellung der Steuerbox für die Socket-Kommunikation kann im Setup-Bildschirm unter System eingestellt werden.

In der Socket-Funktion werden 6 Funktionen wie folgt bereitgestellt.



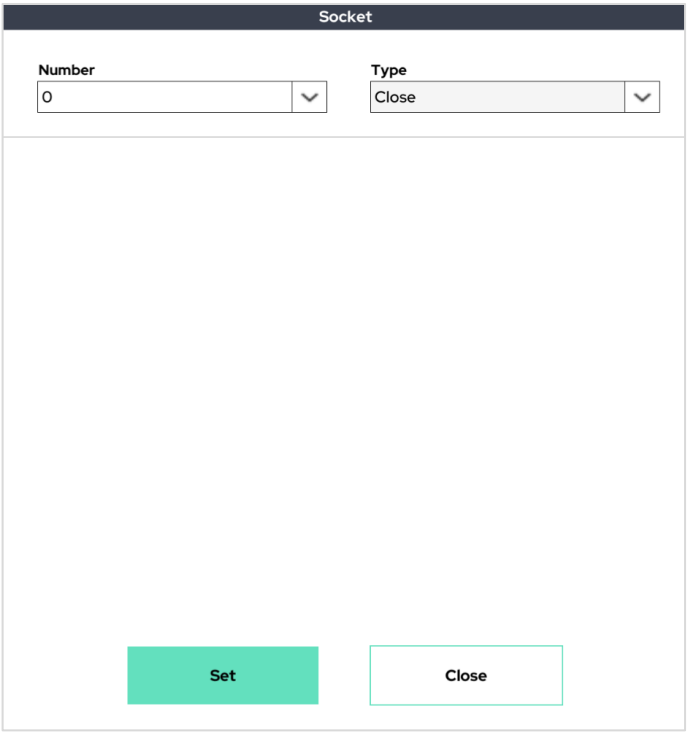
- Close: Der Socket wird geschlossen.
- Open: Der Socket wird geöffnet und mit dem Server verbunden.
- Read ASCII Variable: Zahlen aus einer Zeichenfolge vom Server werden gelesen und in eine Variable von Variable-Typ eingefügt.
- Read ASCII Array: Ein Array aus einer Zeichenfolge vom Server wird gelesen und in eine Variable vom Array-Typ eingefügt.

© RAINBOW ROBOTICS Inc. All rights reserved.

- Read String: Zeichenfolgen aus Server werden gelesen und in eine Variable vom String-Typ eingefügt.
- Send String: Die angegebene Zeichenfolge wird an den Server gesendet

© RAINBOW ROBOTICS Inc. All rights reserved.

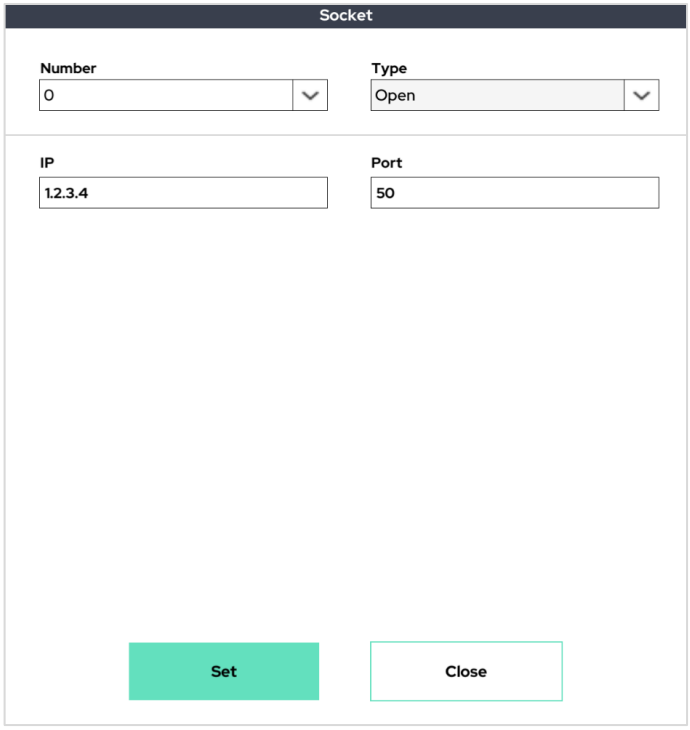
Socket-Funktion: Close



Mit dieser Funktion wird die Socket-Funktion der ausgewählten Nummer geschlossen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Socket-Funktion: Open

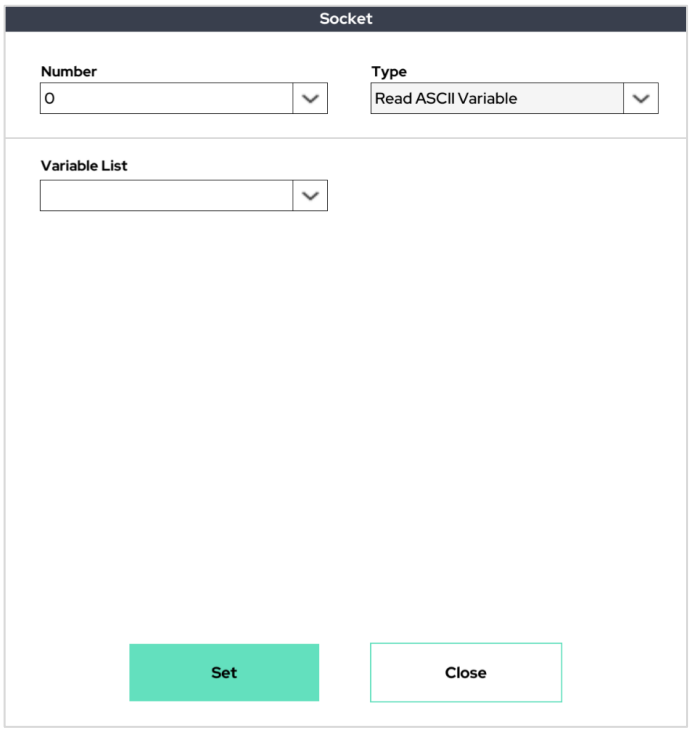


Mit dieser Funktion wird die Socket-Funktion der ausgewählten Nummer aktiviert und mit dem anderen Server verbunden.

Die IP-Adresse und die Anschlussnummer des Servers werden festgelegt, zu dem Sie eine Verbindung herstellen möchten.

© RAINBOW ROBOTICS Inc. All rights reserved.

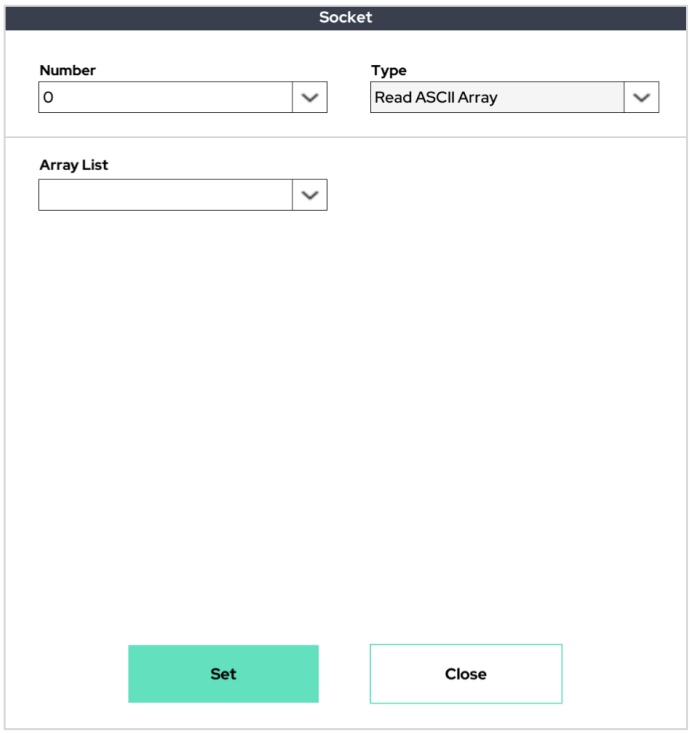
Socket-Funktion: Read ASCII Variable



Mit dieser Funktion wird eine der durch ‘Assign’ usw. deklarierten Variable-Typ-Variablen ausgewählt, und die vom Server empfangene Zahlen werden gelesen und in die zuvor deklarierte Variable-Typ-Variable eingefügt.

© RAINBOW ROBOTICS Inc. All rights reserved.

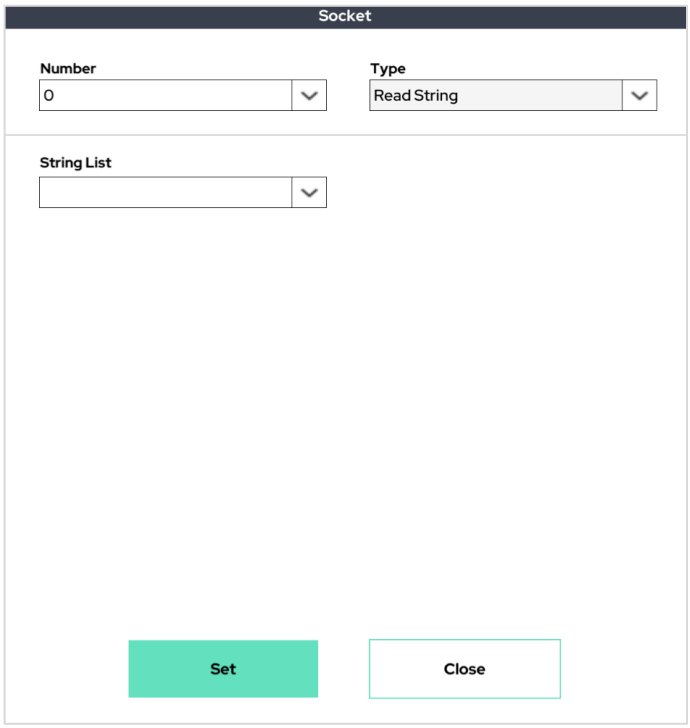
Socket-Funktion: Read ASCII Array



Mit dieser Funktion wird eine der durch ‘Assign’ usw. deklarierten Array-Typ-Variablen ausgewählt, und der vom Server empfangene Array wird gelesen und in die zuvor deklarierte Array-Typ-Variable eingefügt.

© RAINBOW ROBOTICS Inc. All rights reserved.

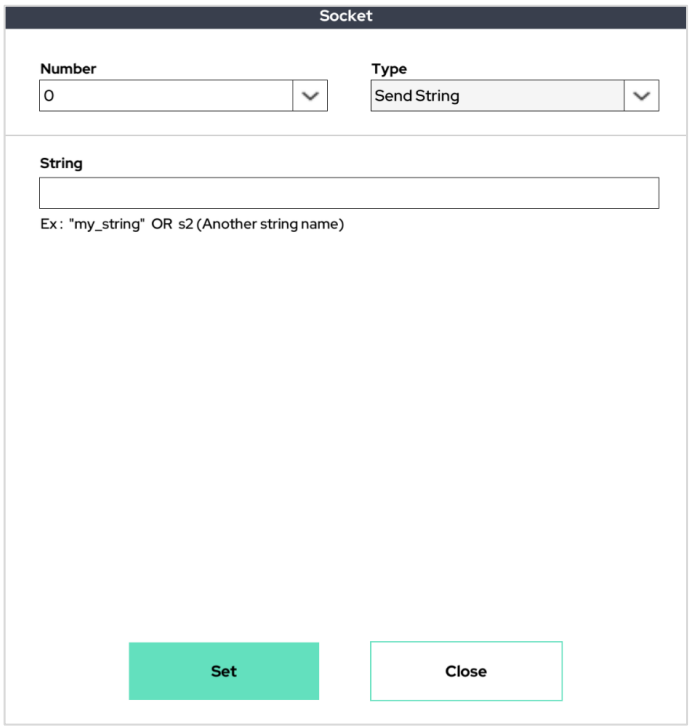
Socket-Funktion: Read String



Mit dieser Funktion wird eine der durch ‘Assign’ usw. deklarierten String-Typ-Variablen ausgewählt, und der vom Server empfangene String wird gelesen und in die zuvor deklarierte String-Typ-Variable eingefügt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Socket-Funktion: Send String



Mit dieser Funktion wird eine bestimmte Zeichenfolge an den Server gesendet. Sie können direkt einen String eingeben (z. B. “req_data”) oder die zuvor deklarierte Variable vom Typ String senden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht:

Die Syntax, die befolgt werden muss:

Um die vom Roboterhersteller bereitgestellten Optionen ‚Read ASCII Variable', ‚Read ASCII Array' und ‚Read String'-Funktion zu verwenden. MUSS das vom Server empfangene Datenformat dem folgenden Format entsprechen. Wenn eine spezielle Kommunikationsgrammatik/Syntax erforderlich ist, wenden Sie sich bitte an den Hersteller.

Read ASCII Variable

Beim Empfang eines Wertes vom Server muss der Wert als Zahlenwert gesendet werden. (d.h. der Zahlenwert muss nicht in Anführungszeichen stehen.)

(e.g. 123, 4567)

Read ASCII Array

Beim Empfang eines Arrays vom Server muss das Array nicht in Anführungszeichen eingeschlossen werden. In diesem Fall müssen geschweifte Klammern und Kommas zwischen jeweiligen Zahlenwerten stehen. (d.h. Senden Sie das Array nicht als Zeichenfolge, indem Sie das Array zwischen Anführungszeichen schreiben.)

(e.g. {100, 200, 300})

Read String

Der Server sollte den String mit Anführungszeichen im Format “this_is_string_from_server” senden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Interne Variablen zum Aktivieren der Socket-Kommunikation:

Interne Variablen existieren wie folgt, um zu wissen, ob die Socket-Kommunikation verbunden ist oder nicht. Beide internen Variablen unten sind wählbar und im ‚Shared Date'-Typ verfügbar.

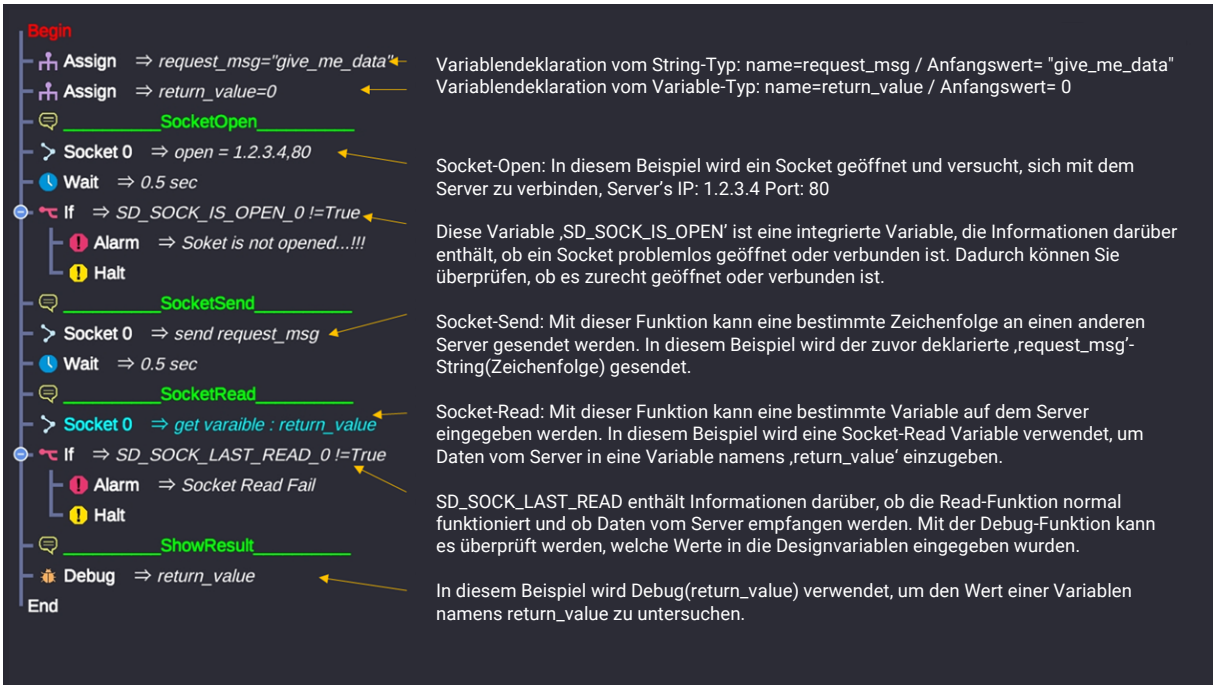
SD_SOCK_IS_OPEN_# (Wobei mit # Socket-Nummer 0-4 bezeichnet)

Dies ist eine Variable, die speichert, ob der Socket der entsprechenden Nummer normal geöffnet oder normal mit dem Server verbunden ist. Nach dem Socket-Open können sie wie bei If(SD_SOCK_IS_OPEN_0) prüfen, ob der Socket richtig verbunden ist.

SD_SOCK_LAST_READ_# (Wobei mit # Socket-Nummer 0-4 bezeichnet)

Dies ist eine Variable, die speichert, ob die Read-Funktion normalerweise mit dem Socket der entsprechenden Nummer ausgeführt wurde. Beispielsweise wird nach Socket-ReadAsciiVariable If(SD_SOCK_LAST_READ_0) verwendet, um zu prüfen, ob die letzte Lesefunktion normal ausgeführt wurde. Wenn keine Daten vom Server kommen, hat diese Variable den Wert 0.

Die folgende Abbildung zeigt ein Beispiel der Socket-Funktion.



© RAINBOW ROBOTICS Inc. All rights reserved.

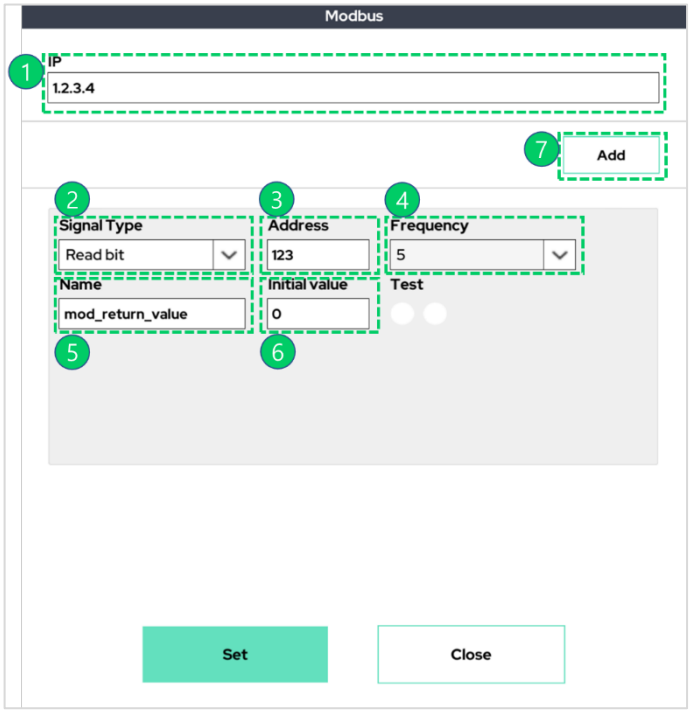
■ Modbus TCP(Client)-Funktion:

Diese Funktion bietet die Möglichkeit, Daten von einer bestimmten IP/Adresse anzufordern und zu empfangen. Häufigkeit und Format der Datenanforderung können angegeben werden.

Die Portnummer für Modbus TCP ist fest auf 502 (Modbus-Standard) festgelegt.

Die mit Modbus TCP-Servern verbundenen Protokolle und Formate sind im Anhang aufgeführt.

Die Modbus-TCP-Client-Funktion muss oben im Pre.P-Programm unter hinzugefügt werden.



① Geben Sie die IP-Adresse des Servers ein.

② Wählen Sie einen Signaltyp von ‚Read bit (1bit)', ‚Read word (16bit)', ‚Write bit (1bit)', ‚Write word (16bit)'.

③ Geben sie die Adresse der Endpunktverbindung auf dem Server ein.

④ Wählen Sie die Häufigkeit der 'Read/Write'-Anfragen pro Minute (Hz)

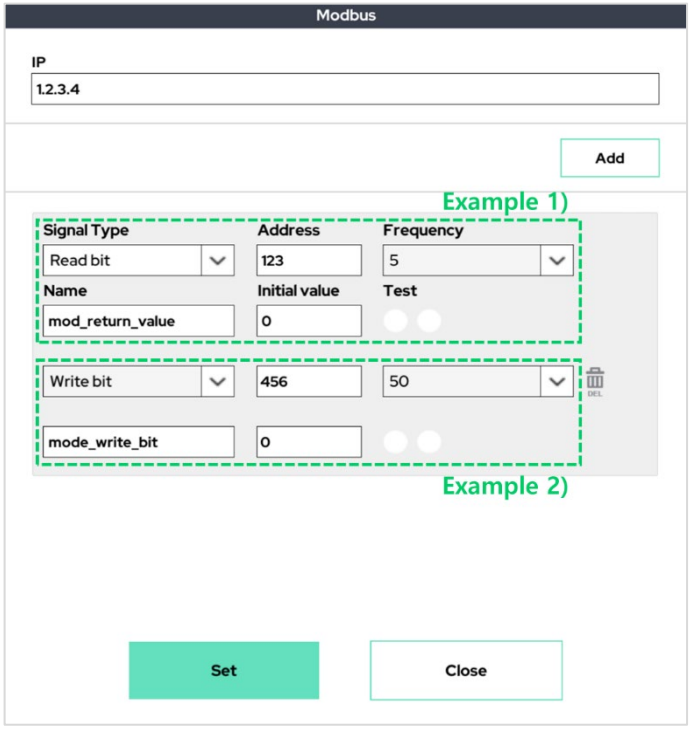
© RAINBOW ROBOTICS Inc. All rights reserved.

⑤ Enthält bei Verwendung einer ‘Read’-Methode den Variablennamen zum Speichern des gelesenen Werts. Wenn Sie eine ‚Write’-Methode verwenden, legen Sie den Variablennamen für die Ausgabe fest.

⑥ Angangswert der in Schritt 5 gesetzten Variablen

⑦ Schaltfläche zum Hinzufügen des Signals

Nachfolgend finden Sie ein Beispiel für die Modbus-Funktionseinstellungen.



Bsp. 1 Interpretation)

Eine Funktion zum Lesen der Worttypinformationen in der Adresse 123 des Servers (IP: 1.2.3.4) und zum Einfügen in eine Variable namens ‘mod_return_value’. Die Periode zum Lesen von Informationen beträgt 5 Hz.

Bsp. 2 Interpretation)

Eine Funktion zum Schreiben des in mod_write_bit gespeicherten Werts an die Adresse 456 des Servers (IP: 1.2.3.4). Überträgt den Wert von mod_write_bit in einem Takt von 50 Hz an den Server.

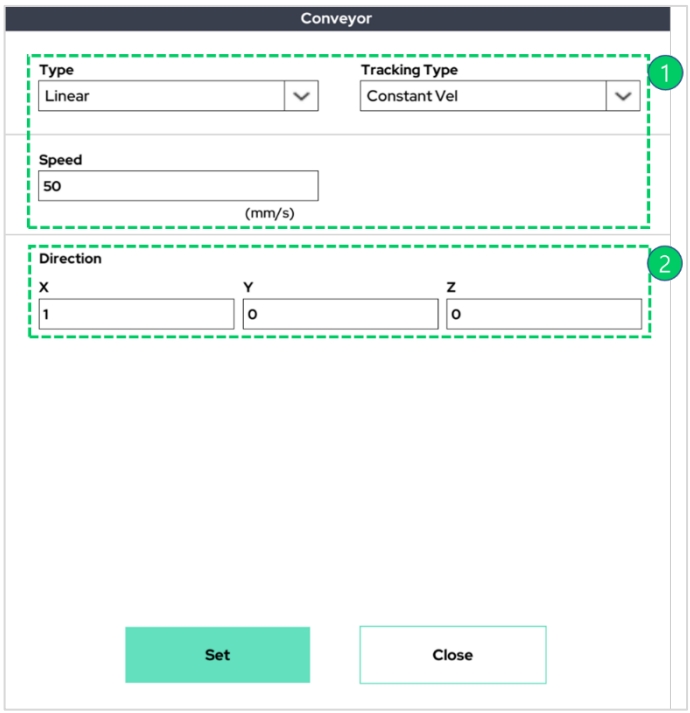
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Conveyor-Funktion:

Förderer-Tracking-Funktion. Nach dem Einstellen der Fördergeschwindigkeit und -richtung werden die Unteraktionen der Förderfunktion (MoveL, MoveLB, Circle usw.) dem Förderfluss hinzugefügt.

Gemeinsame Bewegung (MoveJ, MoveJB usw.) kann nicht als Unterelement von Conveyor verwendet werden. Nur MoveL, MoveLB, MovePB, MoveITPL, Circle werden unterstützt.

Fügen Sie die Förderfunktion zum Programmbaum hinzu und klicken Sie auf die Funktion, um die Optionen anzuzeigen.

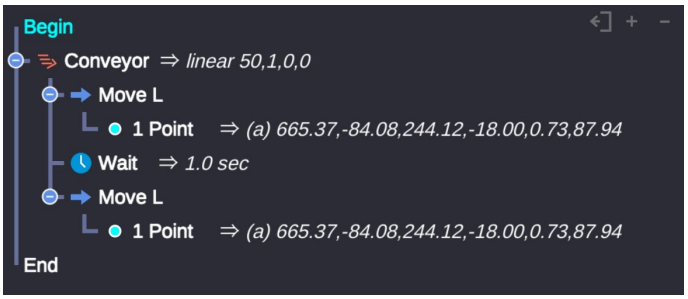


① Stellen Sie den Bewegungstyp und die Geschwindigkeit des Förderers ein.

② Legen Sie die Richtung für die Förderbandbewegung fest (x-, y-, z-Wert basiert auf dem Basiskoordinatensystem des Roboterarms). Der Roboter bewegt sich mit der angegebenen Geschwindigkeit in die angegebene Richtung, bis die Förderbandbewegung endet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Darauf basierend wird das Programm wie folgt konfiguriert.



© RAINBOW ROBOTICS Inc. All rights reserved.

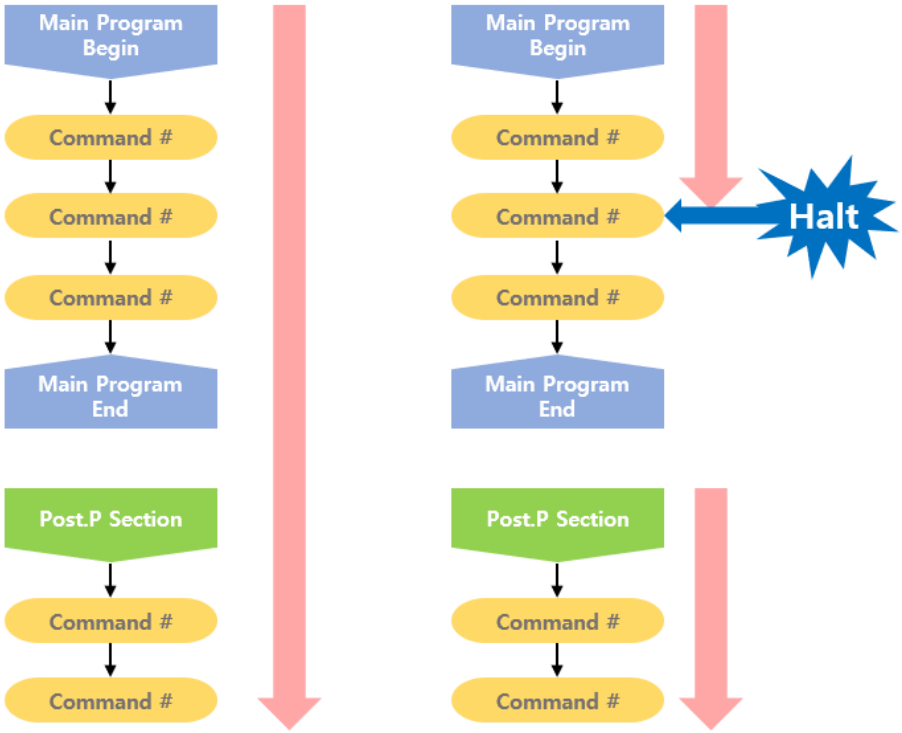
■ Post.P(Post Program)-Funktion :

Die Funktion definieren, die als abschließende Aktion nach Programmende ausgeführt werden sollte.

Post Program-Funktion ist eine Funktion, um einen Befehl auszuführen, nachdem das Programm beendet ist.

Unter Post.P deklarierte Befehle werden nach Programmende sequenziell ausgeführt.

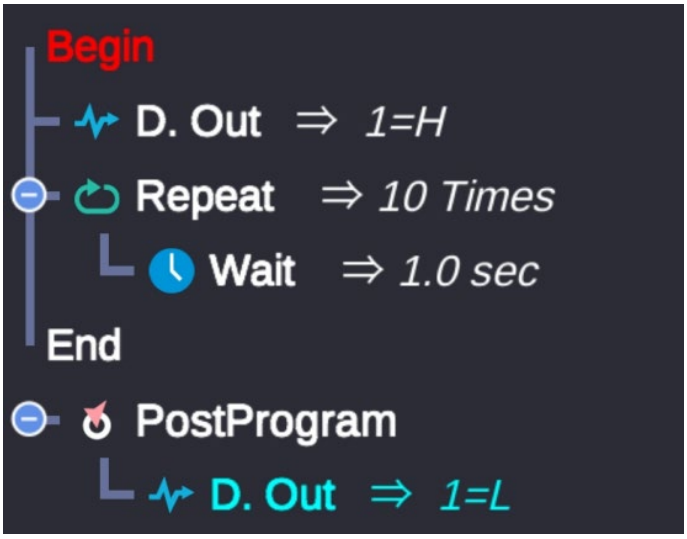
Die Ausführung von Post.P-Funktion verläuft wie im folgenden Diagramm dargestellt.



© RAINBOW ROBOTICS Inc. All rights reserved.

Post.P Bsp. 1)

Das folgende Beispiel zeigt ein Beispiel mit Post.P-Funktion. Gleichzeitig mit dem Programmstart (unter bei Begin) wird über die D.out-Funktion ein High-Signal an den Anschluss 1 gesendet. Obwohl die Einstellung nicht programmiert wird, wie D.out, Nr. 1 vor dem Ende des Programms (vor End) nicht ausgeschaltet wird, sendet Anschluss 1 automatisch ein Low-Signal, wenn das Programm endet. Das liegt daran, dass 'D.out 1=L' für die Post.P-Funktion verwendet wurde.

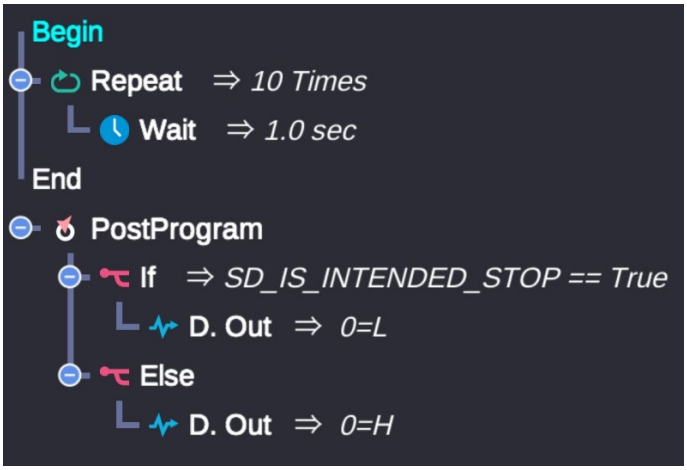


Wie im obigen Beispiel kann die Post.P-Funktion für Sicherheitsfunktionen usw. verwendet werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Post.P Bsp. 2)

Das folgende Beispiel zeigt ein Beispiel mit Post.P. Dies ist ein Beispiel dafür, dass die Warnleuchte (verbunden mit D.out, Nr. 0) nicht eingeschaltet wird, wenn das Programm normal endet, aber die Warnleuchte eingeschaltet wird, wenn es anormal endet.



SD_IS_INTENDED_STOP, das in diesem Beispiel verwendet wird, ist eine systeminterne Variable und wird immer auf 0 (falsch) initialisiert, wenn das Programm startet. Wenn das Programm normalerweise nach Benutzerabsicht gestoppt wird, ist diese Variable 1 (wahr). Wenn das Programm aus verschiedenen Gründen abnormal stoppt, bleibt diese Variable falsch.

Endsignale durch Benutzerauswahl, wie z. B. Beendigung durch Klicken auf die UI-Endschaltfläche, Beendigung durch Ein/Aus-Stoppsignal und andere Kommunikationsbeendigungssignale, werden als normale Beendigung bestimmt (SD_IS_INTENDED_STOP = wahr).

Wenn das Programm aufgrund eines Zugriffs auf einen singulären Punkt oder aufgrund eines Befehlssyntaxproblems beendet wird, wird eine nicht vom Benutzer beabsichtigte Beendigung als abnormal Beendigung bestimmt (SD_IS_INTENDED_STOP = falsch).

- Auch wenn das Programm nicht normal beendet wird (z. B. wenn Halt-Stop bei Alarm gedrückt wird), wird die im Post-Program definierte Funktion ausgeführt.
- Befehle, die sich auf Roboterarmbewegungen beziehen, wie MoveJ und MoveL usw. können nicht im unteren Teil von Post.P verwendet werden.
- Post.P-Funktion funktioniert nur in Programmen der obersten Ebene, Wenn die Post.P -Funktion in dem durch die Sub.P-Funktion aufgerufenen Unterprogramm verwendet wird, wird der relevante Teil nicht ausgeführt.

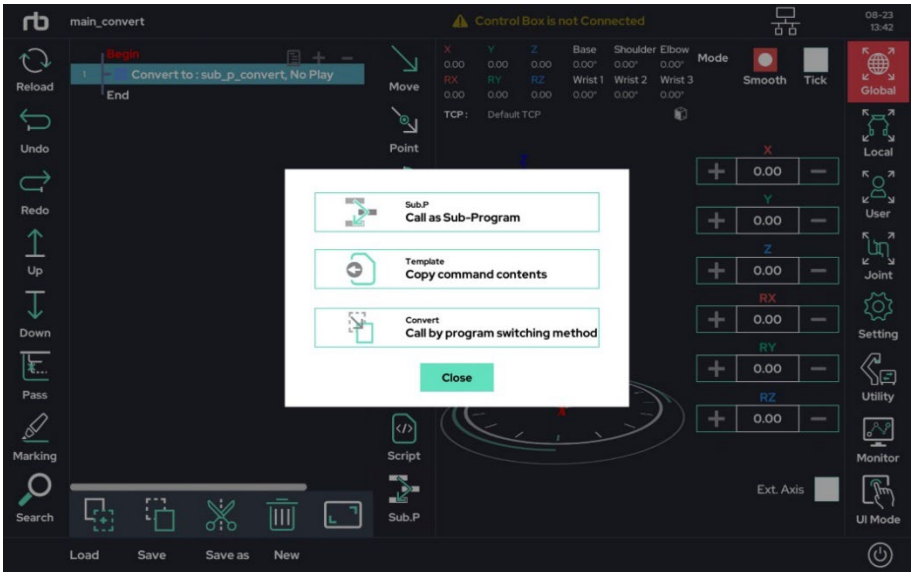
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Template-Funktion:

Dies ist eine Funktion, um andere vorgefertigte Programmdateien (Teaching-File) in bearbeitbarer Form in das aktuelle Dokument einzufügen.

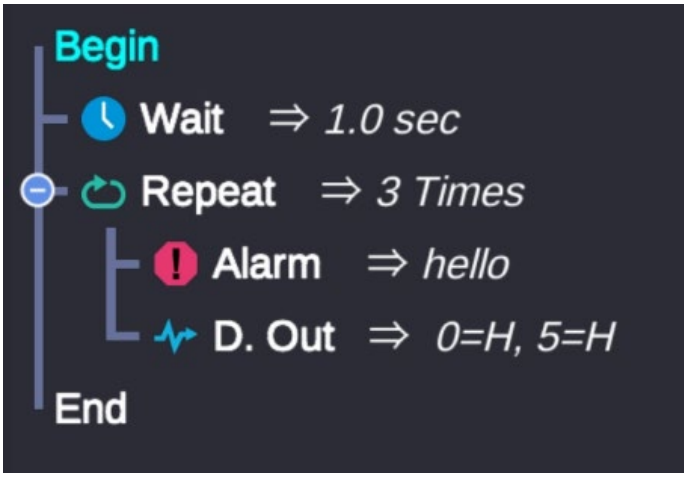
Die Funktion ähnelt der Funktion von Sub.P, aber Sub.P ist eine Funktion zum einfachen Aufrufen des entsprechenden Programms. Template-Funktion ist eine Funktion zum Kopieren und Einfügen eines anderen Programms in eine Form, die im aktuellen Programm geändert werden kann.

Wenn Sie im Programm auf das ‘Sub.P‘-Symbol klicken, wird das folgende Popup-Fenster angezeigt, und klicken Sie dabei auf Befehl ‚Copy Contents‘.

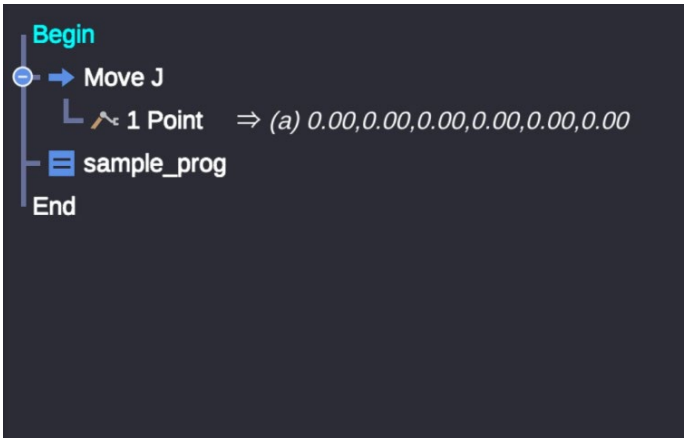


Angenommen, ein Projekt mit dem Namen “sample_prog” wird wie unten gezeigt erstellt.

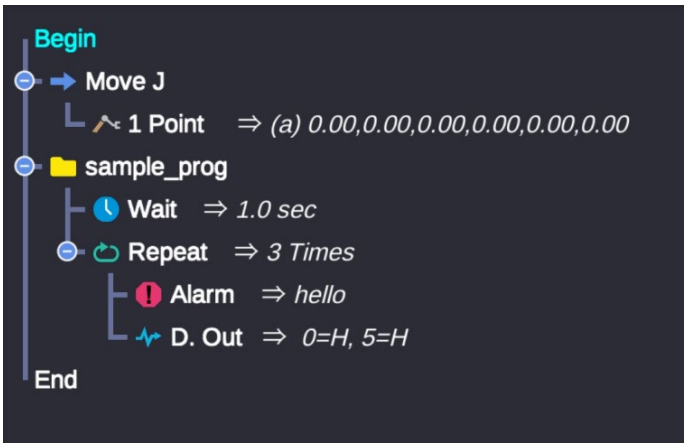
© RAINBOW ROBOTICS Inc. All rights reserved.



Bsp. 1) Wenn ‘sample_prog’ mit der ‚Sub.P’-Funktion aufgerufen wird



Bsp. 2) Wenn ‘sample_prog’ mit der Template-Funktion aufgerufen wird



© RAINBOW ROBOTICS Inc. All rights reserved.

Wie im Beispiel 1) gezeigt, wird beim Aufrufen von Sub.P das Projekt ausgeführt, aber eine Modifikation im oberen Programm ist nicht möglich. Auch wenn das importierte Unterprojekt geändert wird, wird auch der Betrieb des übergeordneten Programms geändert.

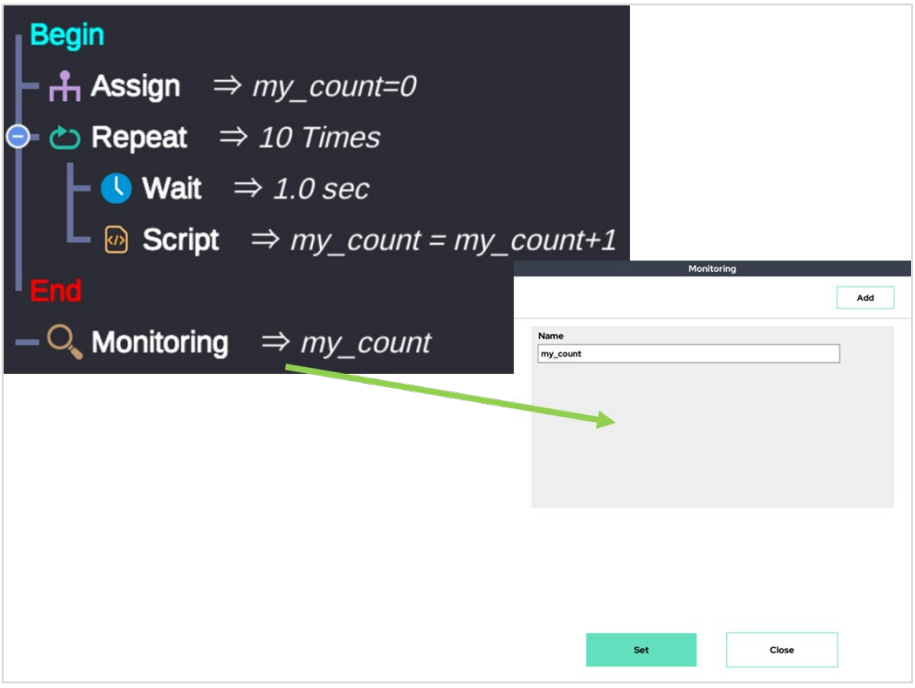
Wie im Beispiel 2) gezeigt, wird mit der Template-Funktion aufgerufen, so wird es in einer Form aufgerufen, die im oberen Programm modifiziert werden kann. Nach dem Kopieren in Template ändert sich der Inhalt nicht, selbst wenn das ursprüngliche Unterprogramm geändert wird.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Monitor-Funktion:

Dies ist eine Funktion zum Deklarieren von Variablen (Einzelvariable, Array, Punkt-Variable usw.), die in Echtzeit beobachtet werden sollen, während das Programm ausgeführt wird.

In der Monitor-Funktion deklarierte Variablen können in Echtzeit beobachtet werden, während das Programm läuft, indem Sie auf das Monitor-Symbol auf der rechten Seite der Make/Play-Seite klicken.



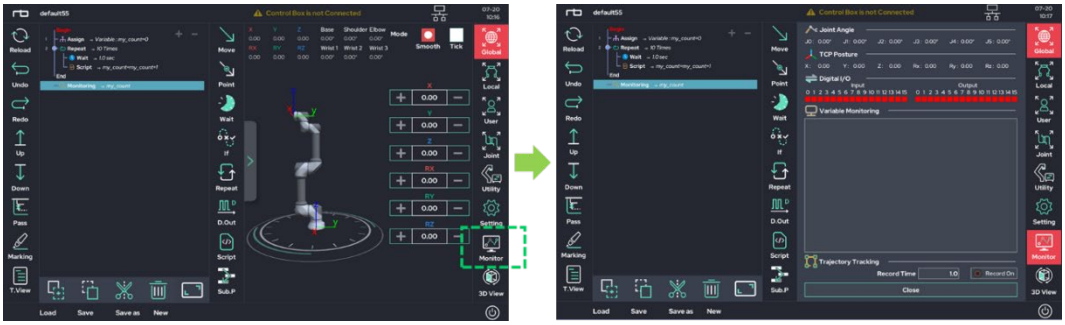
Im obigen Programmbeispiel wird eine Variable namens ‘my_count’ deklariert und eine Funktion zum Erhöhen von ‘my_count’ um 1 pro Sekunde wird durch eine Wiederholungsanweisung (Repeat) implementiert.

Und durch Verwendung der ‘Monitor’-Funktion wird ‚my_count’ als das zu beobachtende Objekt angegeben.

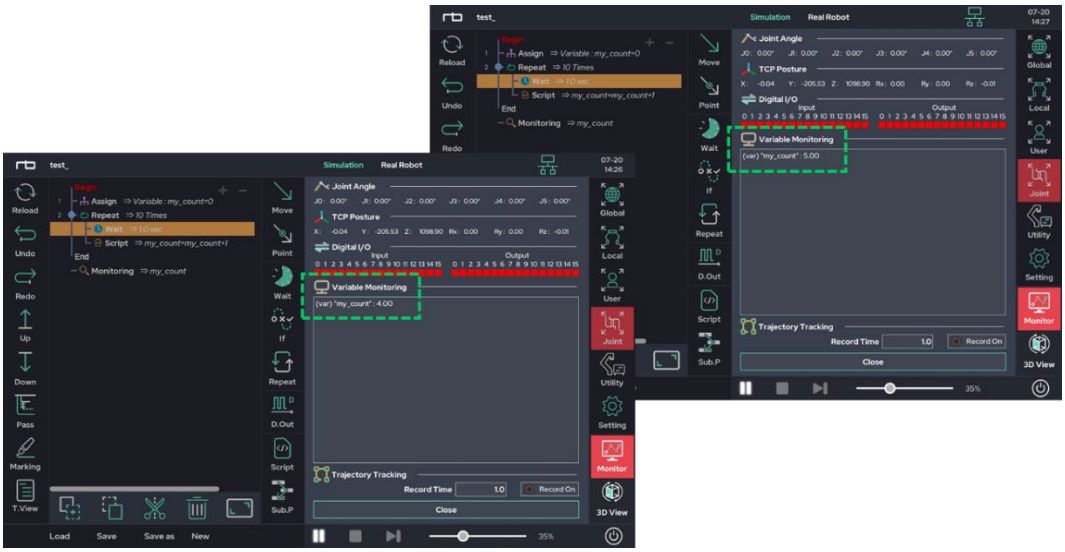
Sie können den Namen der Variablen, die Sie beobachten möchten, in der Monitor-Funktion eingeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

Wenn Sie den Wert der dem ‘Monitor‘ zugewiesenen Variable in Echtzeit beobachten möchten, während das Programm läuft, klicken Sie auf das Monitor-Symbol auf der rechten Seite des Bildschirms.



Wenn Sie danach das Programm wiedergeben (Run, ▷), können Sie in Echtzeit beobachten, wie der Wert von ‘my_count’ alle 1 Sekunde ansteigt.



© RAINBOW ROBOTICS Inc. All rights reserved.

■ Muster(Pattern)-Funktion:

Eine Funktion, die eine sich wiederholende Bewegung definiert.

Wenn Sie die Informationen über den Raum definieren, um die sich wiederholende Bewegung auszuführen, und die Bewegung definieren, die an jedem Ort ausgeführt werden sollte, wird dieselbe Bewegung an allen räumlichen Punkten ausgeführt.

Durch diese Funktion kann eine Palettierung (palletizing) realisiert werden.

Es gibt 3 Untereinstellungen.

- Pattern Property (Mustereigenschaft):

  Die Zielfläche/ den Zielraum, auf dem die Bewegung wiederholt werden sollte, definieren.

  Es werden verschiedene Formen unterstützt, wie gerade Linien, Ebenen, 3D-Würfel und beliebige Punkte .

- Pattern Anchor (Musteranker):

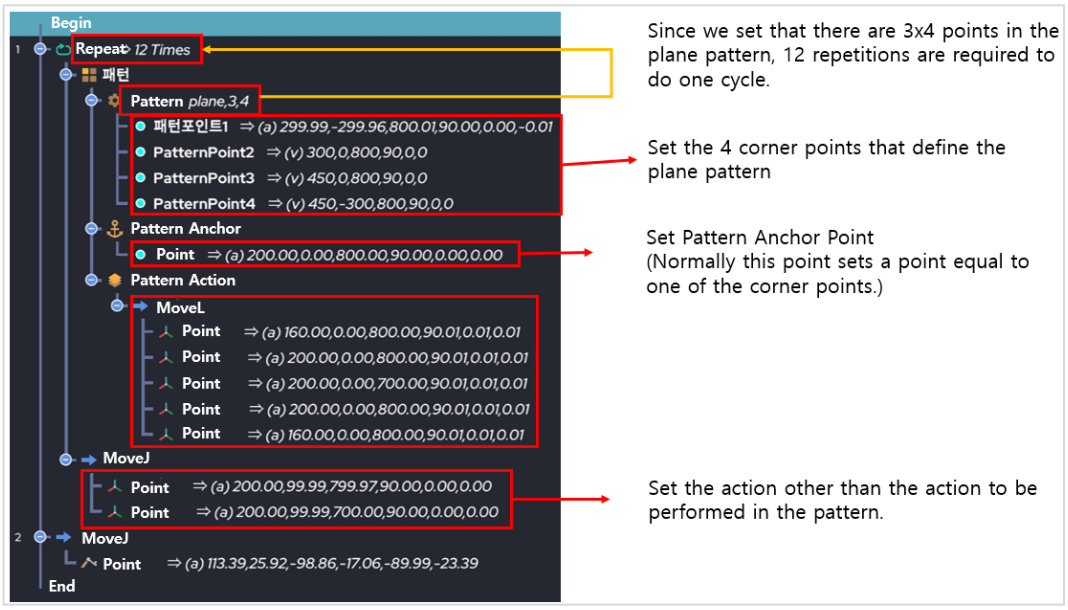
  Der Basispunkt der in der ‘Pattern Action’ zu definierenden Bewegung.

- Pattern Action (Musteraktion):

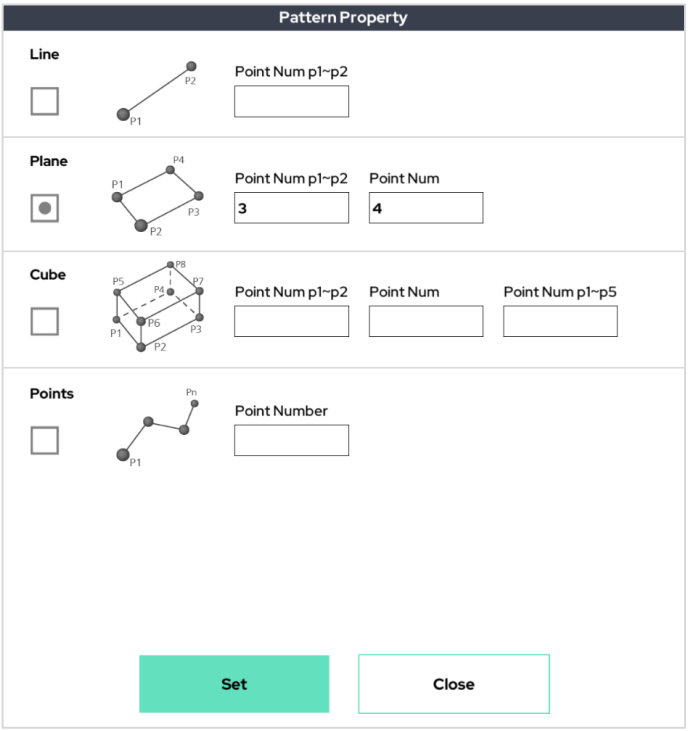
  Dies ist der Bereich, in dem die relative Bewegung in Bezug auf den in ‘Pattern Anchor’ festgelegten Referenzpunkt definiert wird. Die definierte relative Bewegung wird an jedem in der ‘Pattern Property’ festgelegten Musterpunkt wiederholt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Das Folgende ist ein Beispiel für die Pattern(Muster)-Funktion.

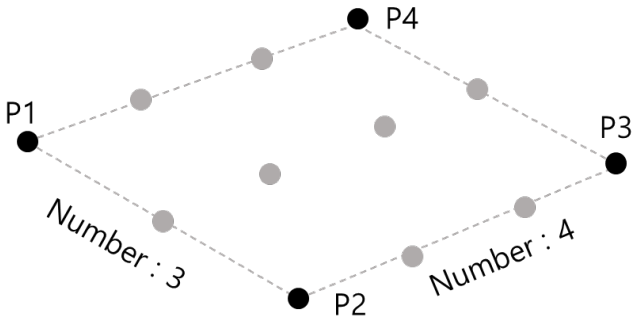


Step 1) Legen sie die ‘Pattern Property (Mustereigenschaft)’ wie unten gezeigt fest.

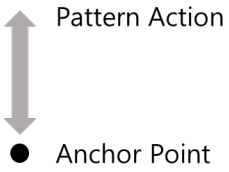


© RAINBOW ROBOTICS Inc. All rights reserved.

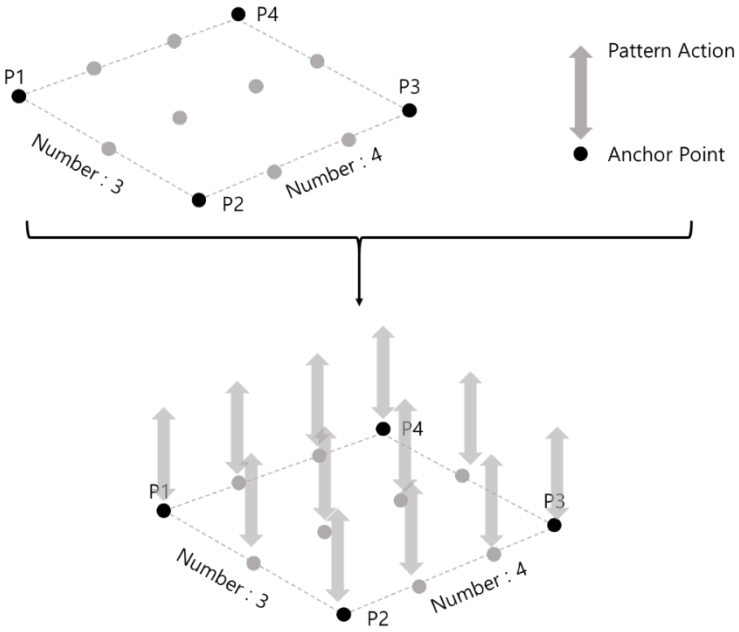
Mit dieser Einstellung werden die unteren und sich wiederholenden Punkte im Raum gebildet.



Step 2) Die folgenden relativen Bewegungen werden durch ‘Pattern Anchor’ und ‘Pattern Action’ definiert.

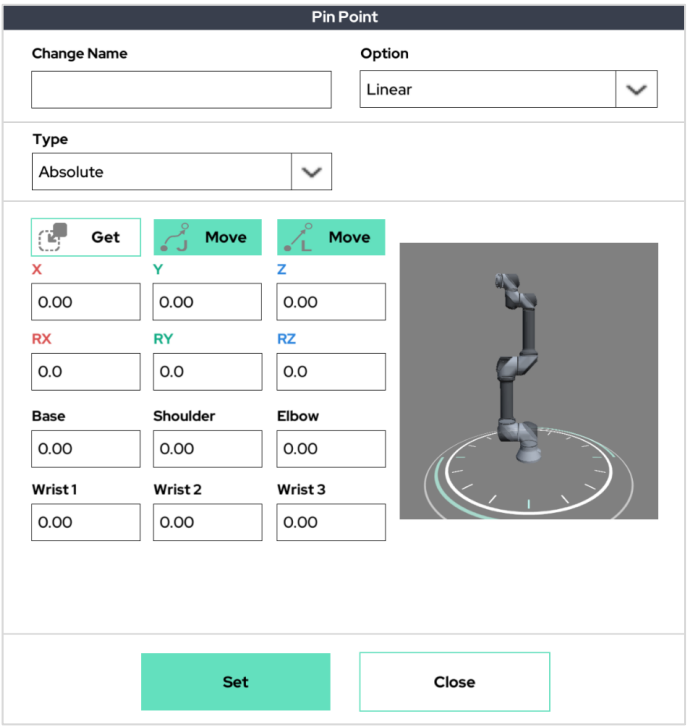


Step 3) Die sich wiederholende Bewegung wird vervollständigt, wenn die in Step-2 festgelegte relative Bewegung auf alle in Step-1 festgelegten Musterpunkte angewendet wird.



© RAINBOW ROBOTICS Inc. All rights reserved.

■ Pinpoint-Funktion :

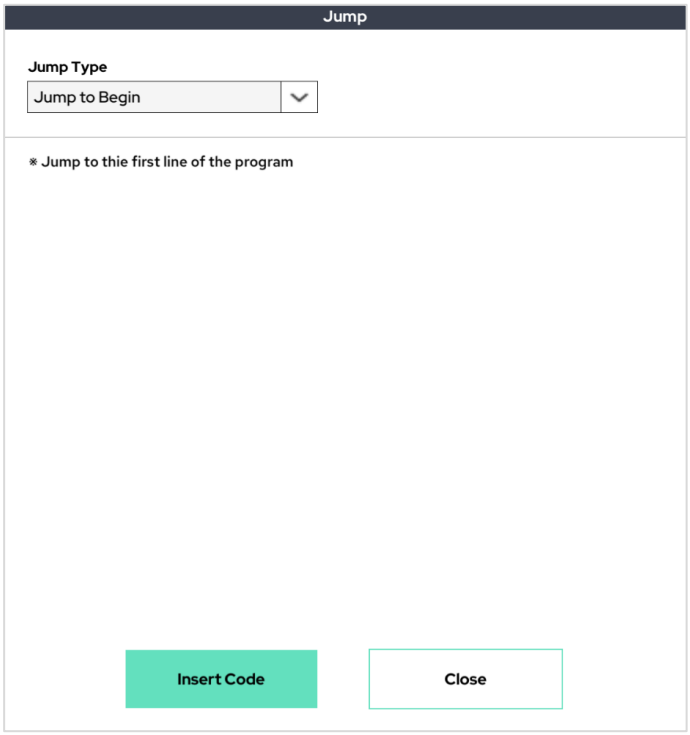


Eine spezielle Funktion nur zum Speichern von Haltungsinformationen. Dies ist eine Funktion, um die Informationen einer bestimmten Körperhaltung/ Position als ‚Point'-Variable zu speichern. Wenn Sie nach Teaching einer bestimmten Haltung einen PinPoint erstellen und den Namen des PinPoints angeben, wird die entsprechende Haltungsinformation in eine Variable in Form einer Point-Variablen umgewandelt.

Die als Point-Variable gespeicherte Informationen können in anderen Betriebsbefehlen/ Einstellungen usw. verwendet werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Jump (Überspringen)-Funktion:



Mit dieser Funktion kann der Programmablauf diskontinuierlich gesteuert werden. Sie können den Programmablauf durch mehrere Unteroptionen ändern.

Option) Jump to Begin

Der Programmablauf wird in die erste Zeile verschoben.

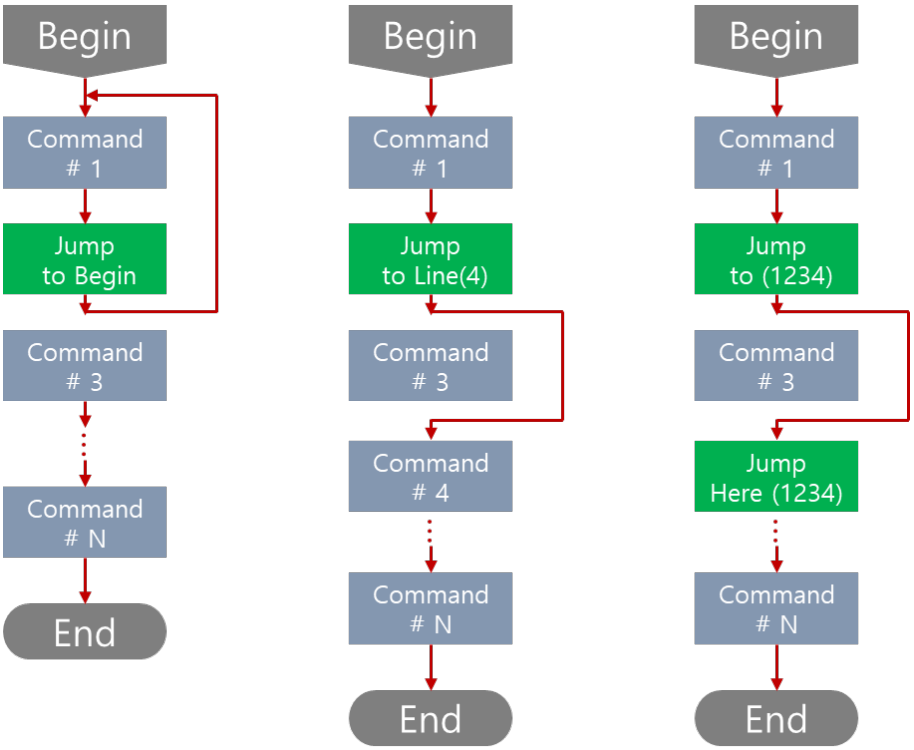
Option) Jump to Line

Der Programmablauf wird zu einer bestimmten Zeile verschoben.

Option) Jump To / Here

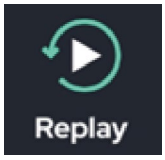
Der Programmablauf wird zu einem bestimmten Adresswert verschoben. Bei ‚JumpTo' wird der Adresswert aufgerufen und bei ‚JumpHere' sollte der Adresswert eingegeben werden.

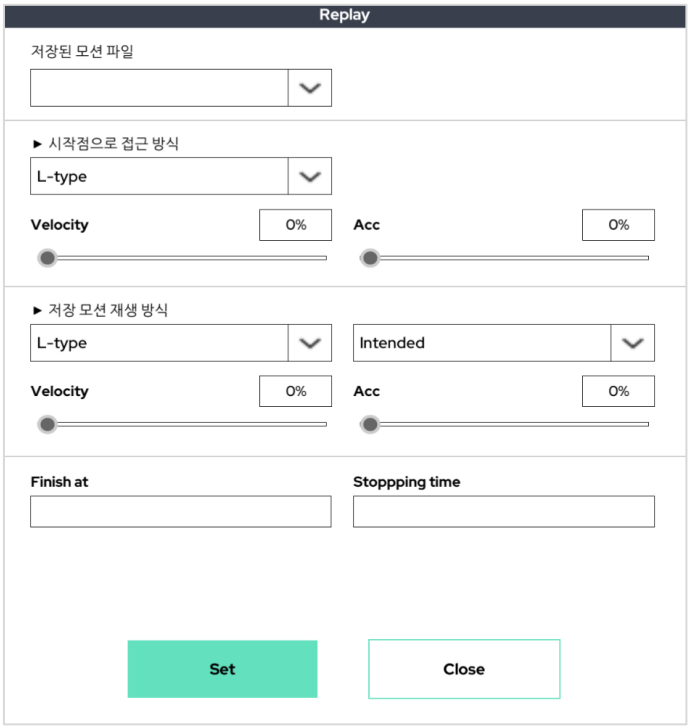
© RAINBOW ROBOTICS Inc. All rights reserved.



© RAINBOW ROBOTICS Inc. All rights reserved.

■ Replay-Funktion :



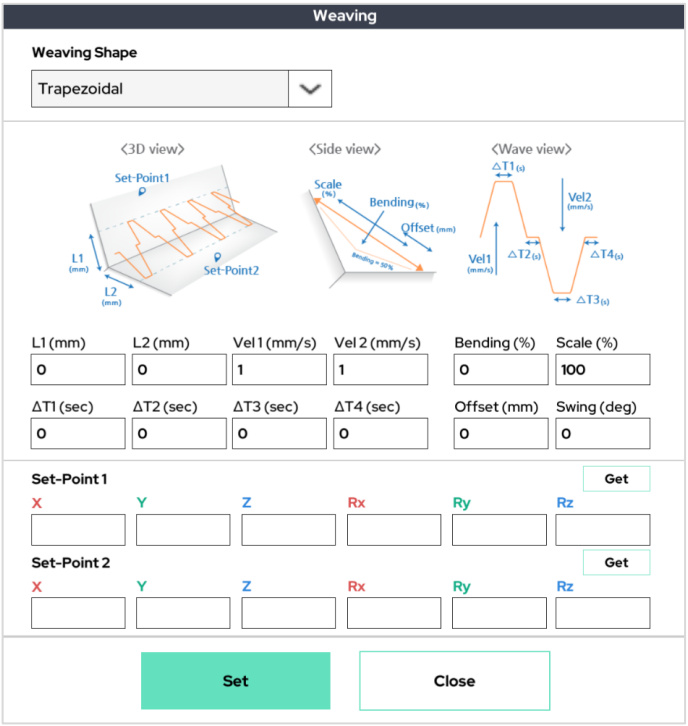


Diese Funktion dient zum Abspielen der aufgezeichneten Teaching-Bewegung. Die Bewegungsaufzeichnung wird in der Einstellung der Make-Seite durchgeführt. Wenn Sie den Namen und die Bewegungsgeschwindigkeit/Eigenschaft der aufgezeichneten Bewegung auswählen, wird die aufgezeichnete Bewegung erneut abgespielt.

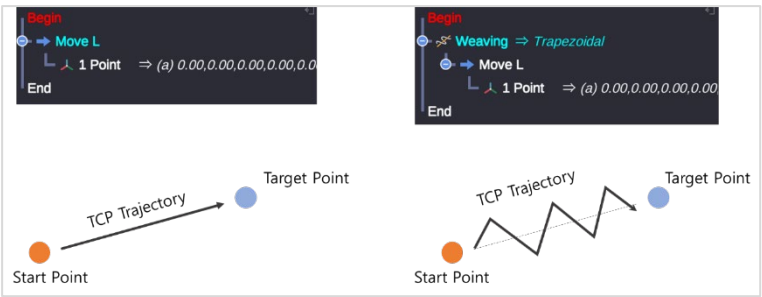
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Weaving-Funktion :

Dies ist eine spezielle Funktion zum Schweißen von Weaving (Weben). TCP-Trajektorien werden automatisch geändert, um die eingestellte Weaving-Aktion festzulegen. Gewünschte Webform und Weboptionen können einfach ausgewählt und eingegeben werden.



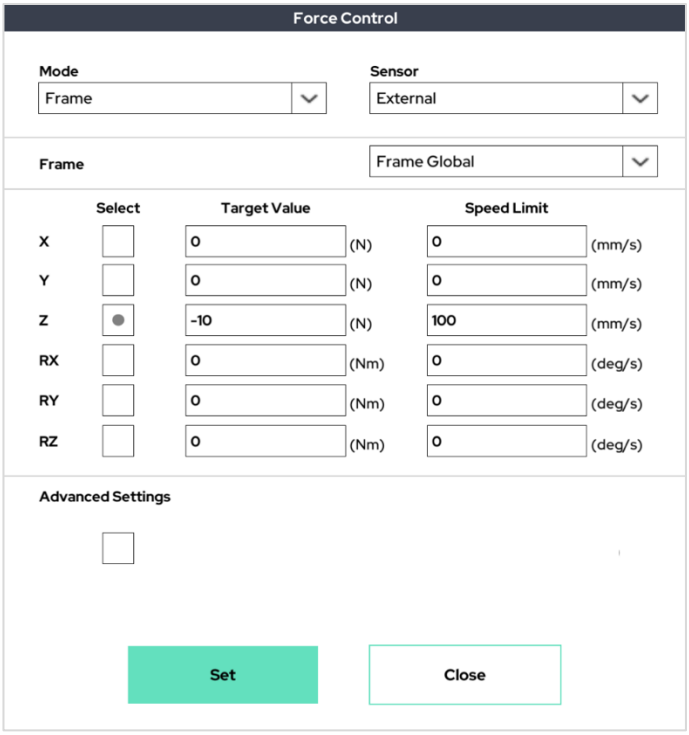
Die linke Abbildung unten gilt nur für den Normalbetrieb. Wenn diese Bewegung al sein Unterpunkt des Webens eingesetzt wird, wird eine TCP-Trajektorie gezeichnet, die Webbahn widerspiegelt (rechte Abbildung im Fall eines Dreieckswellenwebens).



© RAINBOW ROBOTICS Inc. All rights reserved.

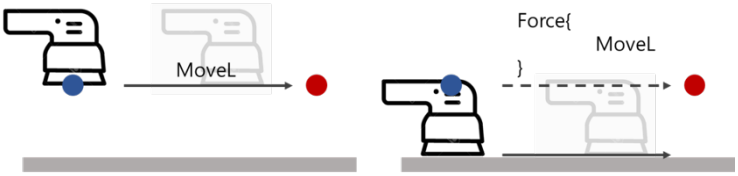
■ Force-Funktion :

Diese Funktion dient der Kraftregelung. Die Bewegungen unterhalb der Force-Control-Funktion ändern automatisch die Fahrbahn, um die eingestellte Kraft zu erreichen.



Wählen und geben Sie den gewünschten Kraftregelungsmodus, den für die Kraftregelung zu verwendenden Sensor und Zielwert der Kraftregelung ein.

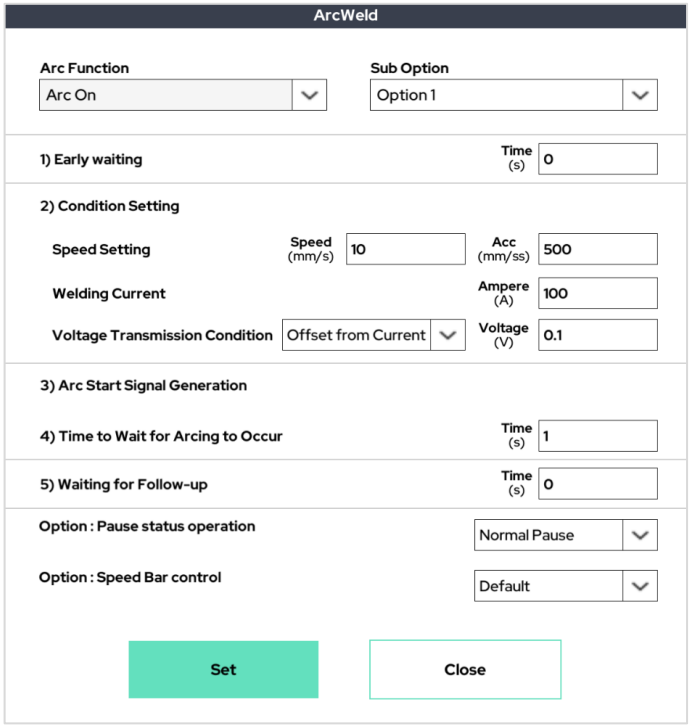
Die linke Abbildung unten gilt nur für den Normalbetrieb. Die Bewegung beginnt in der Luft und endet in der Luft. Wenn diese Bewegung als Unterpunkt der Kraftregelung eingesetzt wird, ändern sie sich darin, mit einer bestimmten Kraft auf den Boden zu drücken (Bei der Einstellung der Kraftregelung in Bodenrichtung).



[Vor der Kraftregelung] [Nach der Kraftregelung]

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Arcweld-Funktion:



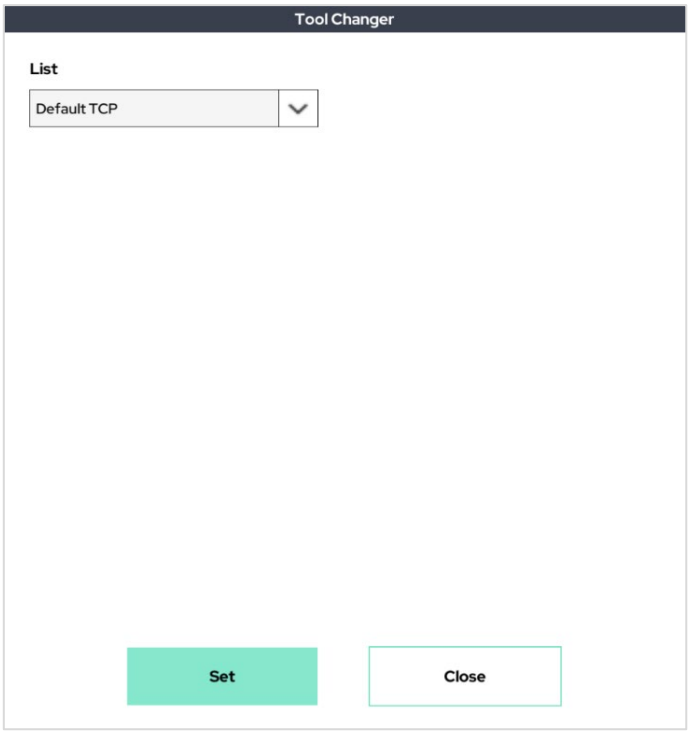
Dies ist eine spezielle Funktion für das Lichtbogenschweißen. Eine spezielle Makrofunktion, die entwickelt wurde, um schnell implementierbare Funktionen wie Wait / D.out zu aktivieren.

Um diese Funktion zu verwenden, muss im Voraus die Einstellung der Parameter(Variable) und Verbindungsinformation für das Schweißgerät im Device(Gerät)-Feld auf der Setup-Seite eingegeben werden.

Wie oben dargestellt, sind Schweißgeschwindigkeit/ Schweißstrom/ Spannungseinstellungen/ Sicherheitssignalverarbeitungsoption in diesem Programm schnell und einfach einzufügen.

© RAINBOW ROBOTICS Inc. All rights reserved.

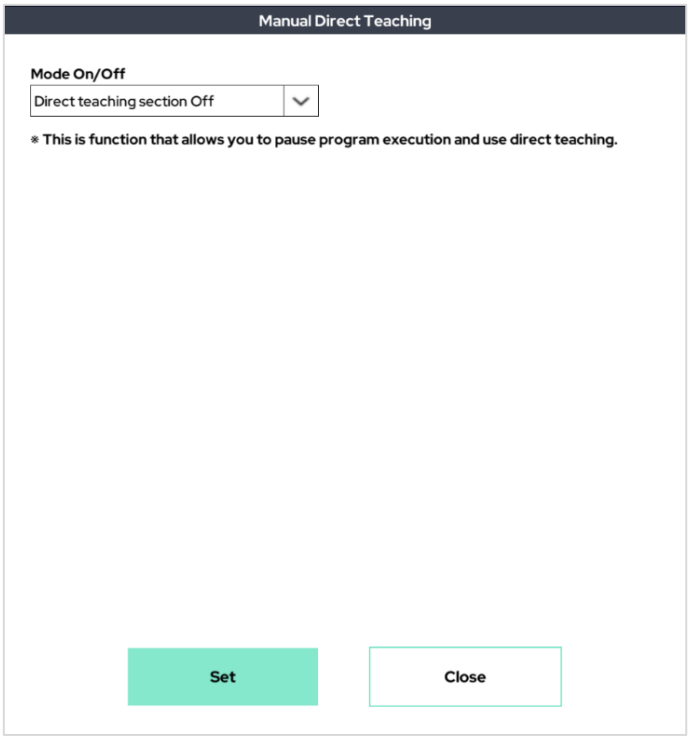
■ TCP-Einstellung:



Mit dieser Funktion wird TCP-Wert während der Programmausführung mit dem TPC-Wert geändert, der in der Werkzeugliste von Setup vorgespeichert ist. Er ändert sich erst wieder, wenn der TCP-Wert ersetzt oder das Programm beendet wird.

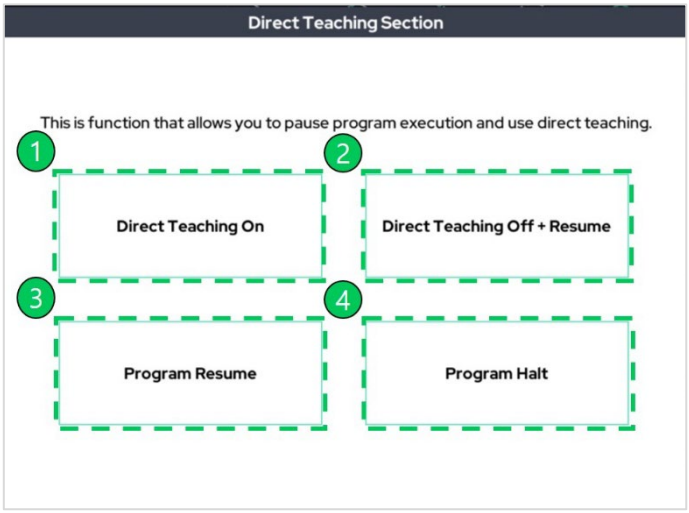
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Manuelle Betriebsfunktion :



Mit dieser Funktion ist Direkt-Teaching möglich während der Programmausführung. Bei ‘Modus On‘ wird das Programm angehalten, wenn der manuelle Bedienbefehl ausgeführt wird. Und ein Popup-Fenster erscheint auf dem Bildschirm wie unten gezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.



Im Popup-Fenster kann eine von vier Funktionen ausgewählt werden.

① Die Direkt-Teaching-Funktion wird verwendet, während das Programm angehalten ist.

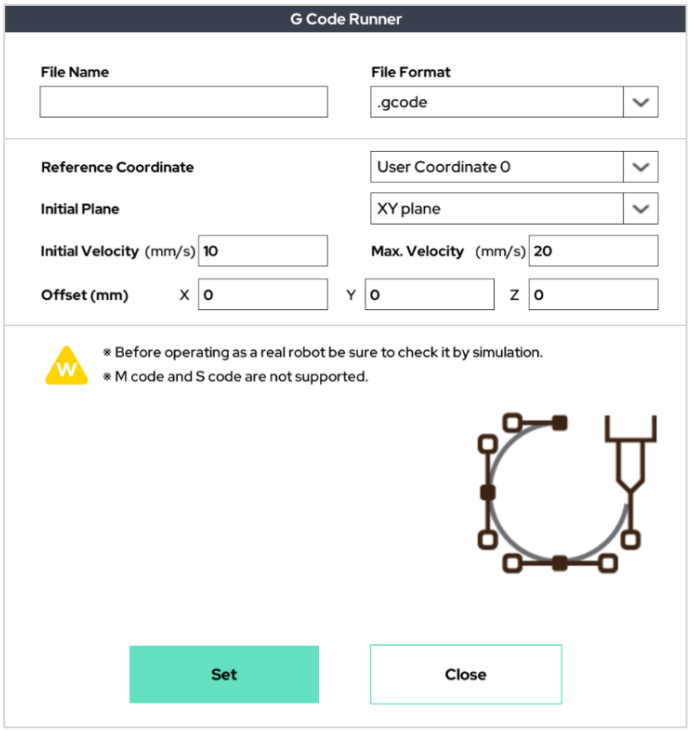
② Wenn die Direkt-Teaching-Funktion in ① verwendet wurde, sollte der Benutzer die Direkt-Teaching-Funktion ausschalten und dann das Programm fortsetzen.

③ Ignorieren Sie den manuellen Betrieb und setzen Sie das Programm fort.

④ Programm wird beendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ G -Code-Funktion :



Mit dieser Funktion wird der Roboter auf die im G-Code gespeicherte Bahn bewegt. Die G-Code-Datei muss vorab in einem Ordner im angegebenen Pfad (\Tablet\Android\data\com.rainbow.cobot\files\work) gespeichert werden, um verfügbar zu sein.

Geben Sie den Namen der G-Code-Datei ein, die in Dateinamen gespeichert wurde. Die Ebene, in der sich der Roboter bewegt, können dann die xy-, yz- und zx-Ebenen des benutzerdefinierten Koordinatensystems als Startebene angegeben werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Interface (Schnittstellenfunktion):

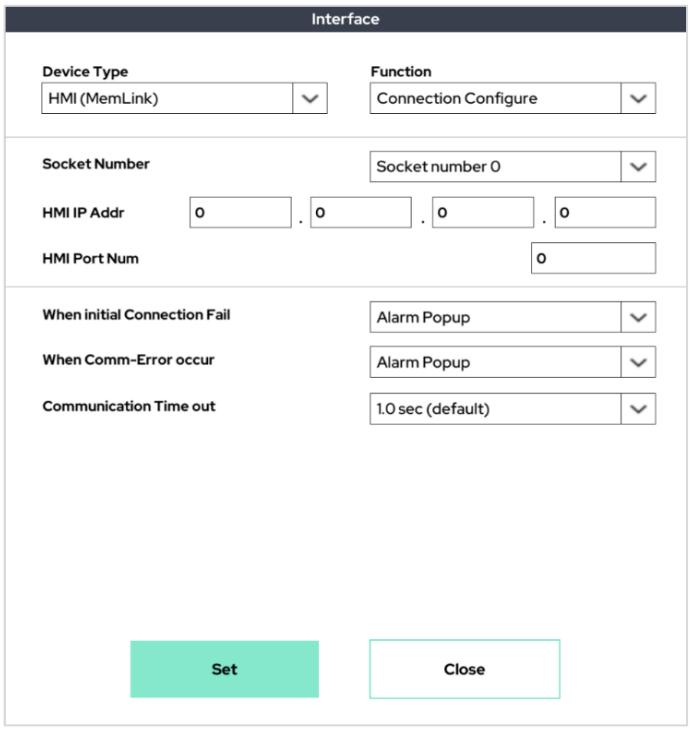
Die Schnittstellefunktion dient zum Verbinden externer Geräte wie PLC, HMI und PC mit der Steuerbox. Die Liste der externen Geräte, die über die Schnittstelle verwendet werden können, ist wie folgt.

- HMI (MemLink) - Proface, TOP
- PLC (MC-Protokoll) – Mitsubishi PLC
- Music Player
- PLC (XGT-Protokoll) - LS Electric PLC
- CSV File
- Pickit
- Modbus Client (Interrupt)

Da jedes externe Gerät über unterschiedliche detaillierte Funktionen verfügt, sollten Sie sich auf die folgenden Informationen beziehen.

© RAINBOW ROBOTICS Inc. All rights reserved.

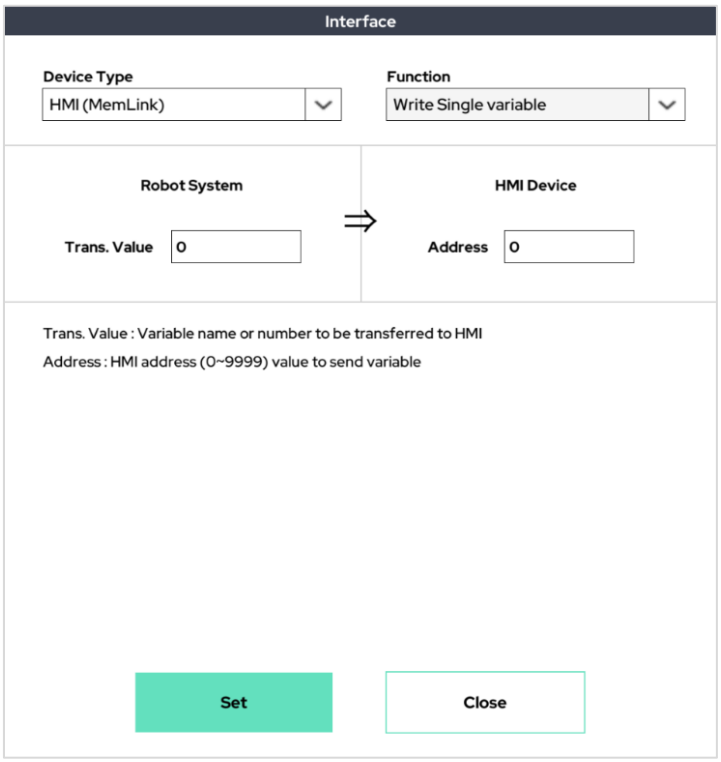
HMI (MemLink) – Verbindung konfigurieren



Mit dieser Funktion ist die Kommunikation zwischen HMI und dem RB-System zu verbinden. Socket-Nummer, IP-Adresse sollten eingegeben werden. Der Benutzer kann auch entscheiden, ob das Alarm-Popup im Falle, eines Verbindungsausfalls oder eines Kommunikationsfehlers aktiviert oder ignoriert werden sollte, und ein Kommunikations-Timeout festlegen.

© RAINBOW ROBOTICS Inc. All rights reserved.

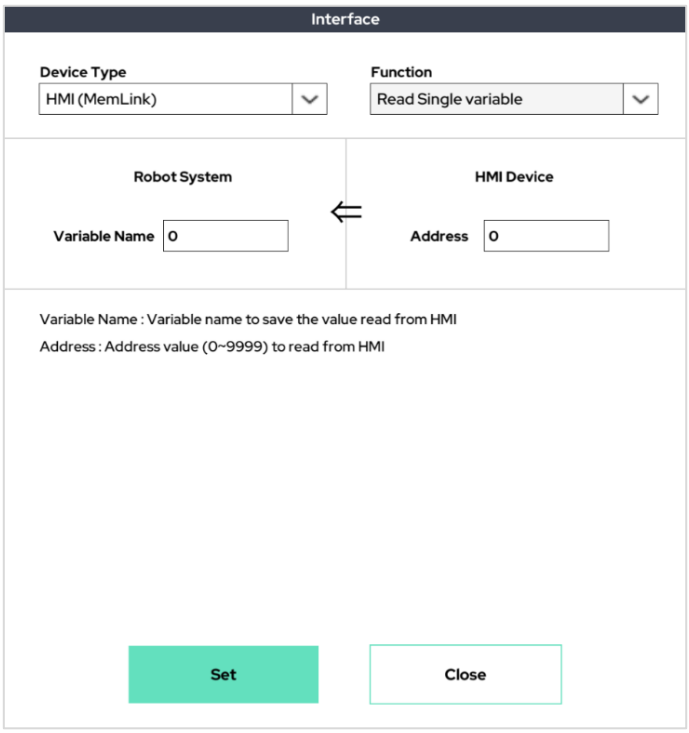
HMI (MemLink) – Einzelne Variable schreiben



Mit dieser Funktion sind Werte von einer Adresse in HMI einzugeben. Dabei werden die Zahl oder der Name der Variable als die Übergabewerte eingegeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

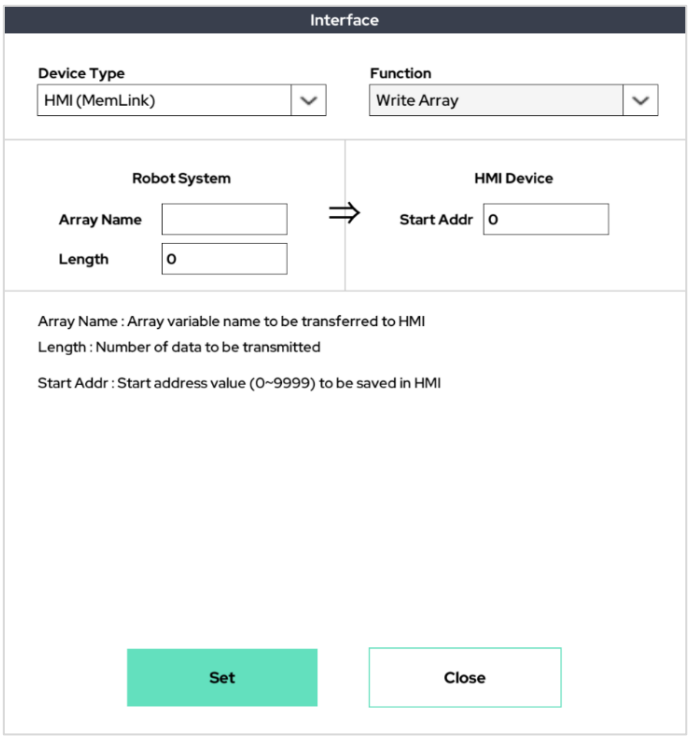
HMI (MemLink) – Einzelne Variable lesen



Mit dieser Funktion sind Werte aus einer Adresse in HMI zu lesen. Dabei werden die gelesenen Werte in der von Benutzer angegebenen Variable gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

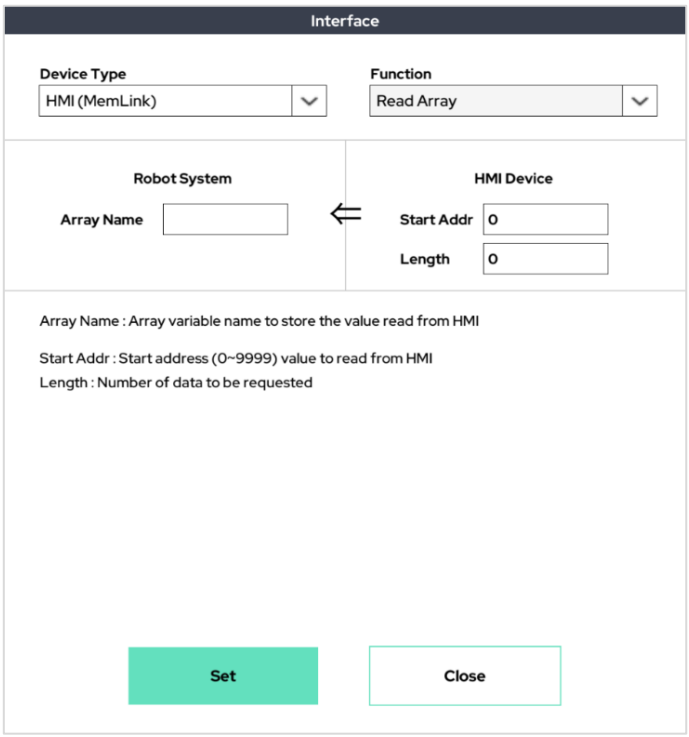
HMI (MemLink) – Mehrere Zahlen schreiben (Array)



Mit dieser Funktion sind die Daten von der Startadresse der HMI bis zur angegebenen Anzahl von Adressen einzugeben. Dabei muss das vordeklarierte Array in ‘Array Name‘ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten

© RAINBOW ROBOTICS Inc. All rights reserved.

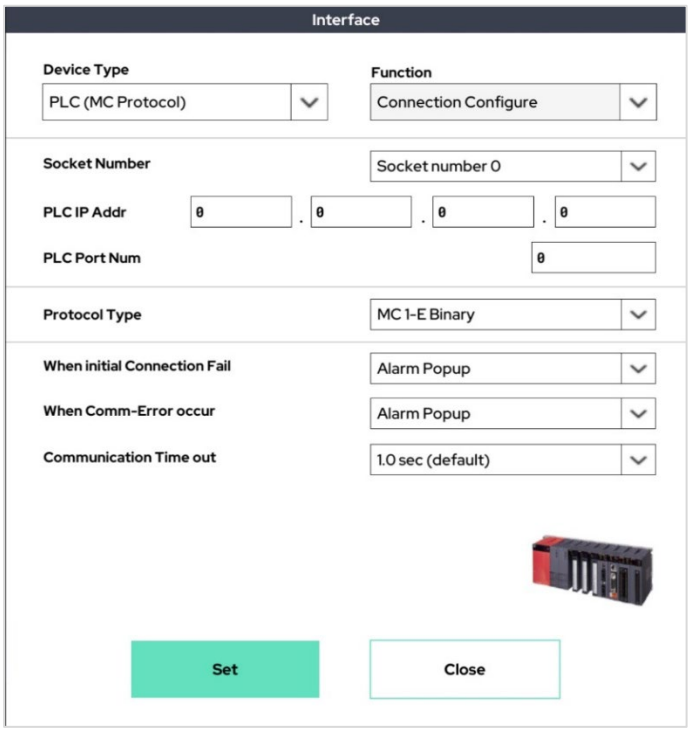
HMI (MemLink) – Mehrere Zahlen lesen (Array)



Mit dieser Funktion sind die Daten von der Startadresse der HMI bis zur angegebenen Anzahl von Adressen zu lesen. Dabei muss das vordeklarierte Array in ‘Array Name‘ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

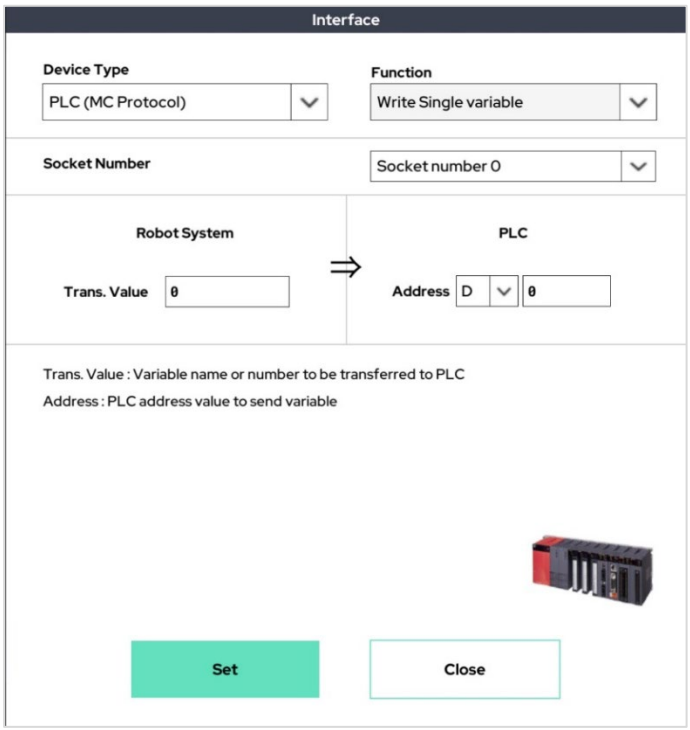
PLC (MC Protocol) – Verbindung konfigurieren



Mit dieser Funktion ist die Kommunikation zwischen Mitsubishi-PLC und dem RB-System zu verbinden. Socket-Nummer, IP-Adresse sollten eingegeben werden. Der Benutzer kann auch entscheiden, ob das Alarm-Popup nach Protokolltyp im Falle, eines Verbindungsausfalls oder eines Kommunikationsfehlers aktiviert oder ignoriert werden sollte.

© RAINBOW ROBOTICS Inc. All rights reserved.

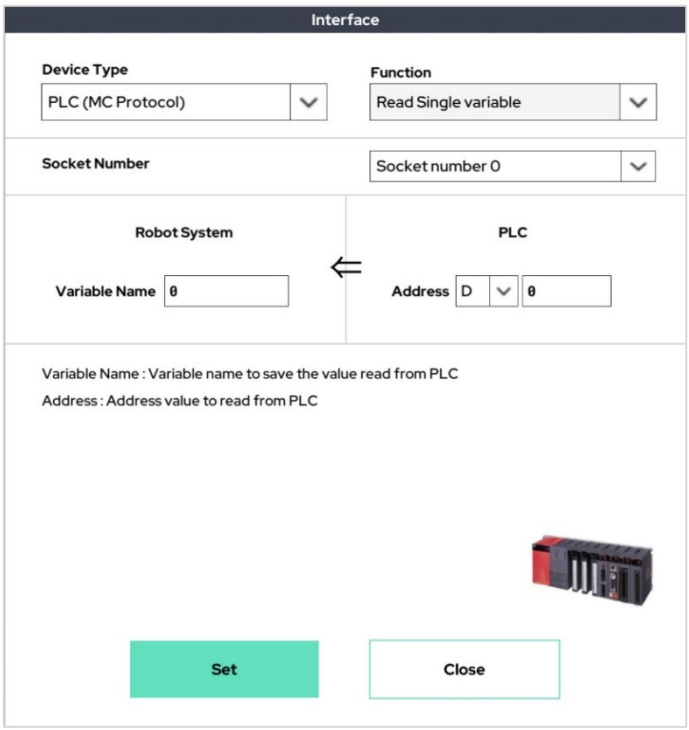
PLC (MC-Protokoll) – Einzelne Variable schreiben



Mit dieser Funktion sind Werte von einer Adresse in PLC einzugeben. Dabei werden die Zahl oder der Name der Variable als die Übergabewerte eingegeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

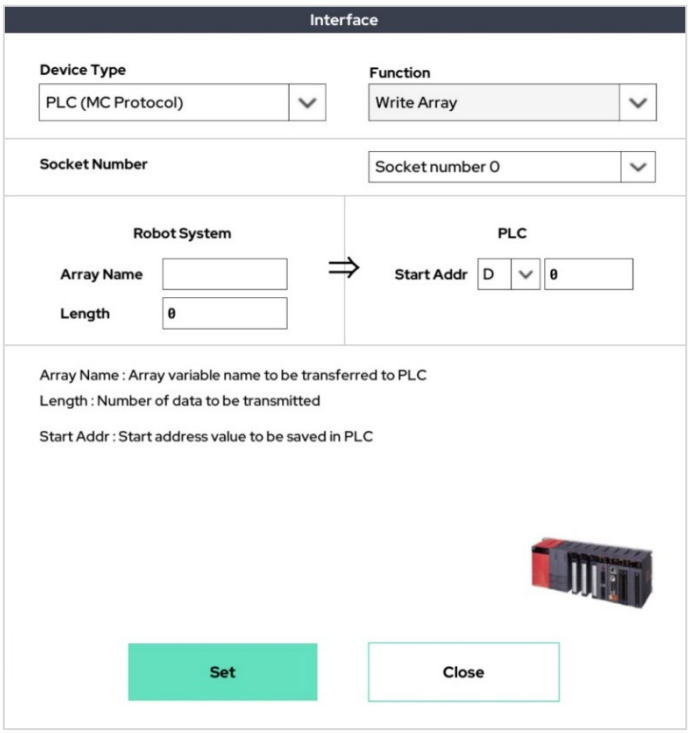
PLC (MC-Protokoll) – Einzelne Variable lesen



Mit dieser Funktion sind Werte aus einer Adresse in PLC zu lesen. Dabei werden die gelesenen Werte in der von Benutzer angegebenen Variable gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

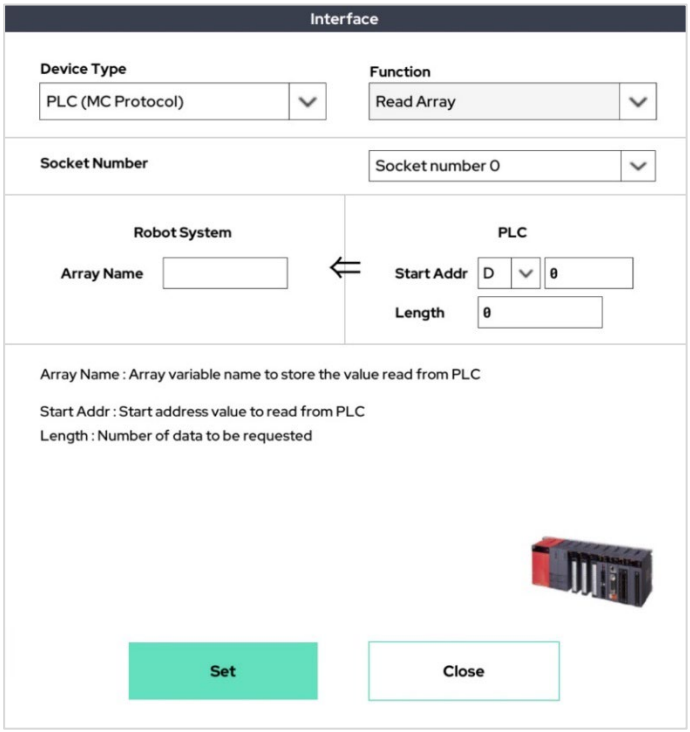
PLC (MC-Protokoll) – Mehrere Zahlen schreiben (Array)



Mit dieser Funktion sind die Zahlen von der Startadresse der PLC bis zur angegebenen Anzahl von Adressen einzugeben. Dabei muss das vordeklarierte Array in ‘Array Name‘ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

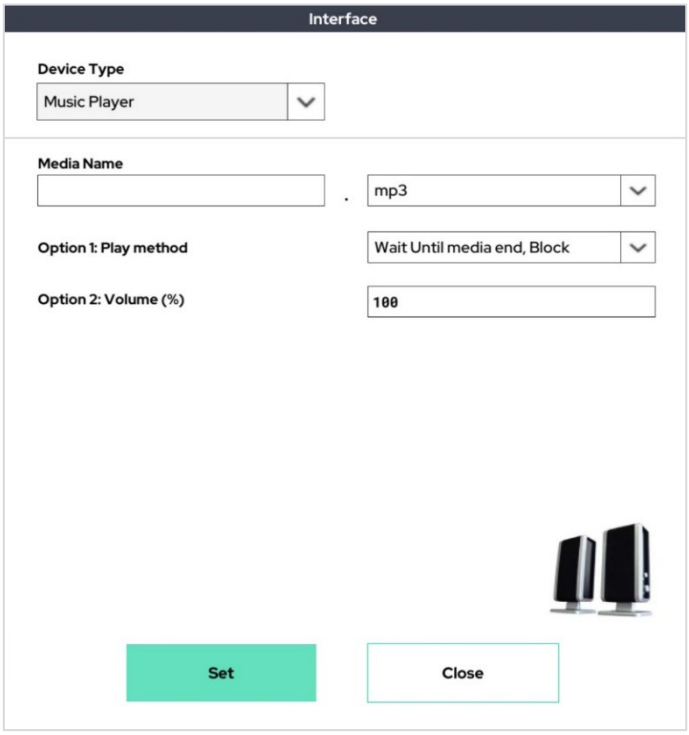
PLC (MC-Protokoll) – Mehrere Zahlen lesen (Array)



Mit dieser Funktion sind die Daten von der Startadresse der PLC bis zur angegebenen Anzahl von Adressen zu lesen. Dabei muss das vordeklarierte Array in ‘Array Name‘ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

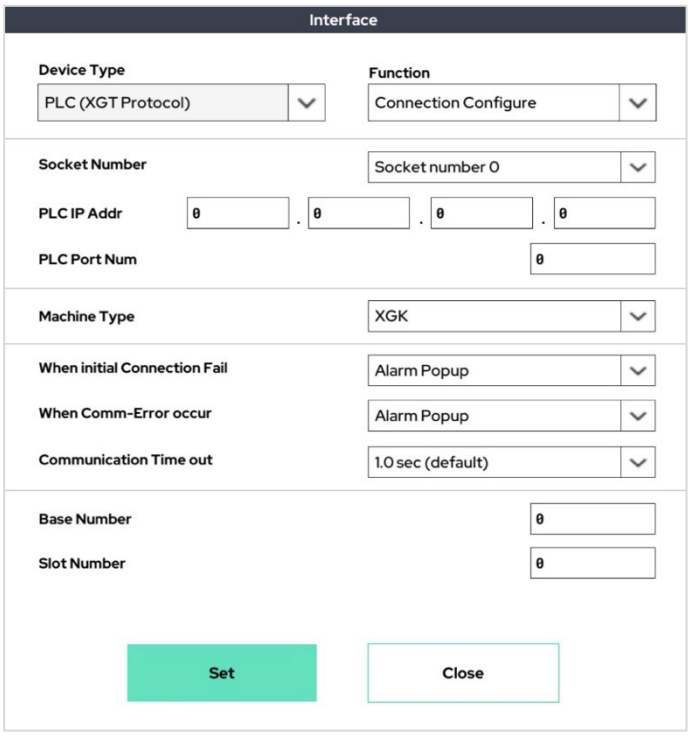
Music Player



Mit dieser Funktion wird eine MP3-Datei abgespielt. Über RB-Treiber (RB-Driver) muss der Musiktreiber installiert werden, und die MP3-Datei, die Sie abspielen möchten, muss im angegebenen Pfad vorhanden sein.

© RAINBOW ROBOTICS Inc. All rights reserved.

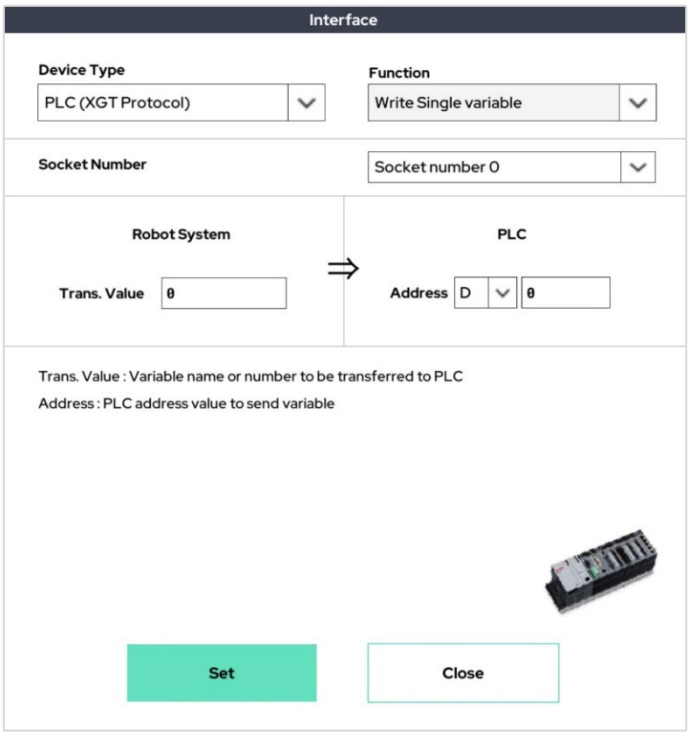
PLC (XGT-Protokoll) – Verbindung konfigurieren



Mit dieser Funktion ist die Kommunikation zwischen LS Electric PLC und dem RB-System zu verbinden. Socket-Nummer, IP-Adresse sollten eingegeben werden. Der Benutzer kann auch entscheiden, ob das Alarm-Popup nach Protokolltyp im Falle, eines Verbindungsausfalls oder eines Kommunikationsfehlers aktiviert oder ignoriert werden sollte, und ein Kommunikations-Timeout festlegen.

© RAINBOW ROBOTICS Inc. All rights reserved.

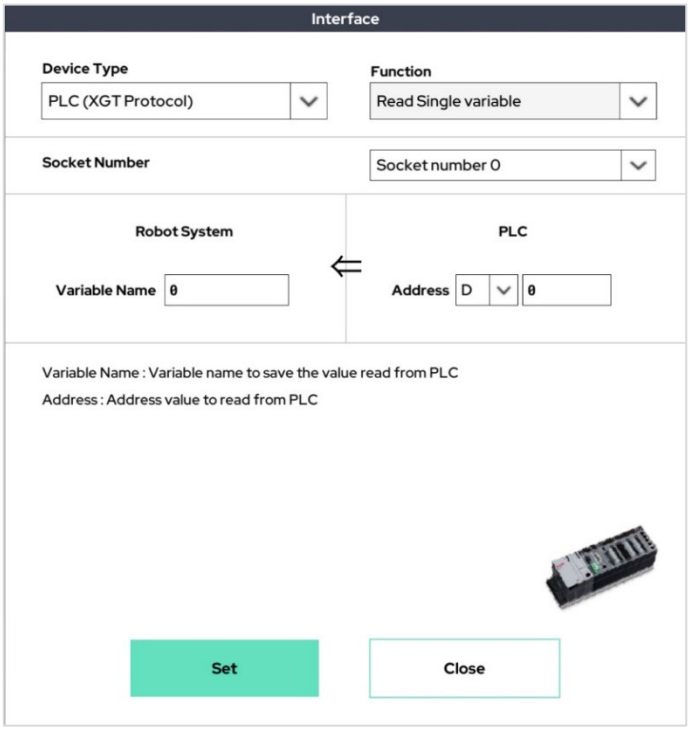
PLC (XGT-Protokoll) – Einzelne Variable schreiben



Mit dieser Funktion sind Werte von einer Adresse in PLC einzugeben. Dabei werden die Zahl oder der Name der Variable als die Übergabewerte eingegeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

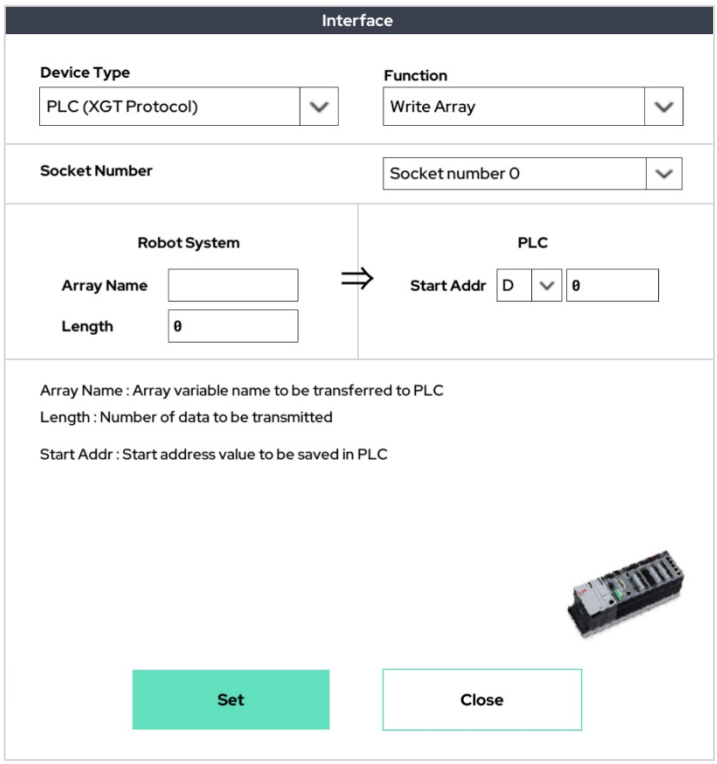
PLC (XGT-Protokoll) – Einzelne Variable lesen



Mit dieser Funktion sind Werte von einer Adresse in PLC zu lesen. Dabei werden die gelesenen Werte in der von Benutzer angegebenen Variable gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

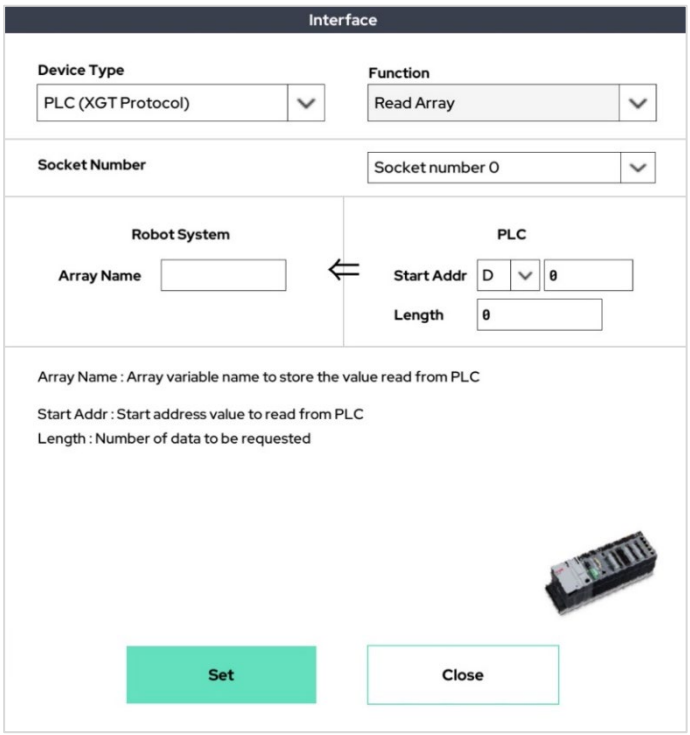
PLC(XGT-Protokoll) – Mehrere Zahlen schreiben (Array)



Mit dieser Funktion sind die Daten von der Startadresse der PLC bis zur angegebenen Anzahl von Adressen einzugeben. Dabei muss das vordeklarierte Array in ‘Array Name‘ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

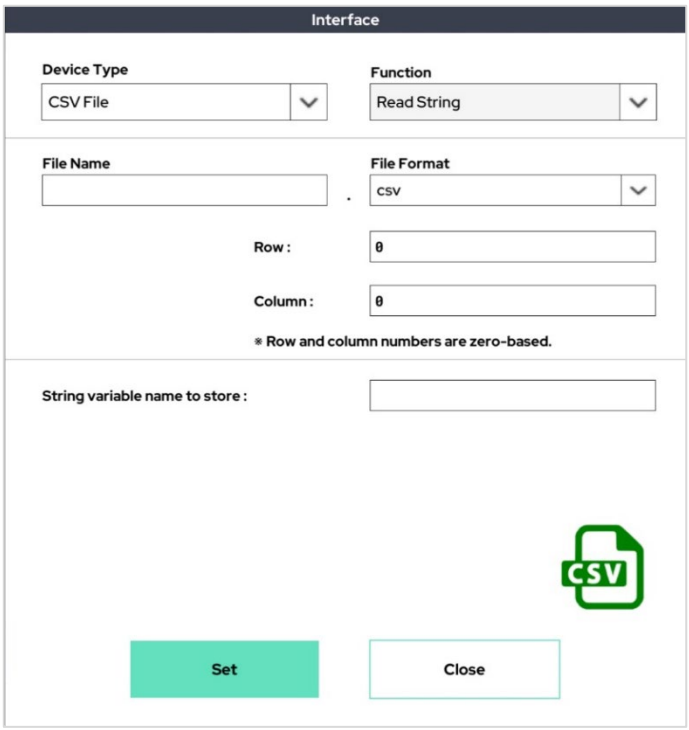
PLC (XGT-Protokoll) – Mehrere Zahlen lesen (Array)



Mit dieser Funktion sind die Daten von der Startadresse der PLC bis zur angegebenen Anzahl von Adressen zu lesen. Dabei muss das vordeklarierte Array in ‘Array Name‘ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

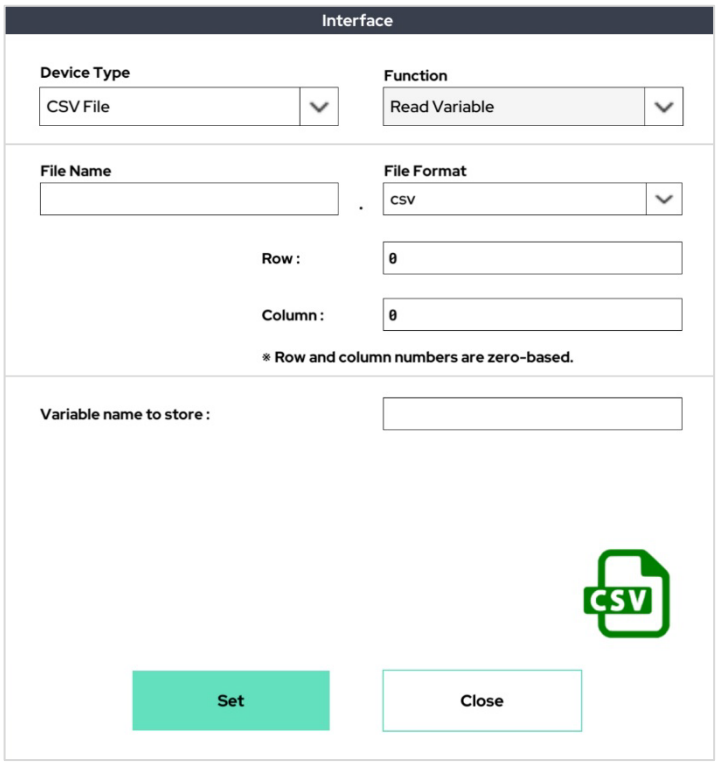
CSV File – Zeichenfolge lesen



Mit dieser Funktion ist die Zeichenfolge au seiner CSV-Datei zu lesen. Die CSV-Datei muss im angegebenen Pfad gespeichert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

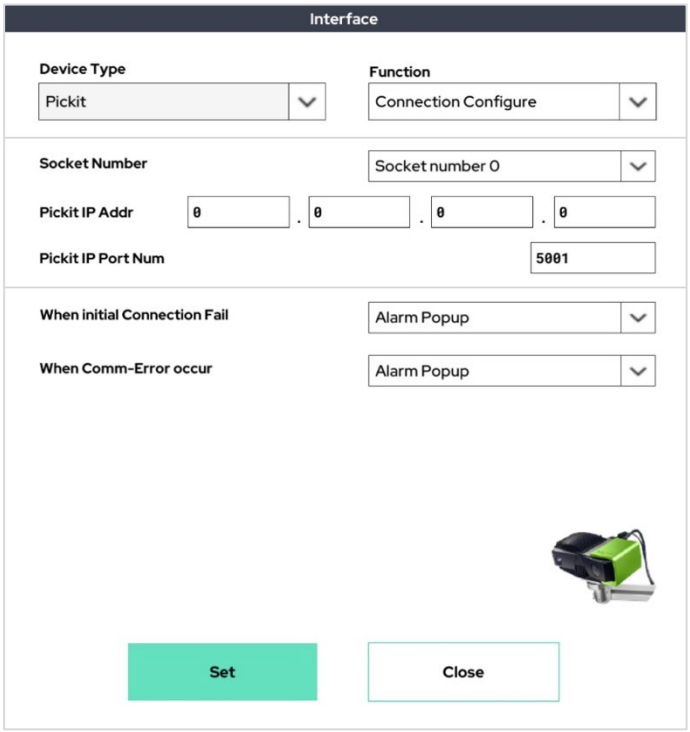
CSV File – Einzelne Variable lesen



Mit dieser Funktion ist einzelne Zahl aus CSV-Datei zu lesen. Die CSV-Datei muss im angegebenen Pfad gespeichert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Pickit – Verbindung konfigurieren

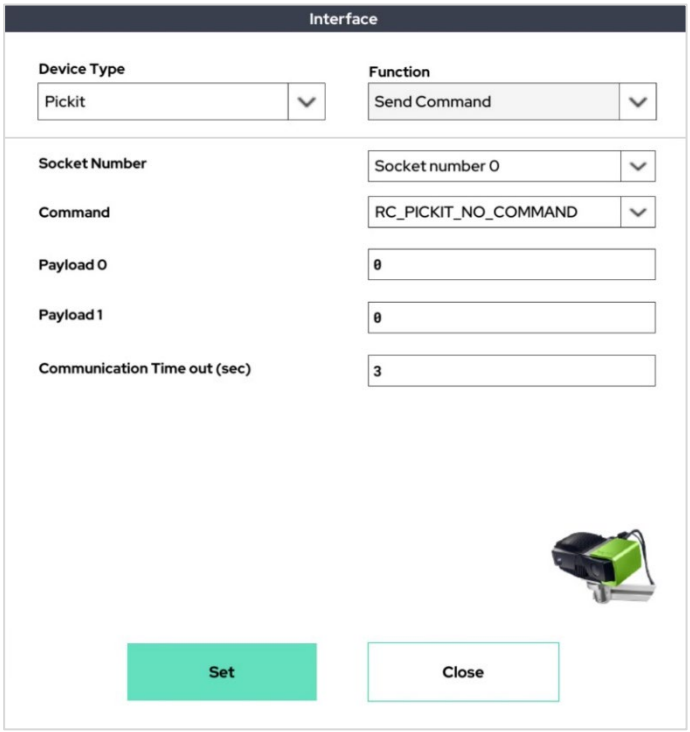


Mit dieser Funktion wird die Kommunikation zwischen Pickit und dem RB-System verbunden. Socket-Nummer, IP-Adresse sollten eingegeben werden. Der Benutzer kann auch entscheiden, ob das Alarm-Popup im Falle eines Verbindungsausfalls oder eines Kommunikationsfehlers aktiviert oder ignoriert werden sollte.

.

© RAINBOW ROBOTICS Inc. All rights reserved.

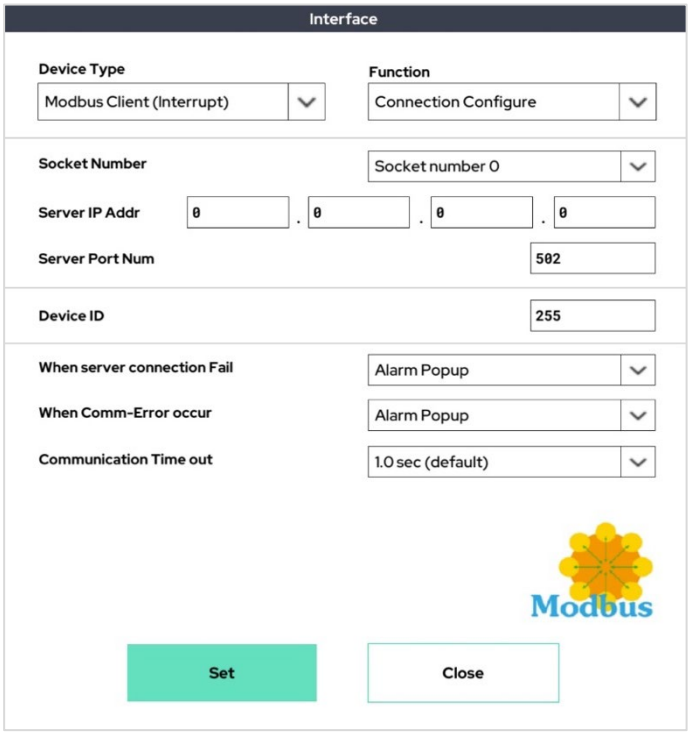
Pickit - Befehl senden



Der an Pickit zu sendende Befehl und die dem Befehl entsprechende Daten werden eingestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

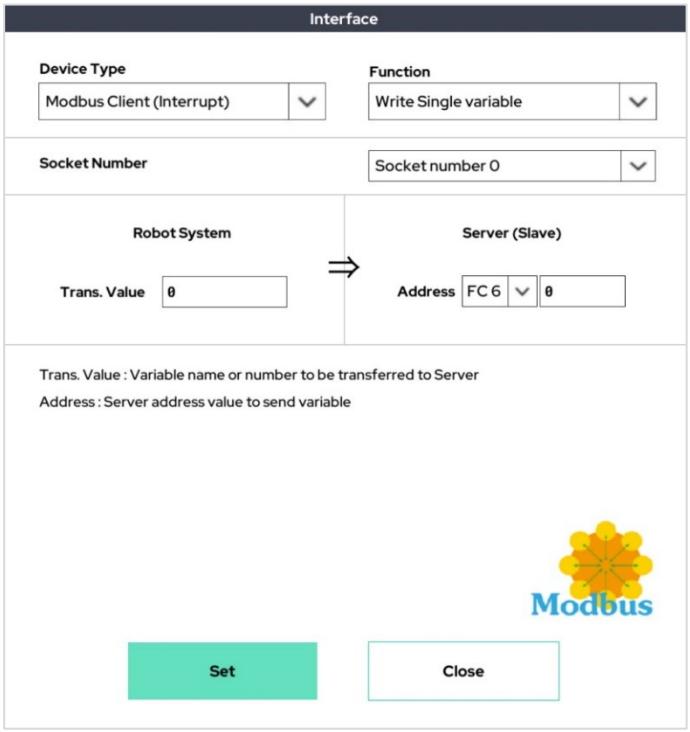
Modbus Client(Interrupt) - Verbindung konfigurieren



Mit dieser Funktion ist das RB-System als Client in der Modbus-Kommunikation zu verbinden. Socket-Nummer, IP-Adresse sollten eingegeben werden. Der Benutzer kann auch entscheiden, ob das Alarm-Popup im Falle eines Verbindungsausfalls oder eines Kommunikationsfehlers aktiviert oder ignoriert werden sollte.

© RAINBOW ROBOTICS Inc. All rights reserved.

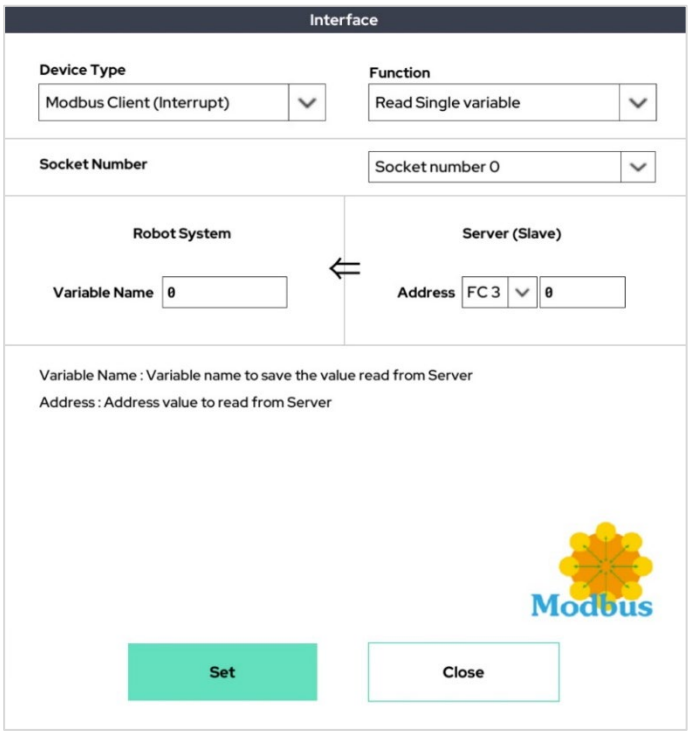
Modbus Client(Interrupt) - Einzelne Variable schreiben



Mit dieser Funktion ist ein Wert von Wortdaten an einer Adresse über die Modus-Kommunikation einzugeben. Dabei wird der Name einer Zahl oder Variablen für den Übertragungswert eingegeben.

© RAINBOW ROBOTICS Inc. All rights reserved.

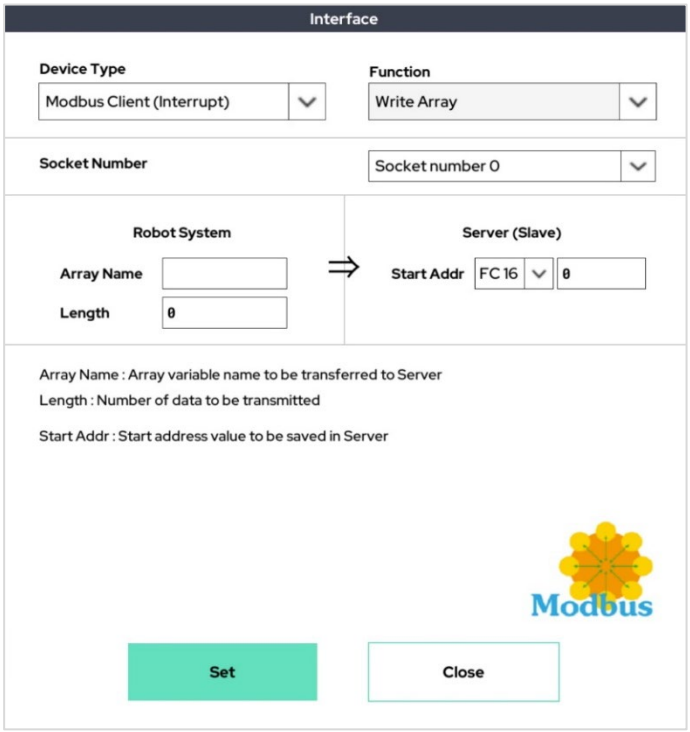
Modbus Client(Interrupt) – Einzelne Variable lesen



Mit dieser Funktion ist ein Wert von Wortdaten an einer Adresse über die Modus-Kommunikation zu lesen. Dabei wird der gelesene Wert in der bezeichneten Variablen gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

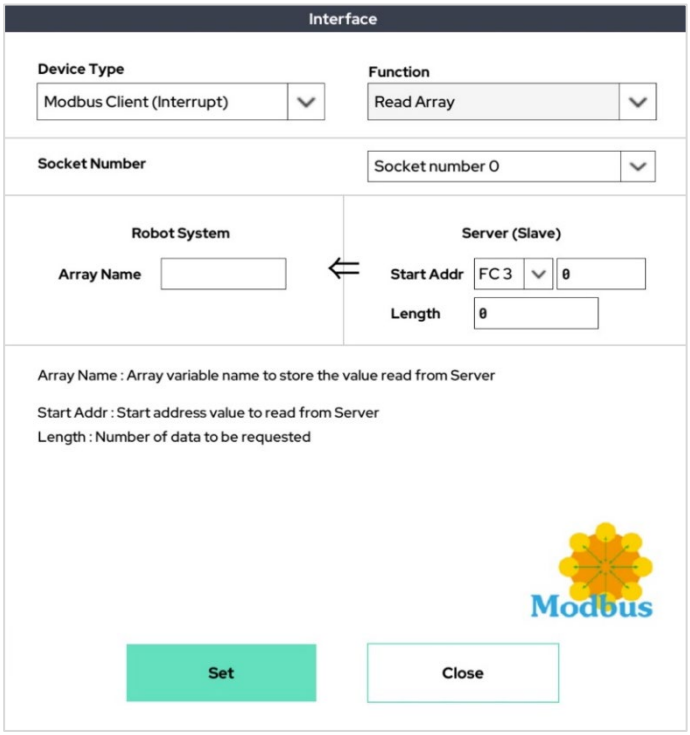
Modbus Client (Interrupt) – Mehrere Zahlen schreiben (Array)



Mit dieser Funktion sind die Wortdaten von der Startadresse bis zur angegebenen Anzahl von Adressen über die Modbus-Kommunikation einzugeben. Dabei muss das zuvor deklarierte Array in ‘Array Name’ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

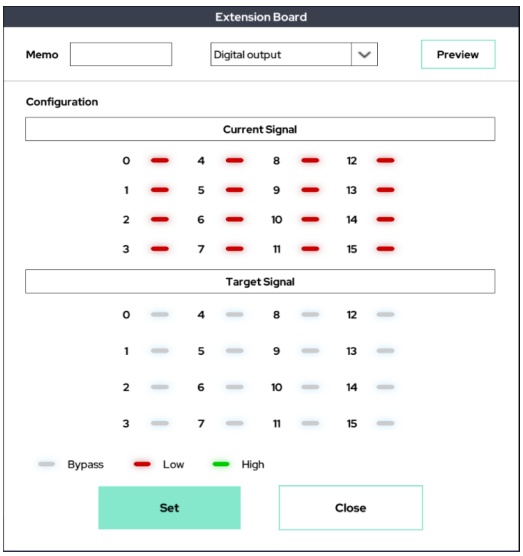
Modbus Client(Interrupt) – Mehrere Variablen schreiben (Array)

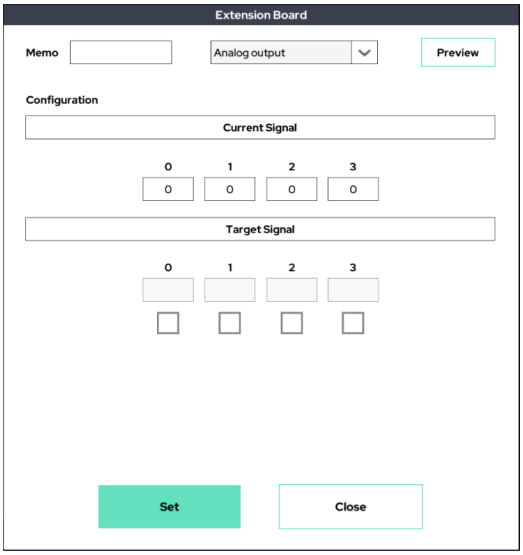


Mit dieser Funktion sind die Daten von der Startadresse bis zur angegebenen Anzahl von Adressen über die Modbus-Kommunikation zu lesen. Dabei muss das zuvor deklarierte Array in ‘Array Name’ geschrieben werden und die Länge darf die maximale Länge des Arrays (20) nicht überschreiten.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Extension Board (Ein/Aus-Erweiterungsfunktion):

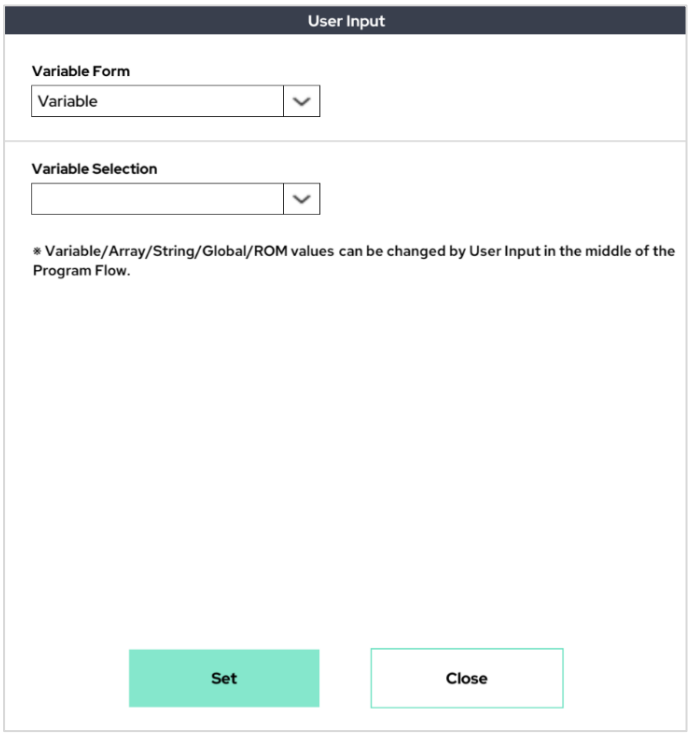




Mit dieser Funktion ist der digitale/analoge Ausgang zu steuern, wenn ein erweitertes Ein/Aus-Modul gekauft und verwendet wird. Die Verwendungsmethode ist die gleiche wie bei den bestehenden D.Output und An.Input.

© RAINBOW ROBOTICS Inc. All rights reserved.

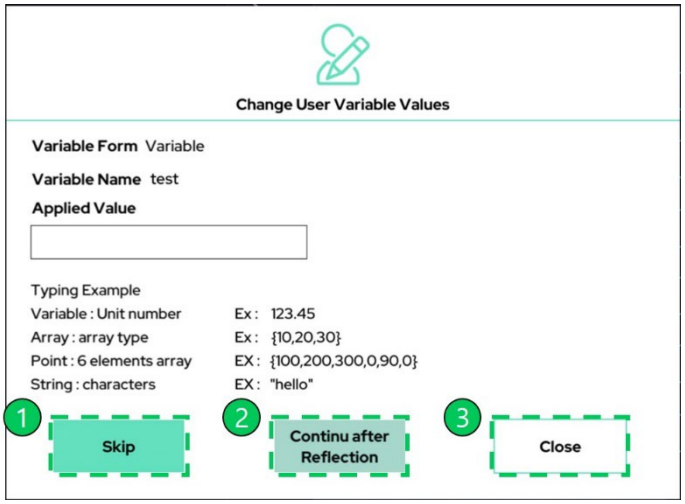
■ Unser Input (Benutzereingabefunktion):



Mit dieser Funktion kann der Benutzer während der Programmausführung den Wert von bestimmten Variablen(Parameter) willkürlich ändern . Unter Variable/Array/Point/String/Global/ROM verfügbar.

© RAINBOW ROBOTICS Inc. All rights reserved.

Bei der Ausführung erscheint ein Popup-Fenster wie folgt:



Drei Funktionen im Popup-Fenster werden ausgewählt.

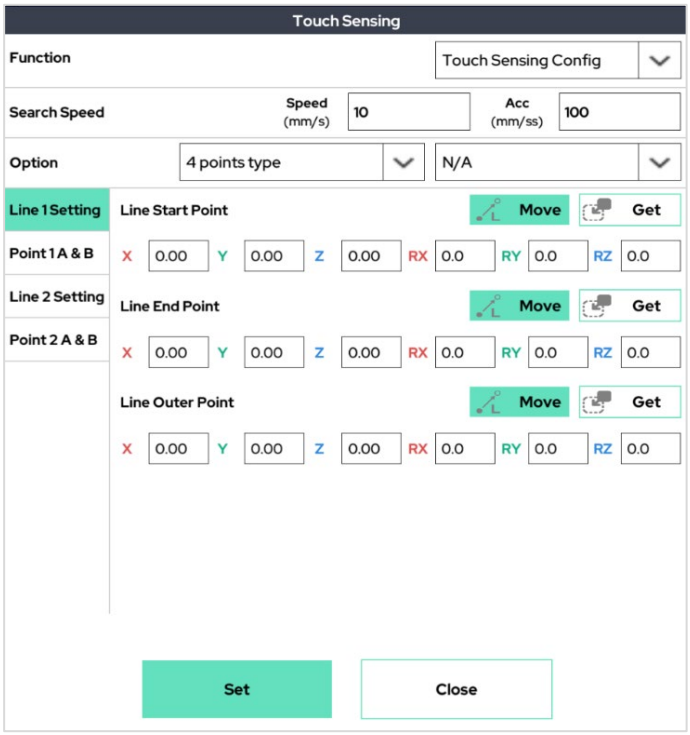
① Das Programm wird fortgesetzt, ohne die entsprechende Variable zu ersetzen.

② Wenn Sie bei der Änderung von ‘Applied Values’ die Daten eingeben und drücken, um die eingegeben Daten zu übernehmen, dann wird das Programm aktualisiert und fortgesetzt.

③ Das Programm wird beendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Touch Sensing (Berührungserkennungsfunktion):



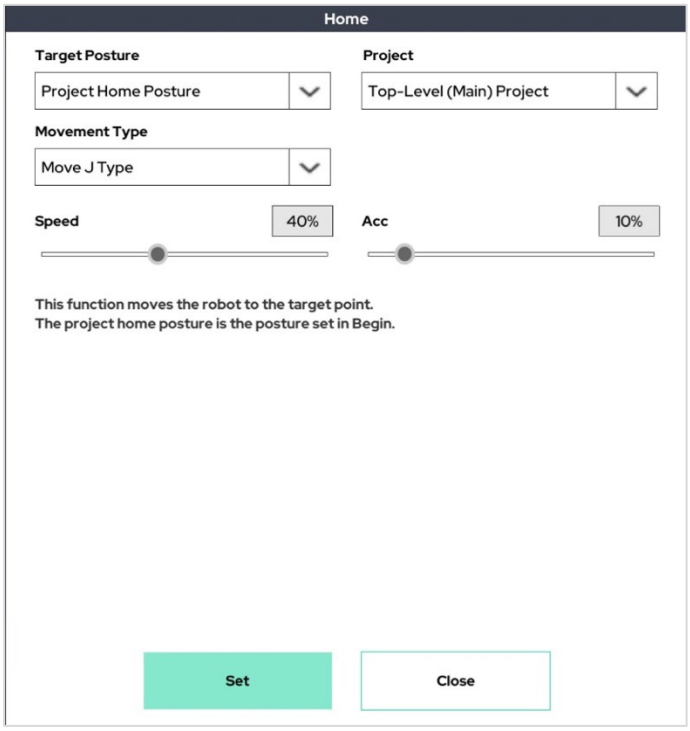
Die Berührungserkennungsfunktion ist eine Unterfunktion für die Schweißanwendung.

Mit dieser Funktion wird die Bewegung des Grundmaterials erkannt und die Bewegungsrichtung des Grundmaterials wird zum Schweißen verwendet.

Eine detaillierte Beschreibung dieser Funktion finden Sie in einem separaten Handbuch.

© RAINBOW ROBOTICS Inc. All rights reserved.

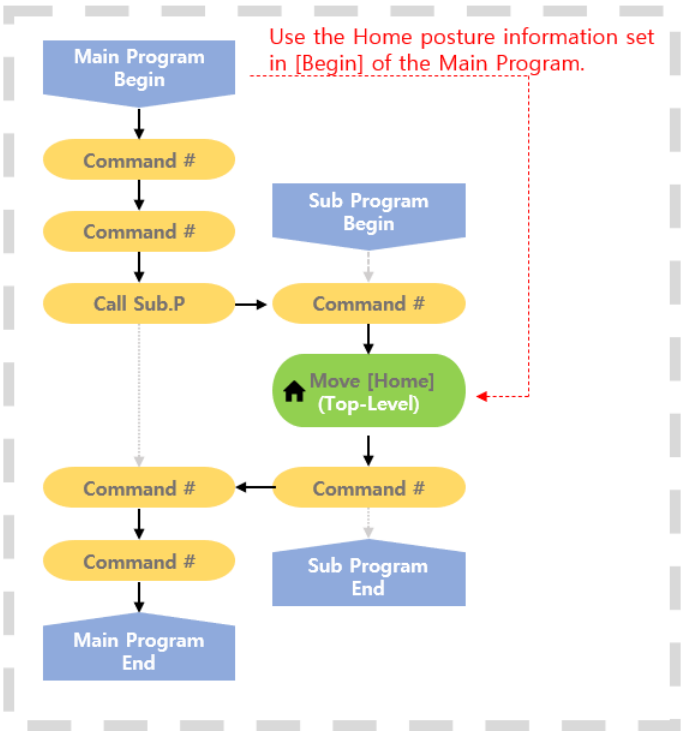
■ Home:

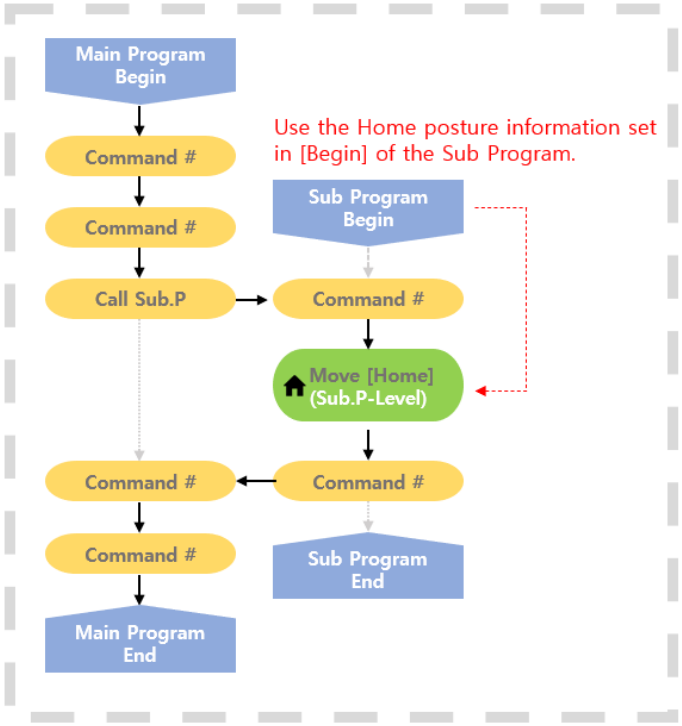


Mit der Home-Funktion wird der Roboter an die Startposition oder an die Gelenk-Nullposition bewegt. Dabei kann der Benutzer den Bewegungstyp auswählen. Wenn der Roboter an die Startposition bewegt wird, kann der Benutzer eine von Startposition des Hauptprogramms und des Unterprogramms auswählen.

© RAINBOW ROBOTICS Inc. All rights reserved.

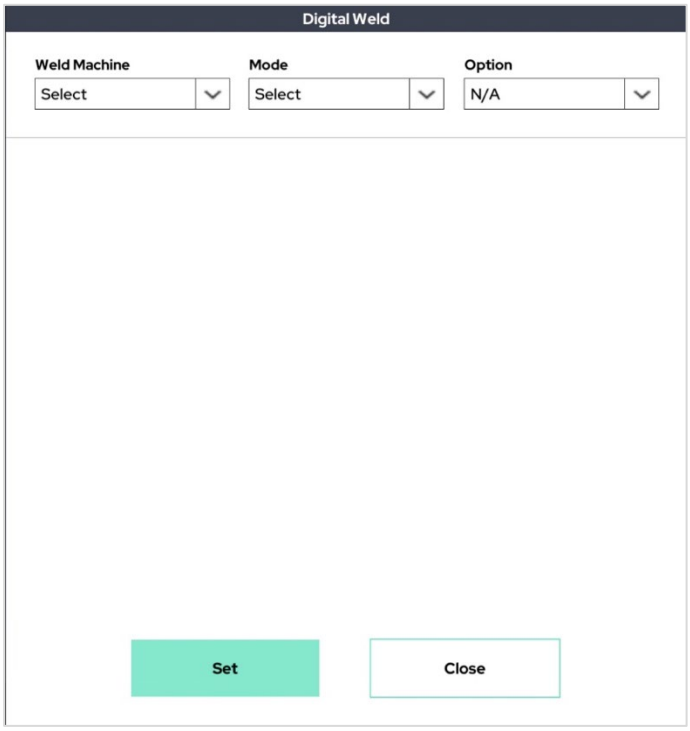
Das folgende Diagramm zeigt den Unterschied zwischen dem Fall, dass der Roboter sich zur Startposition des Hauptprojekts bewegt, und dem Fall, dass der Roboter sich zur Startposition des Unterprojekts bewegt, wenn die Home-Funktion im Unterprogramm verwendet wird.





© RAINBOW ROBOTICS Inc. All rights reserved.

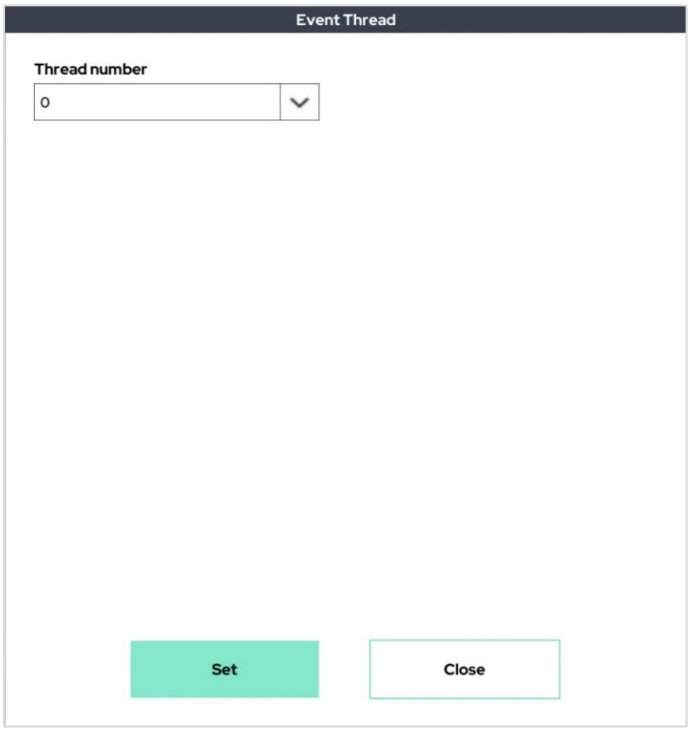
■ D.Weld (Schweißgerät) :



Mit dieser Funktion kann das digitale Schweißgerät verwendet werden. Nach Auswahl des zu verwendenden Schweißgeräts kann der Benutzer mit ‘Weld start’, ‘Weld off’ oder ‘Weld Setting’ fortfahren.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Event-Tread-Call (Aufruf):



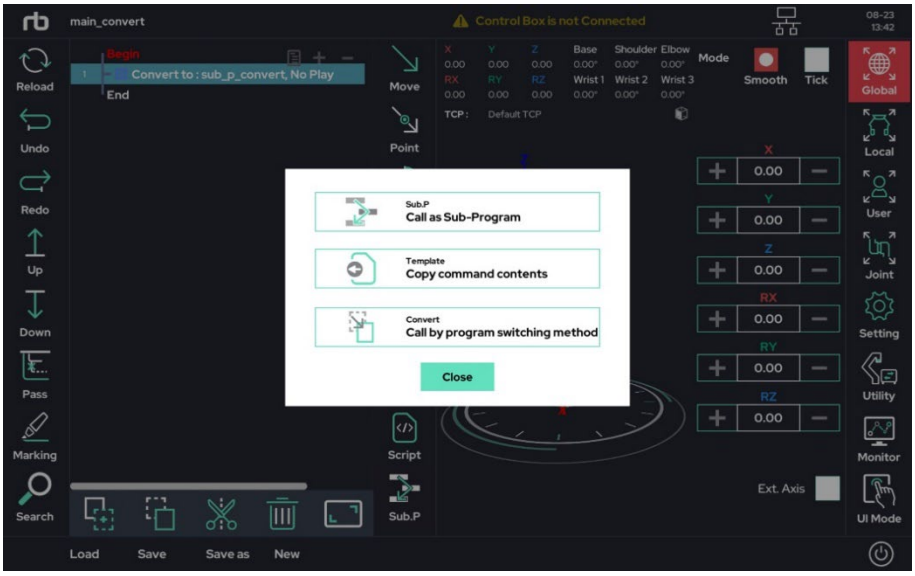
Mit dieser Funktion wird Event-Thread im Hauptprogramm ausgeführt. Der Event-Thread wird nur ausgeführt, wenn der Event-Thread-Call(Aufruf) im Hauptprogramm ausgeführt wird. In diesem Fall kann die Nummer des auszuführenden Event-Threads ausgewählt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

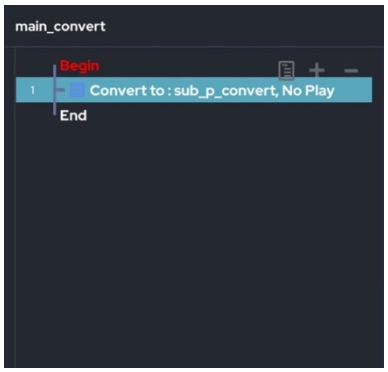
■ Convert (Konvertieren):

Mit dieser Funktion wird das Hauptprogramm konvertiert. Im Gegensatz zu der bestehenden Sub.P und Template ändert sich das Hauptprogramm, sodass sich auch das auf der UI angezeigte Programm ändert.

Wenn Sie im Programm auf das Symbol Sub.P klicken, erscheint das folgende Popup-Fenster. Drücken Sie dann ‚Call by program switching method'.



Wenn Sie das Programm, zu dem Sie wechseln möchten, im Popup-Fenster von Datei-Explorer auswählen, wird der Befehl wie unten gezeigt erstellt.



© RAINBOW ROBOTICS Inc. All rights reserved.

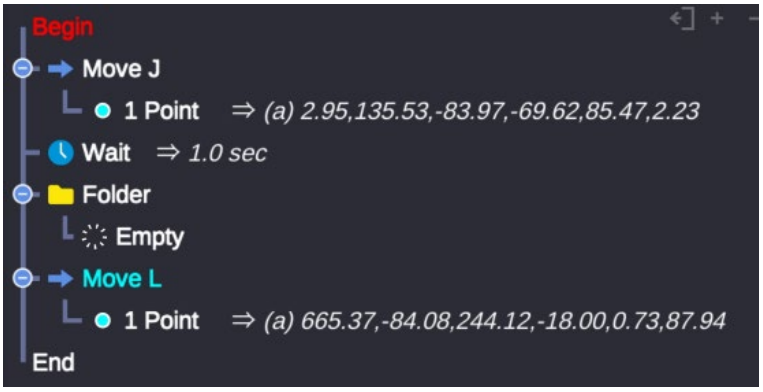
7.5 Programmbearbeitung

Die Reihenfolge oder Struktur der in die Programmierliste eingegebenen Befehle können geändert werden.

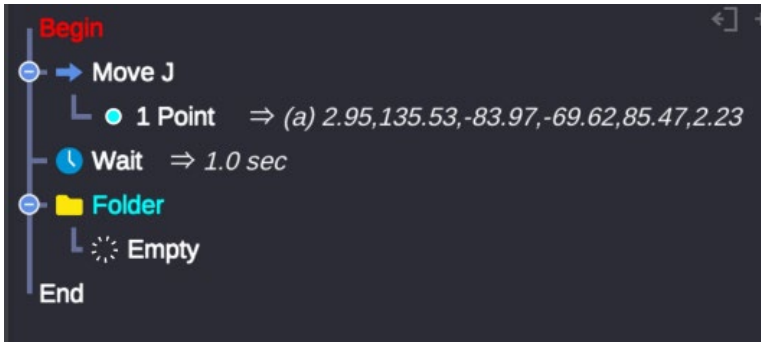
Siehe 6.1 für die Beschreibung des Bearbeitungssymbols und wie das Programm anhand der Beispiele bearbeitet wird.

■ Cut (Ausschneiden)/ Paste (Einfügen)

Step1) Wählen Sie den Befehl aus, den Sie ausscheiden möchten. Der ausgewählte Befehl wird blau angezeigt. Im Beispiel ist die Zeile MoveL ausgewählt.

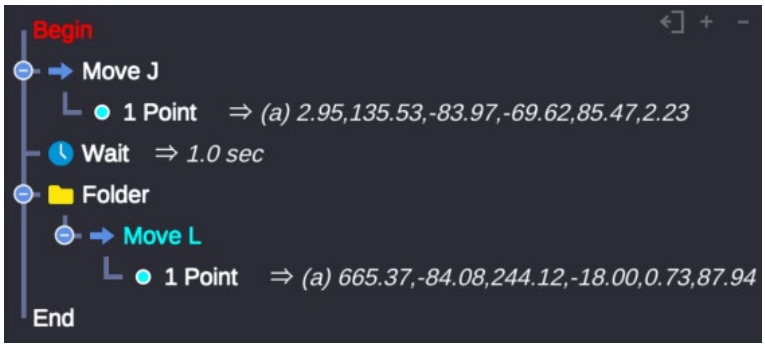


Step2) Drücken Sie die Cut(Ausschneiden)-Taste. Sobald Sie auf 'Cut' klicken, verschwindet die Zeile aus dem Programmbaum.



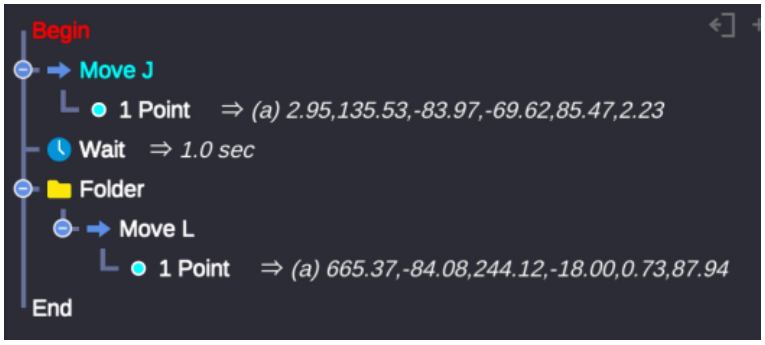
© RAINBOW ROBOTICS Inc. All rights reserved.

Step3) Klicken Sie auf die einzufügende Stelle, und dann auf die Schaltfläche ‘Paste’. Im Beispiel wird der ausgeschnittene Befehl in den Ordner eingefügt.



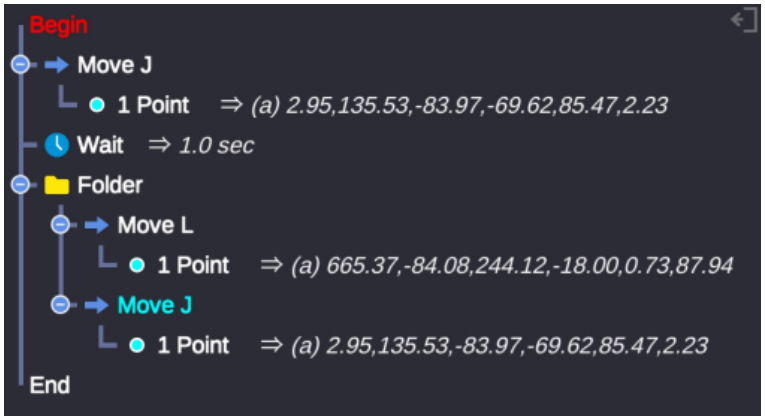
■ Copy (Kopieren)/ Paste (Einfügen)

Step1) Wählen Sie diejenige aus, die Sie kopieren möchten. Der ausgewählte Befehl wird blau angezeigt. Im Beispiel ist die Zeile MoveJ ausgewählt.



Step2) Drücken Sie die Schaltfläche ‘Copy’ (Kopieren).

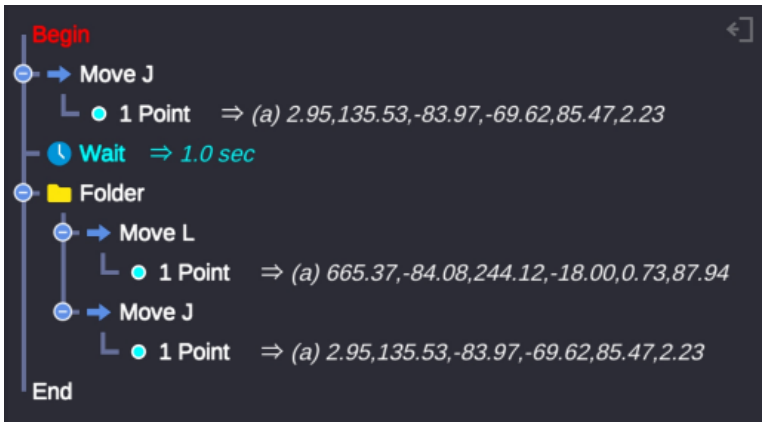
Step3) Klicken Sie auf die gewünschte Stelle und dann auf die Schaltfläche ‚Paste‘. Im Beispiel wird der Unterordner von MoveL ausgewählt und der kopierte Befehl darunter eingefügt.



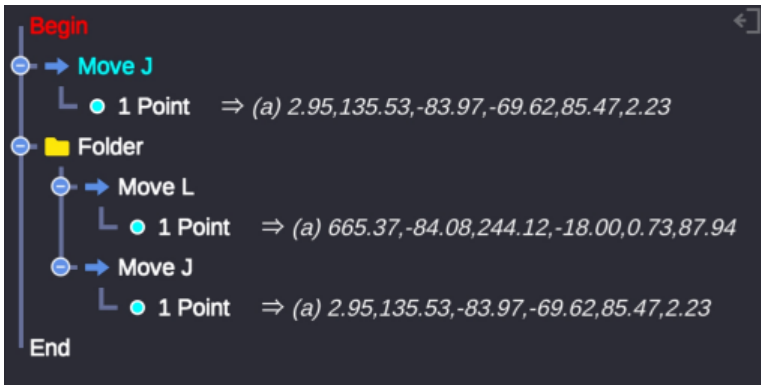
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Delete (löschen)

Step1) Wählen Sie den Befehl aus, den Sie löschen möchten. Der ausgewählte Befehl wird blau angezeigt. Im Beispiel wurde der 'Wait'-Befehl ausgewählt.



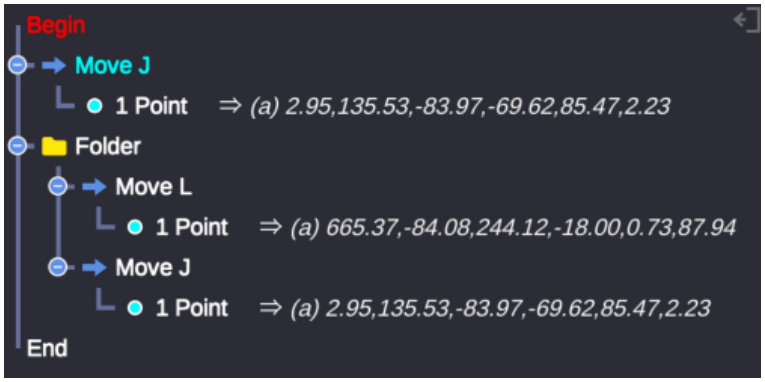
Step2) Klicken Sie auf die Schaltfläche 'Del(Entfernen)'. Der Befehl wurde wie unten gezeigt gelöscht.



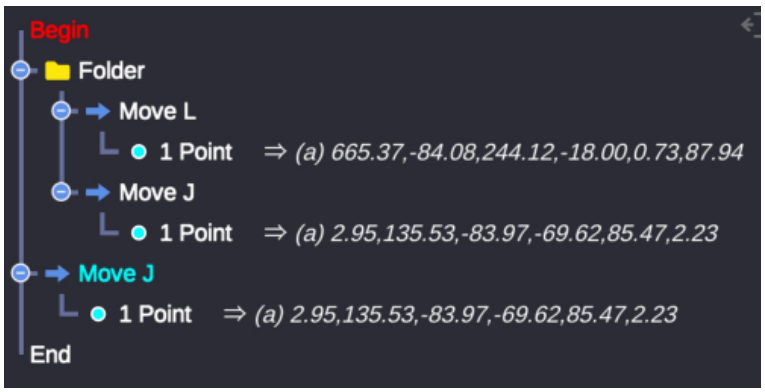
© RAINBOW ROBOTICS Inc. All rights reserved.

■ Move (Verschieben)

Step1) Wählen Sie den Befehl aus, den Sie verschieben möchten. Der ausgewählte Befehl wird blau angezeigt. Im Beispiel wurde MoveJ oben ausgewählt.



Step2) Wenn Sie auf die Schaltfläche 'Down (Abwärts)' klicken, wird der ausgewählte MoveJ wie unten gezeigt nach unten verschoben.

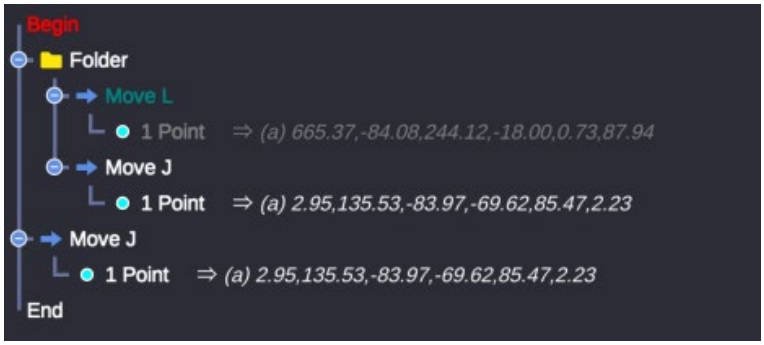


© RAINBOW ROBOTICS Inc. All rights reserved.

■ Pass (Auskommentieren)

Step1) Wählen Sie die Funktion aus, deren Ausführung Sie vorübergehend aussetzen/blockieren möchten. Der ausgewählte Befehl wird blau angezeigt. Im Beispiel wurde der MoveL-Befehl ausgewählt.

Step2) Klicken Sie auf das Pass(Auskommentieren)-Symbol. Der Befehl wird wie unten gezeigt dunkel. Auskommentierte Befehle funktionieren während der Programmausführung nicht. Wählen Sie das die ‚Pass'-Taste erneut, wenn Sie das Auskommentieren abbrechen möchten.

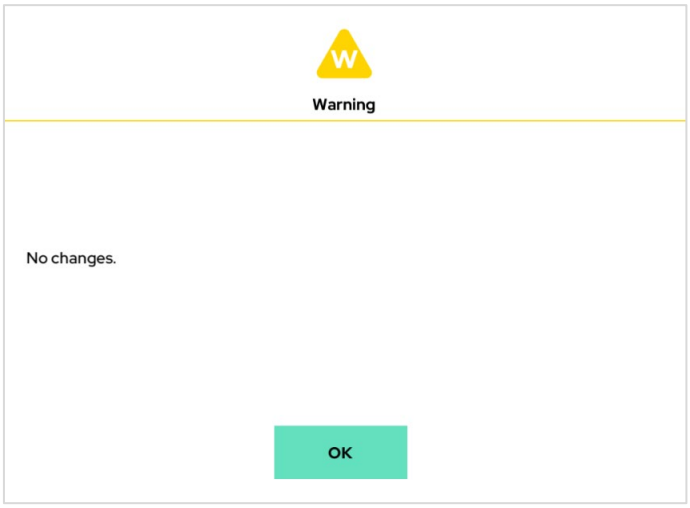


© RAINBOW ROBOTICS Inc. All rights reserved.

7.6 Programmverwaltung

Sie können das Projekt speichern, laden oder unter anderen Namen speichern.

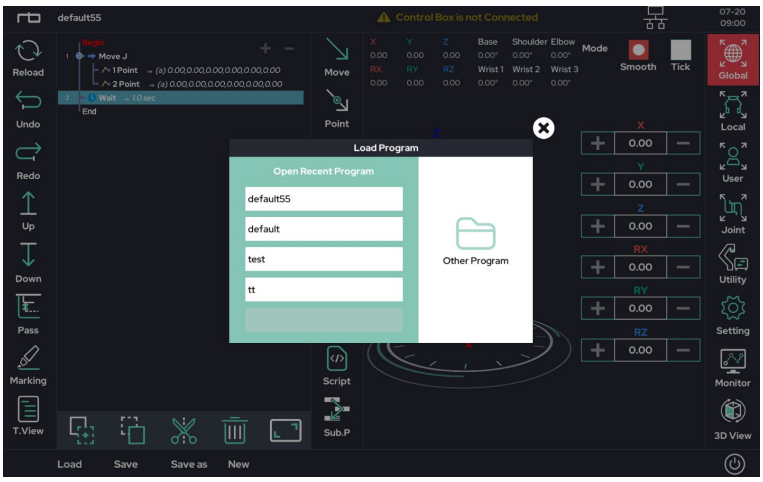
- Projekt speichern (Save project)



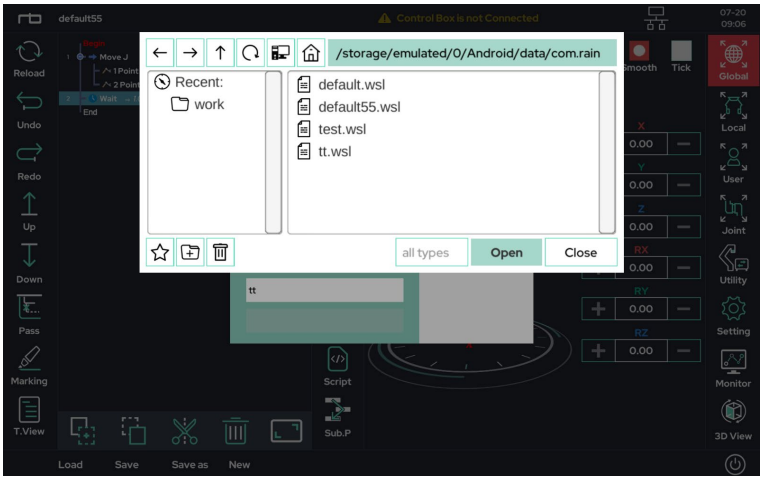
© RAINBOW ROBOTICS Inc. All rights reserved.

- Datei importieren (Open file)

  Um ein gespeichertes Projekt aufzurufen, klicken Sie unten in der Aufgabe auf FILE, wie in der folgenden Abbildung gezeigt, und eine Liste zur Auswahl wird angezeigt. Wenn Sie eine gewünschte Datei aus der Liste auswählen, wird die Datei sofort geladen. Benutzer werden aufgefordert, alle nicht gespeicherten Änderungen am aktuellen Projekt zu speichern. Es werden nur kürzlich verwendeten Dateien angezeigt.



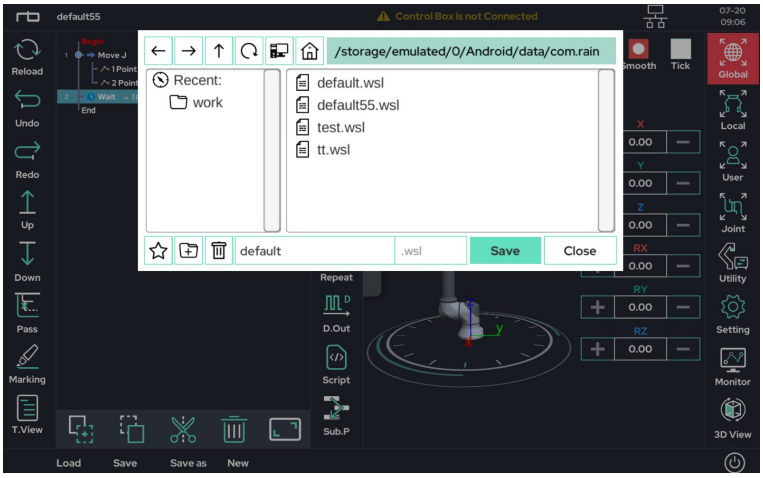
  Wenn Sie eine Datei öffnen möchten, an der Sie zuvor gearbeitet haben, können Sie auf ‚Other Program' klicken, um das Projekt im Datei-Explorer zu finden und das vorherige Programm zu laden.



© RAINBOW ROBOTICS Inc. All rights reserved.

- Speichern unter (Save As)

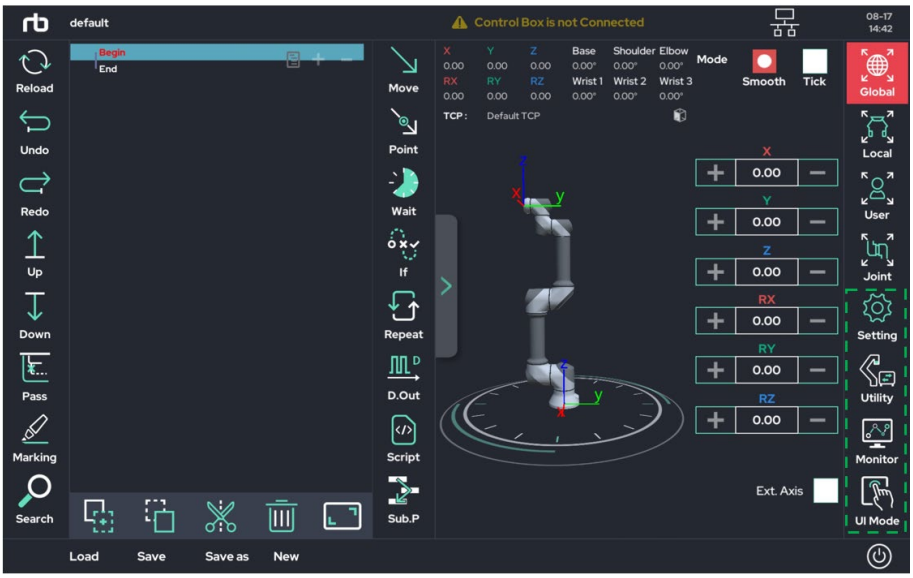
  Wenn Sie ein geöffnetes Programm verwalten, indem Sie es unter einem anderen Namen speichern, klicken Sie unter in der Aufgabe auf FILE und dann in der angezeigten Liste auf Save As (unter dem Namen ‘default‘ ist das Speichern nicht möglich).



© RAINBOW ROBOTICS Inc. All rights reserved.

7.7 Betriebliche Dienstprogramme

Auf der rechten Seite des Make-Bildschirms befinden sich neben der Jog-Funktion weitere Dienstprogramme-Funktion für den Systembetrieb.

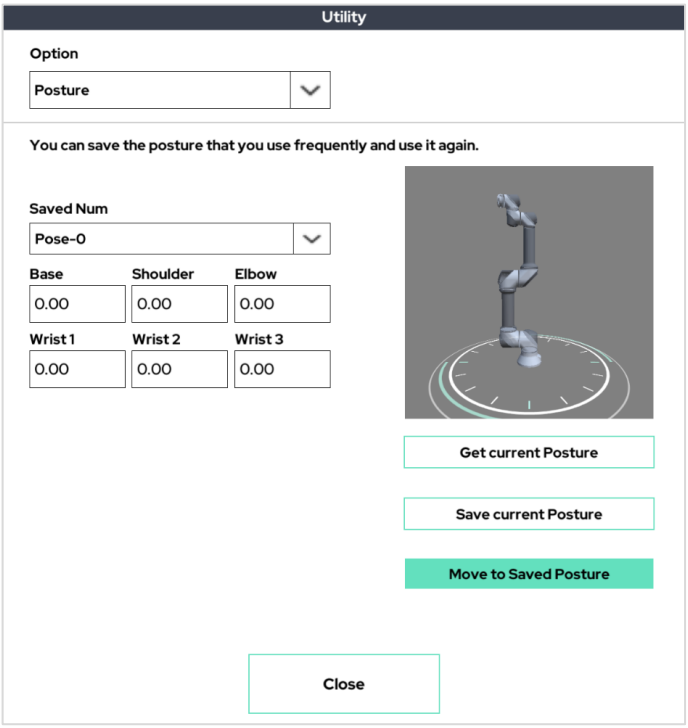


- Utility (Dienstprogramm): Häufig verwendete zusätzliche Funktionen, wie z. B. die Haltungsspeicherfunktion, die Systemeingang-/Ausgangsinformationsansichtsfunktion und die Systemausgangstestfunktion werden umfasst.
- Setting (Einstellung): Unter den im Setup eingestellten Inhalten gibt es einige Einstellungsfunktionen, die bequem mit Jog verwendet werden können.
- Monitor (Überwachung): Echtzeit-Variablen-Überwachungsfenster und Systemvariablen-Überwachungsfunktion werden angeboten.
- UI Mode (UI-Modus): Der UI-Modus kann entsprechend der Ebene des Benutzers und dem Zweck des Benutzers ausgewählt werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Utility(Dienstprogramm)-Unterfunktion

[Utility-Posture]



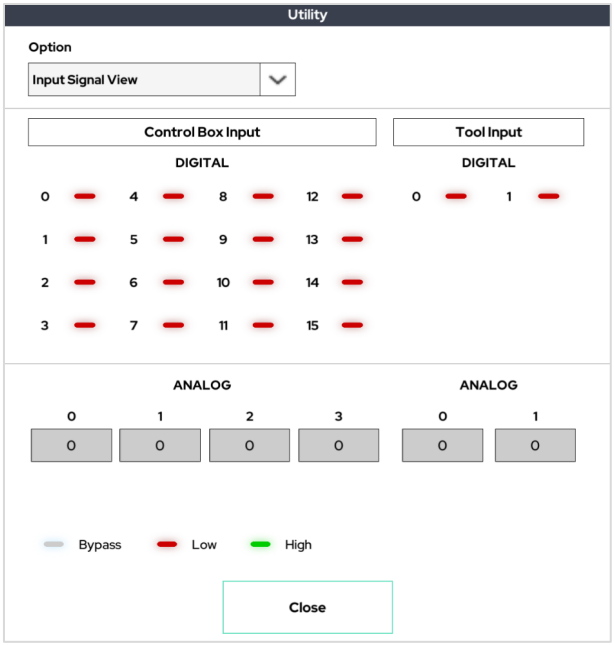
Bis zu 20 häufig verwendete Körperhaltung können auf dem UI-Tablet gespeichert und verwendet werden.

Drücken Sie die ‚Get'-Taste, um die aktuellen Haltungsinformationen abzurufen. Und drücken Sie die ‚Set'-Taste, um sie zu speichern.

Halten Sie die ‚Move'-Taste gedrückt, um zur gespeicherten Position zu wechseln.

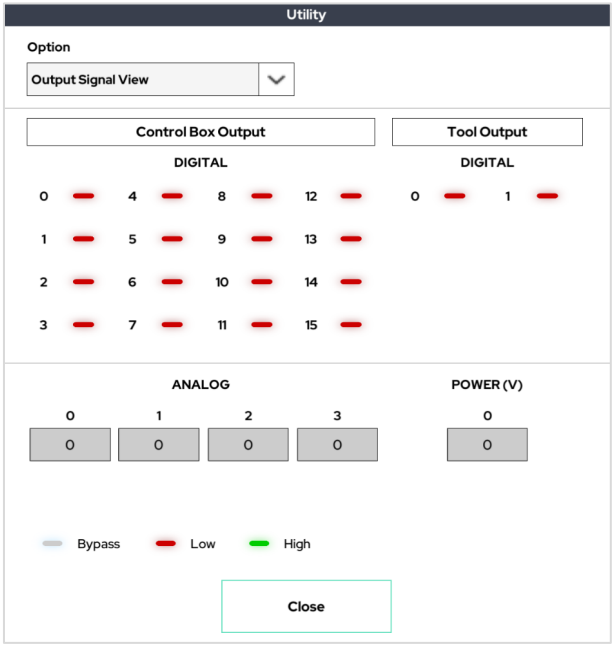
© RAINBOW ROBOTICS Inc. All rights reserved.

[Utility-Input Signal View]



Dies ist das Überwachungsfenster des Eingangssignals der Steuerbox und dem Werkzeugflansch.

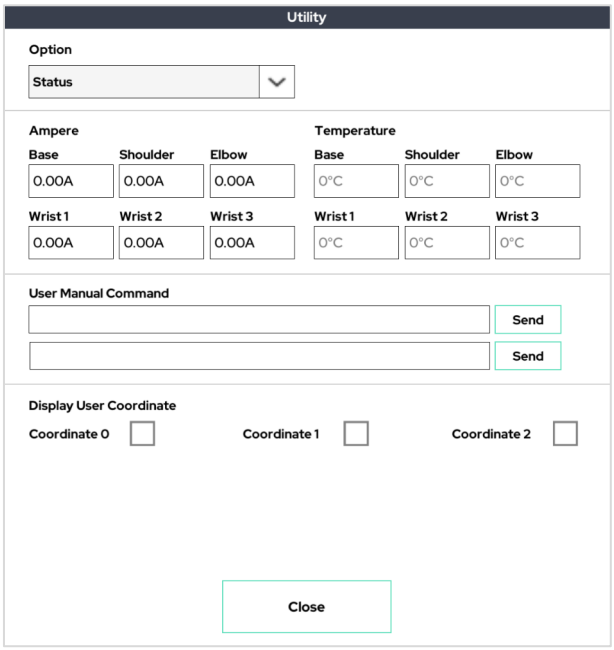
[Utility-Output Signal View]



Dies ist das Überwachungsfenster des Ausgangssignals der Steuerbox und dem Werkzeugflansch.

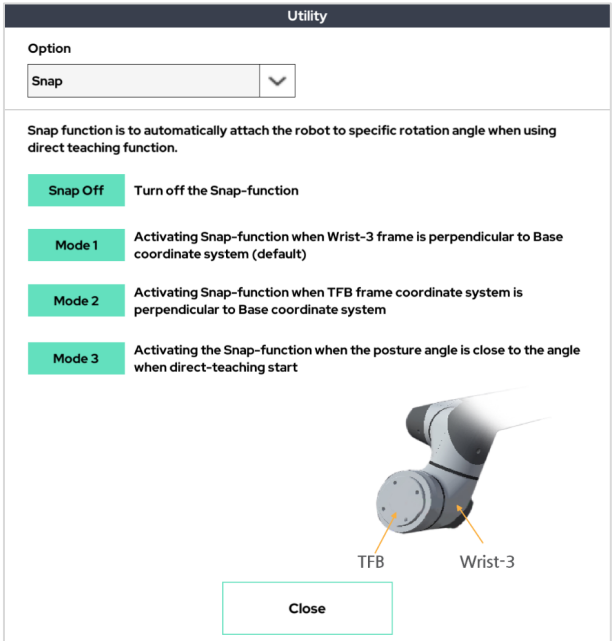
© RAINBOW ROBOTICS Inc. All rights reserved.

[Utility-Status]



Dies ist das Fenster zur Temperatur-/Stromüberwachung und zur Visualisierung des benutzerdefinierten Koordinatensystems des Roboterarms.

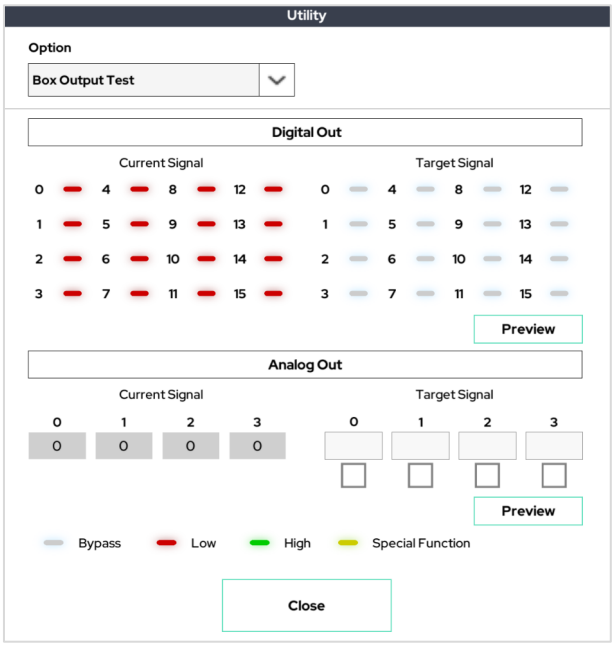
[Utility-Snap]



Dies ist das Auswahlfenster für den ‘Snap‘-Modus, das in ‘Direkt-Teaching’-Modus angewendet wird.

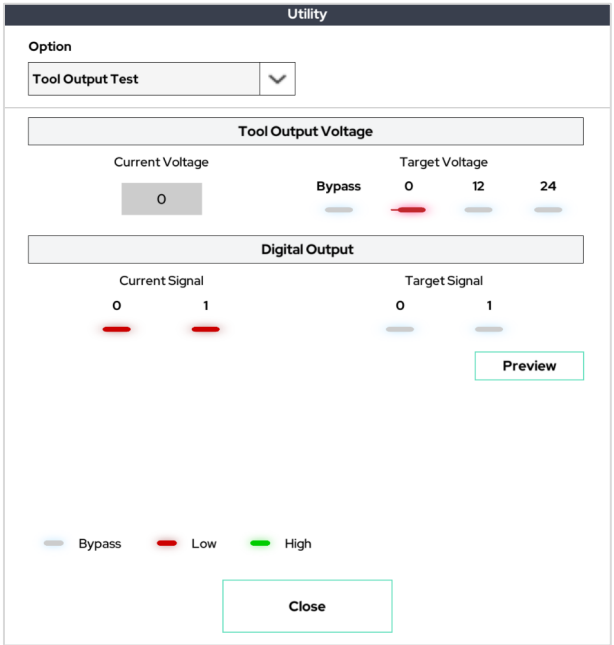
© RAINBOW ROBOTICS Inc. All rights reserved.

[Utility-Box Output Test]



Dies ist das Fenster, in dem die Ausgabe der Steuerbox getestet werden kann.

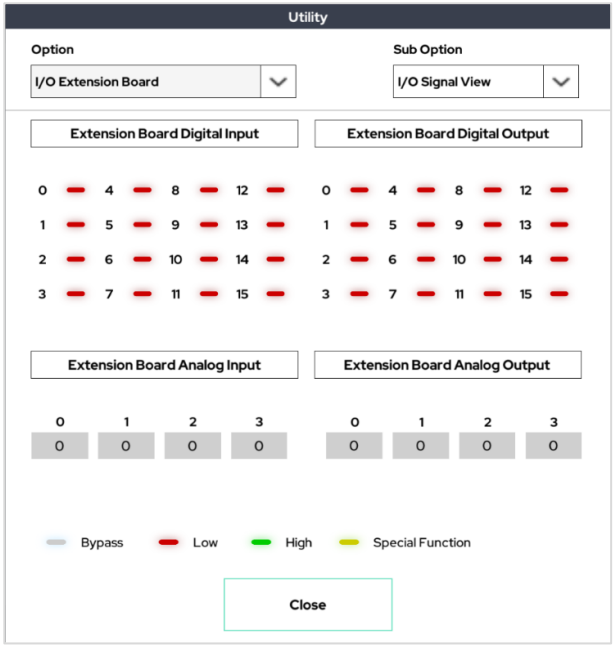
[Utility-Tool Output Test]



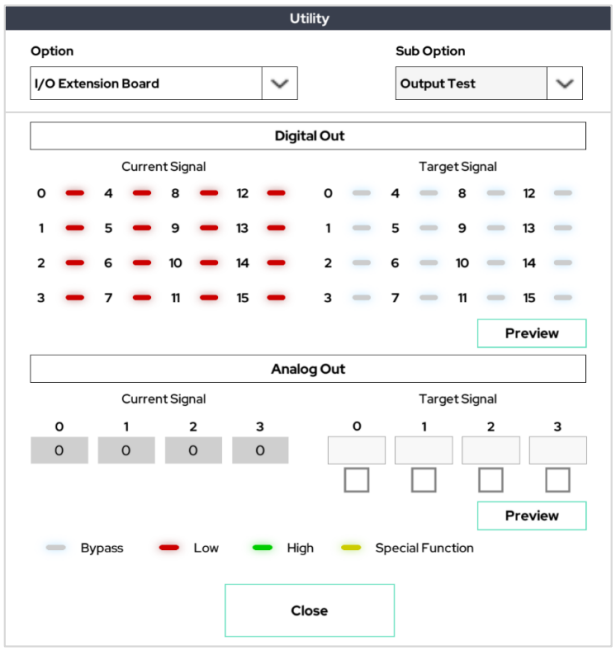
Dies ist das Fenster, in dem die Ausgabe des Werkzeugflansches getestet werden kann.

© RAINBOW ROBOTICS Inc. All rights reserved.

[Utility-I/O Extension Board]



Ein/Aus-Signalüberwachungsfenster des Ein/Aus-Erweiterungsmoduls.

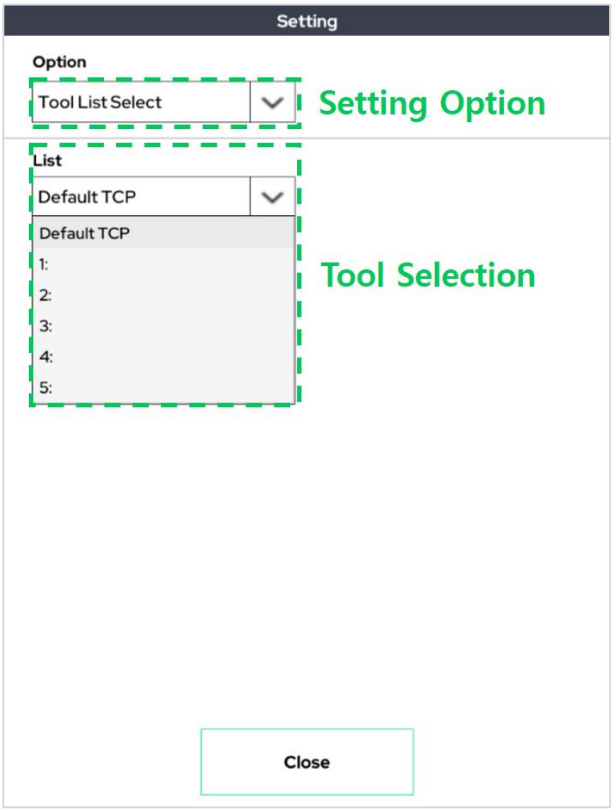


Fenster zum Testen des Ausgangs des Ein/Aus-Erweiterungsmoduls.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Unterfunktion von Setting (Einstellung)

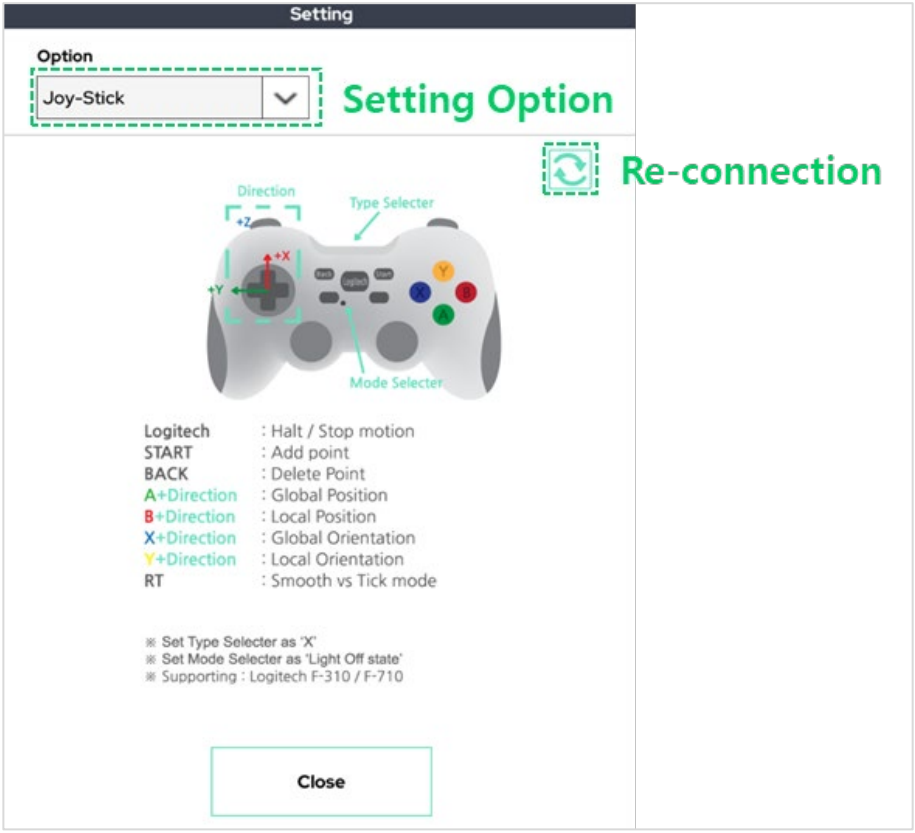
[Setting-Tool List Select]



Es gibt eine Einrichtungsfunktion zur ‘Tool List Select (Werkzeuglistenauswahl)’, die TCP zur Verwendung in einer vorab gespeicherten TCP-Liste einstellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

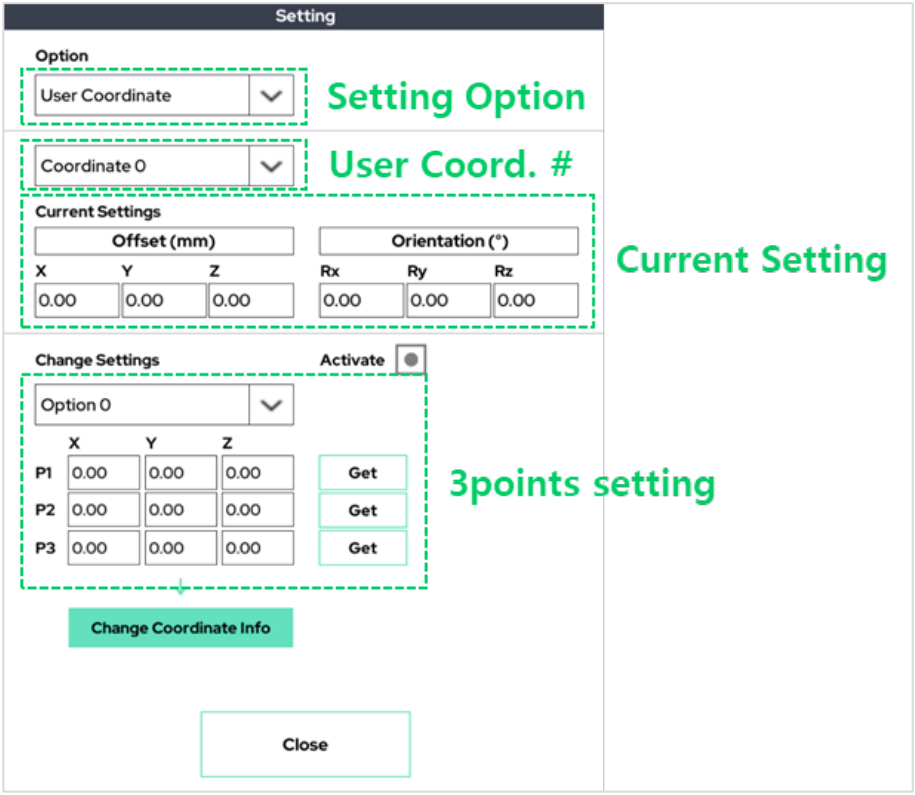
[Setting-Joystick]



Bei Teaching des Roboters mit einem kabelgebundenen/drahtlosen Joystick gibt es eine Funktion zum Einstellen der Joystick-Verbindung.

© RAINBOW ROBOTICS Inc. All rights reserved.

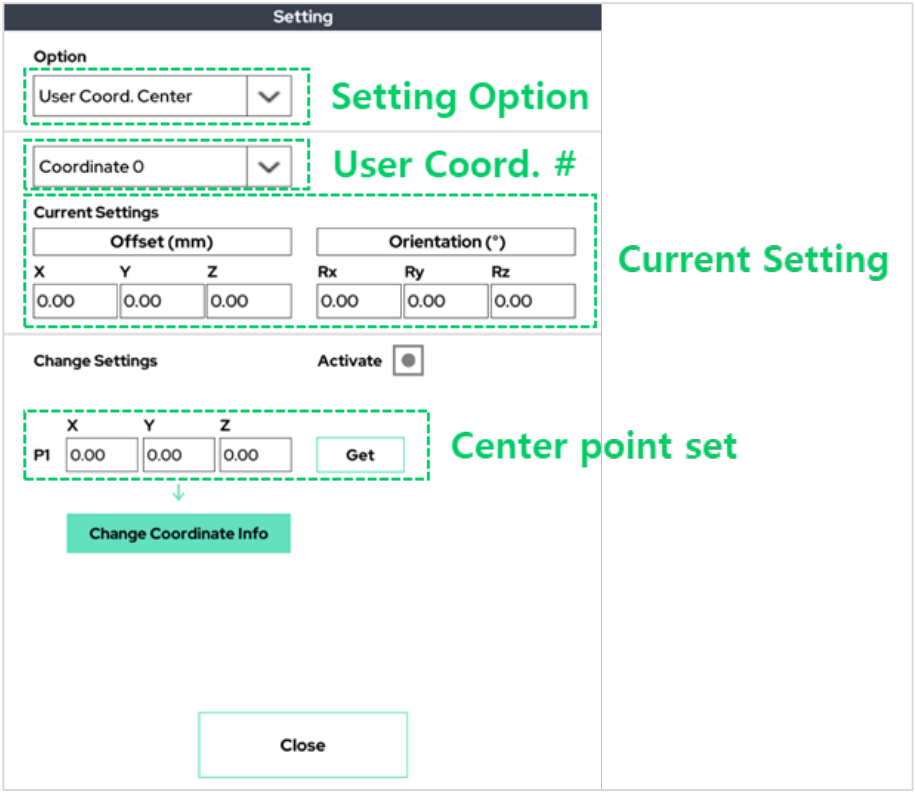
[Setting-User Coordinate]



Benutzerdefiniertes Koordinatensystem wird eingestellt. Es handelt sich um einen 3-Punkt-Einstellmodus (Siehe Setup-Coordinate).

© RAINBOW ROBOTICS Inc. All rights reserved.

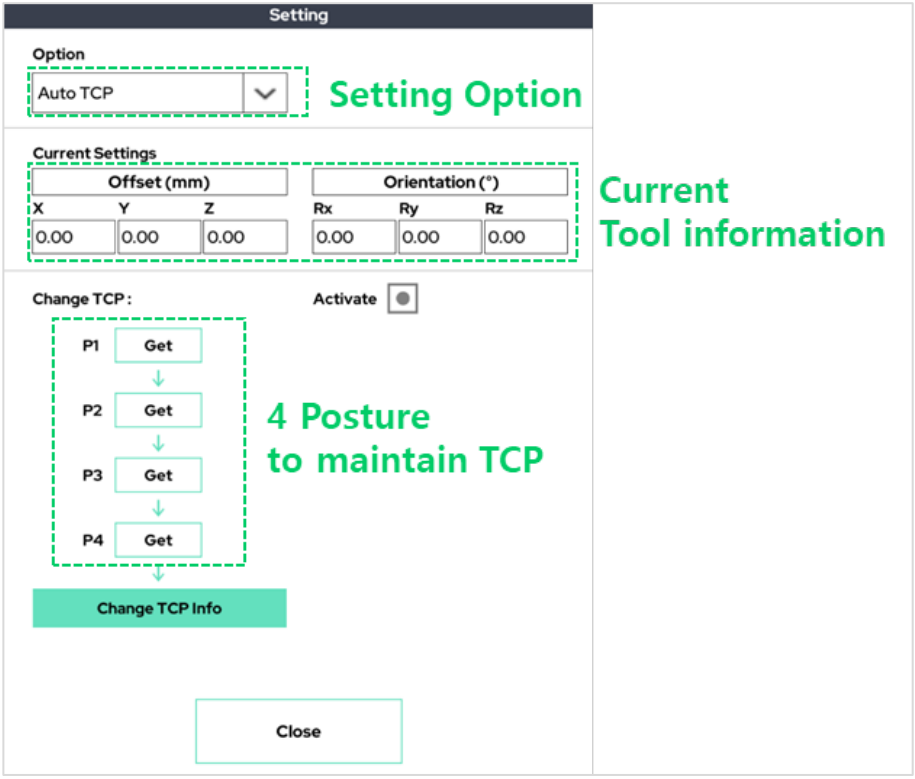
[Setting-User Coordinate Center]



Rotationswert des benutzerdefinierten Koordinatensystems wird behalten und nur der Mittelpunkt (Offset) wird eingestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

[Setting-Auto TCP]

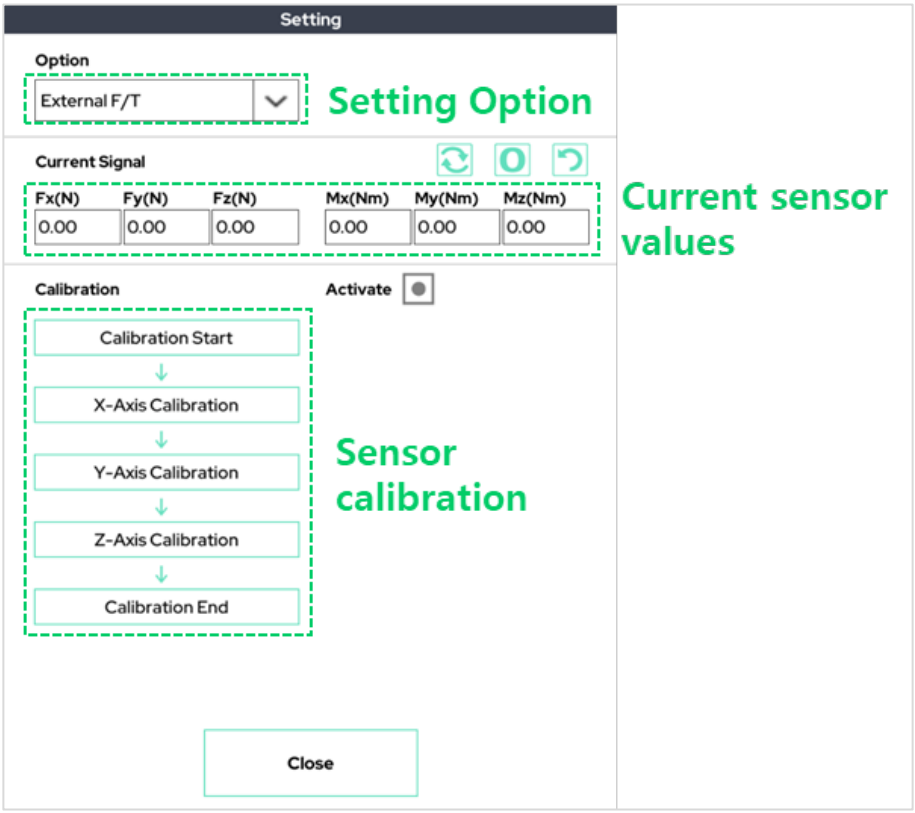


Diese Funktion findet automatisch die Position des Roboterendes (TCP).

Wenn Sie 4 verschiedene Körperhaltungen eingeben, die den TCP-Punkt als denselben Punkt in der 3D-Dimension beibehalten, wird der TCP-Positionswert automatisch berechnet.

© RAINBOW ROBOTICS Inc. All rights reserved.

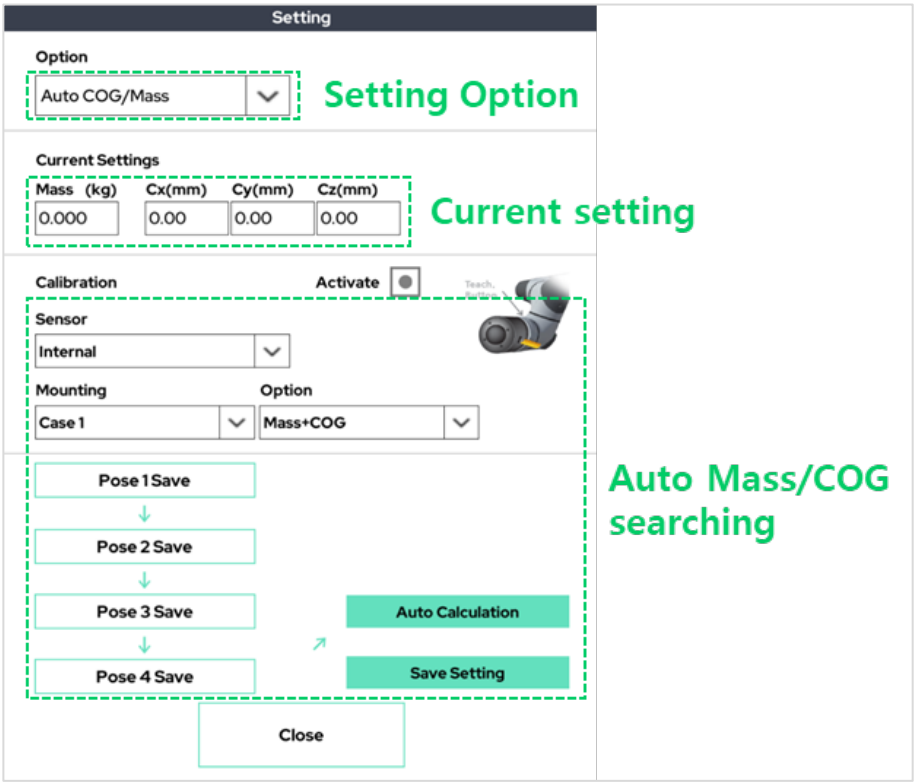
[Setting-External F/T]



In diesem Fenster kann der Wert des externen F/T-Sensors (z. B.: Robotiq F/T-Sensor) überprüft und kalibriert werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

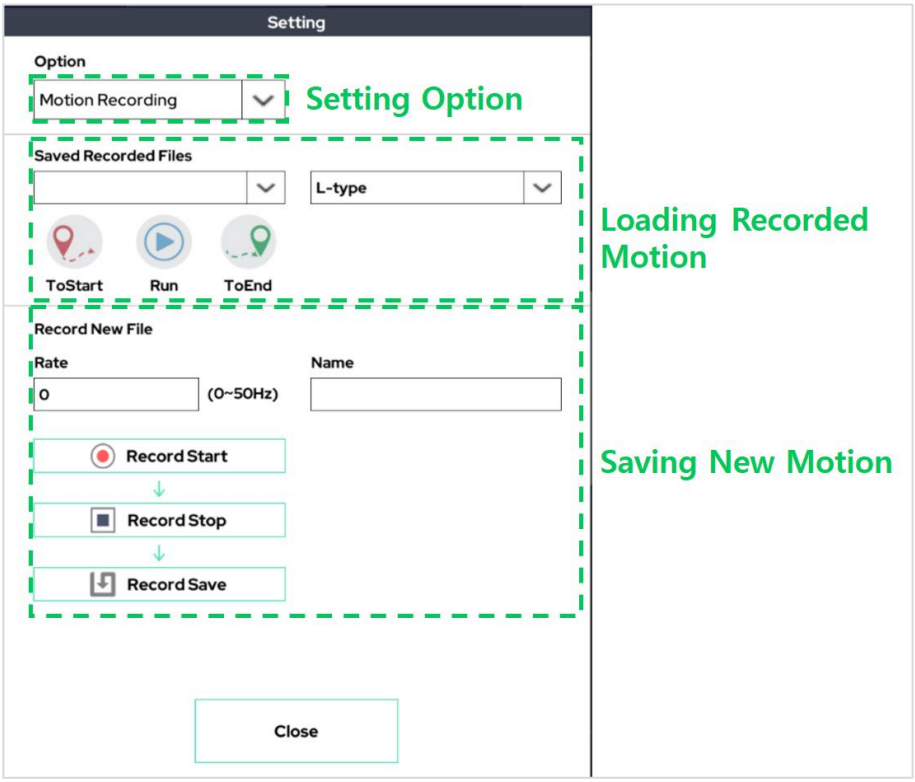
[Setting-Auto COG / Mass]



Die Funktion, die den internen/externen F/T-Sensor verwendet, um das Gewicht und den Schwerpunkt des Werkzeugs zu ermitteln.

© RAINBOW ROBOTICS Inc. All rights reserved.

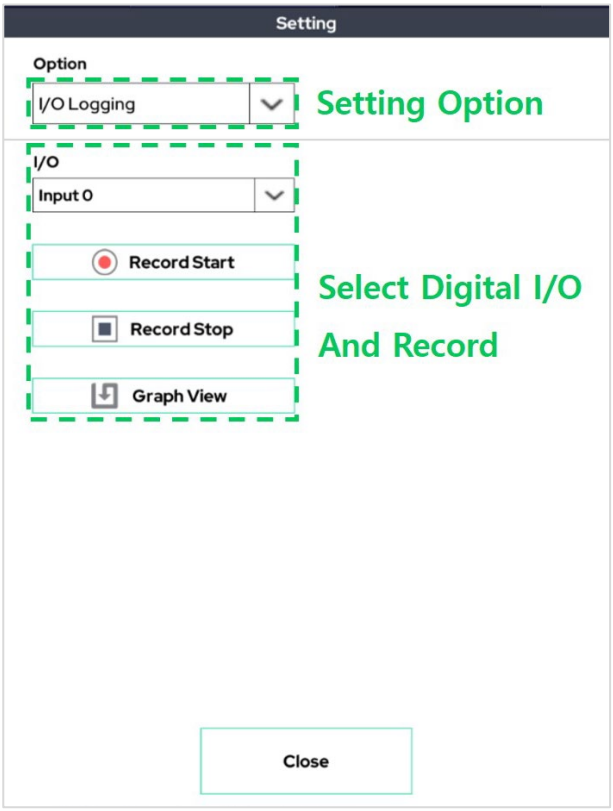
[Setting- Motion Recording]



Mit dieser Funktion wird die Bewegung durch die Direkt-Teaching-Funktion (Schwerkraftkompensation) aufgezeichnet. Die aufgezeichnete Bewegung ist im Programm über die Replay-Funktion verfügbar.

© RAINBOW ROBOTICS Inc. All rights reserved.

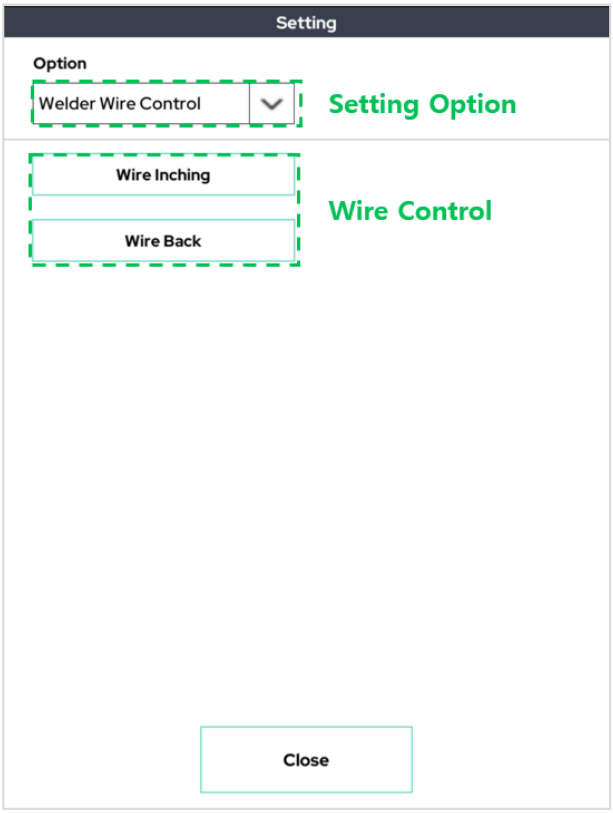
[Setting-I/O Logging]



Mit dieser Funktion wird ein digitaler Eingang/Ausgang ausgewählt und werden die Änderungen des Eingang-/Ausgangswerts aufgezeichnet und als Diagramm angezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.

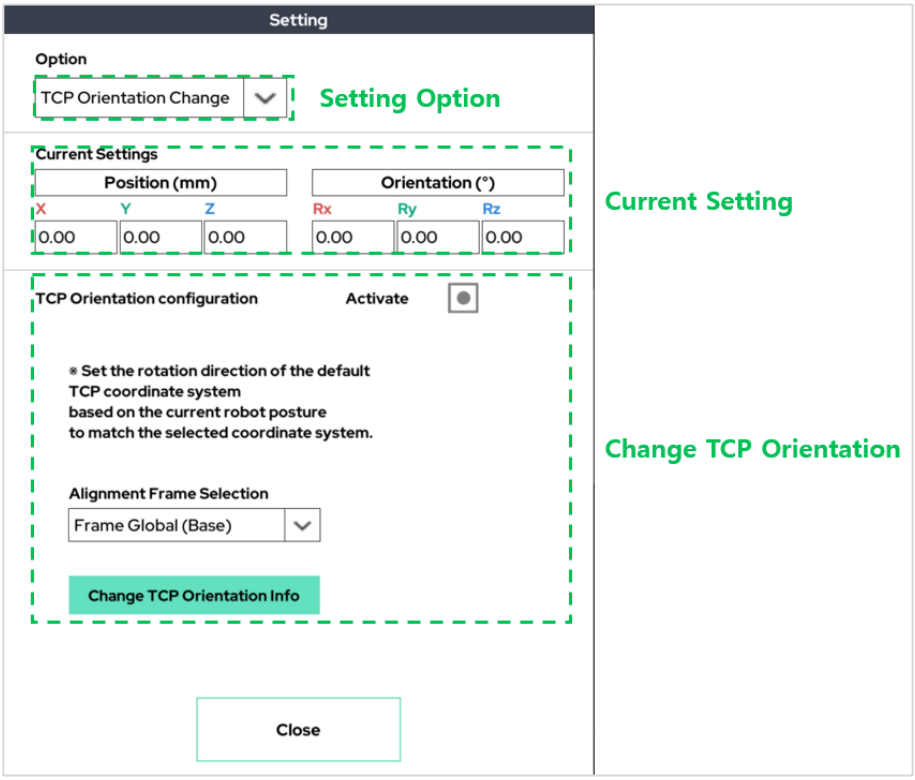
[Setting-Welder Wire Control]



Mit dieser Funktion wird der Draht der Schweißmaschine gesteuert.

© RAINBOW ROBOTICS Inc. All rights reserved.

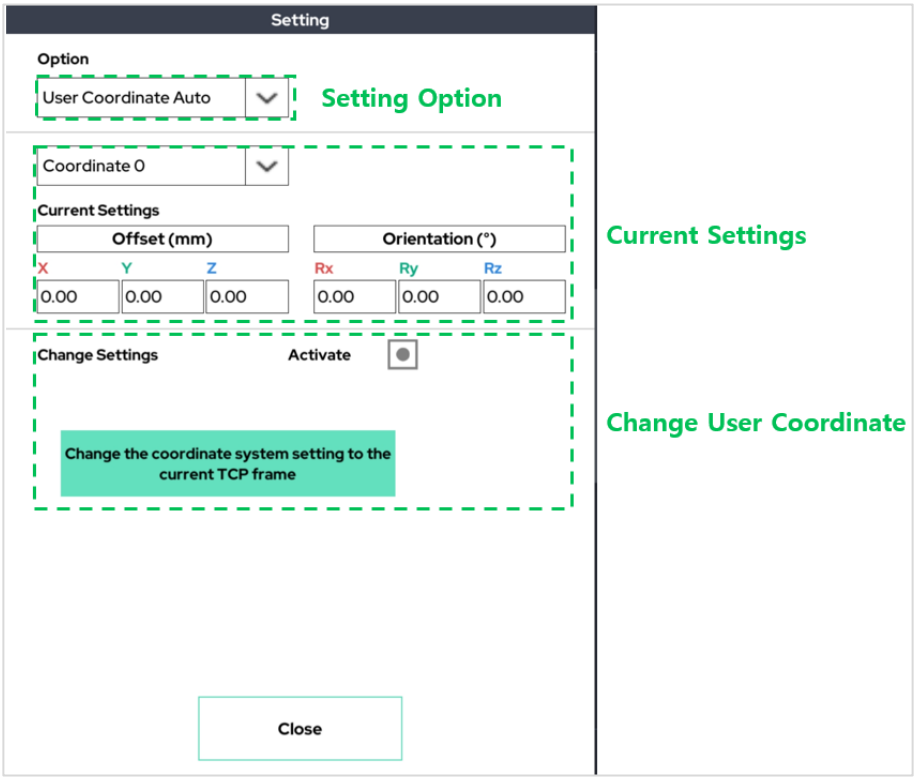
[Setting-TCP Orientation Change]



Die Rotationsrichtung des Standard-TCP-Koordinatensystems wird basierend auf der aktuellen Roboterposition festgelegt, sodass sie mit dem ausgewählten Koordinatensystem übereinstimmt.

© RAINBOW ROBOTICS Inc. All rights reserved.

[Setting-User Coordinate Auto]



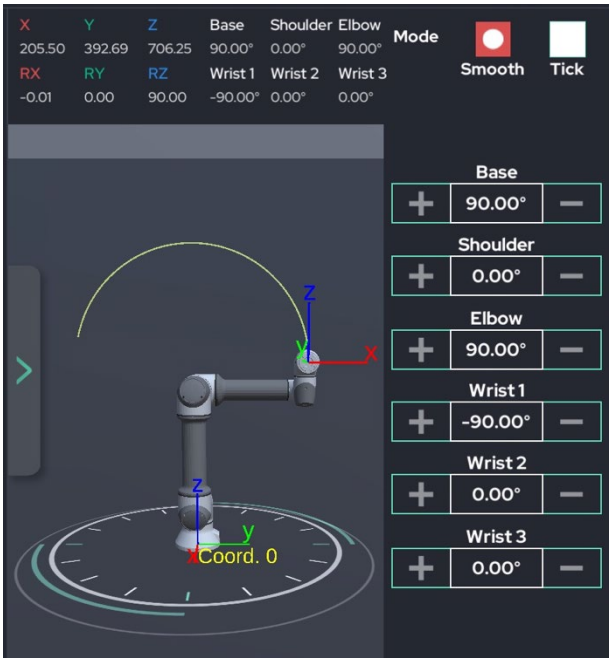
Das aktuelle TCP-Koordinatensystem des Roboters wird als benutzerdefiniertes Koordinatensystem eingestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Monitorfunktion

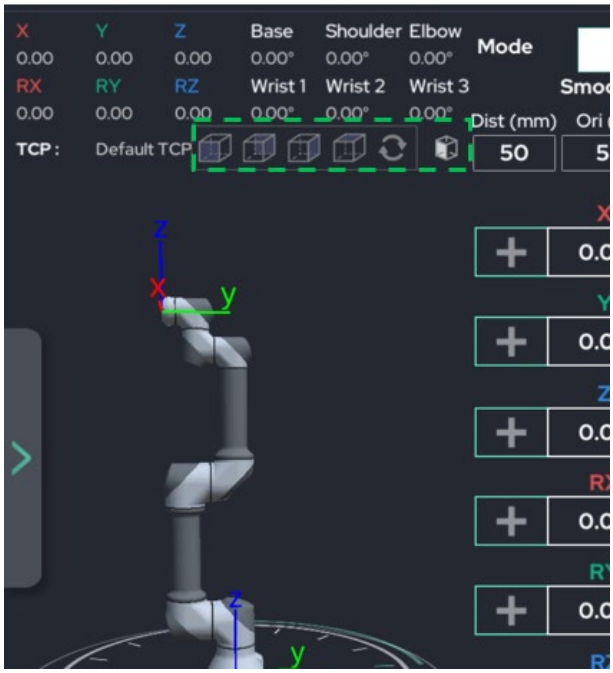
Die Funktion, die zusammen mit dem Monitor-Befehl im Kapitel 6 verwendet wird, bietet ein Fenster, in dem Sie die Werte von Systembasisvariablen und benutzerdefinierten Variablen in Echtzeit beobachtet werden.

Wenn Sie die Aufnahmefunktion oben rechts einschalten, wird die TCP-Trajektorie des Roboterwerkzeugs im 3D-Viewer aufgezeichnet (mit der gelben Linie)



© RAINBOW ROBOTICS Inc. All rights reserved.

■ 3D-Ansichtsfunktion



Es wird ein häufig verwendeter dreidimensionaler Betrachtungswinkel angegeben. Wenn Sie die Perspektive des 3D-Viewers schnell ändern möchten, kann diese Funktion verwendet werden.

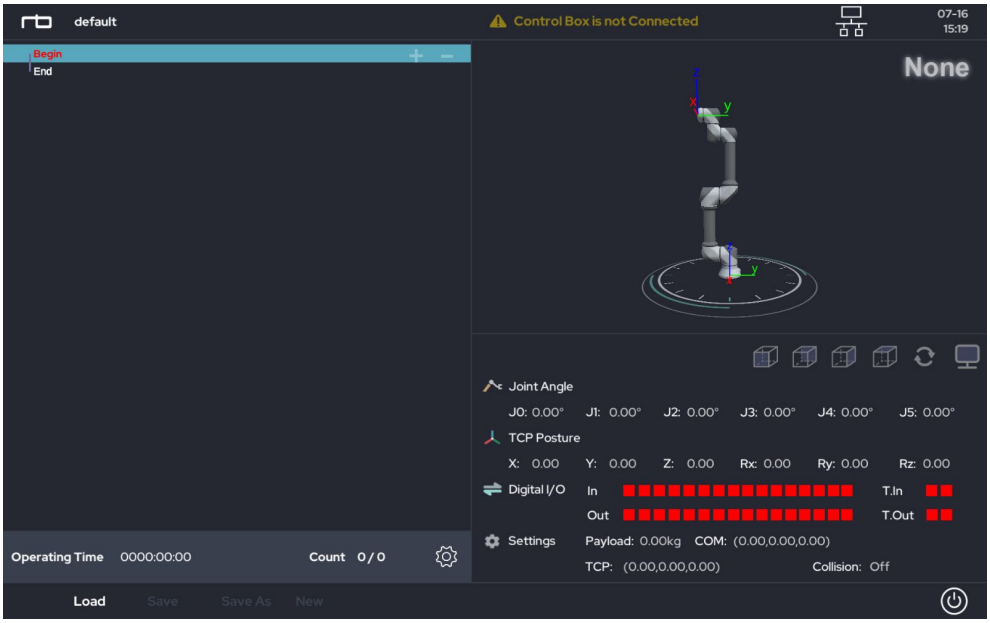
© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 8 Starten von Roboter

8.1 Antreiben von Roboter

Um den Roboter auszuführen, gehen Sie zum Play-Bildschirm.

- Der Ausführungsbildschirm sieht wie folgt aus.



.

- Rufen Sie ein gespeichertes Projekt auf. Siehe 6.6 für den Projektimport.
- Wenn alle Einstellungen abgeschlossen sind, drücken Sie die PLAY-Schaltfläche unten auf dem Bildschirm.
- Wenn sich der Status des Roboters nicht in der Ausgangsposition befindet, erscheint ein Führungsfenster. Drücken Sie die approach-Schaltfläche, um den Roboter in die Ausgangsposition zurückzusetzen.
- Wenn Sie den Roboter auf dem Anführungsbildschirm fahren, wird das laufende Projekt grundsätzlich unbegrenzt wiederholt. Sie können die Anzahl der Wiederholungen festlegen, indem Sie oben auf die Count-Schaltfläche drücken.
- Die Geschwindigkeit kann während des Betriebs angepasst werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

Warnung

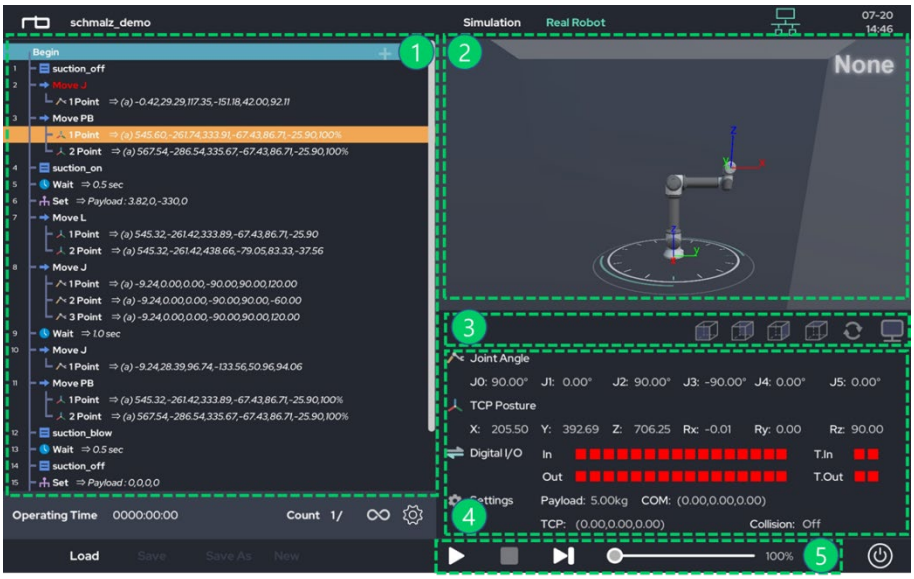
Warnung:

1) Der Benutzer muss eine ausreichende Risikobewertung des Robotersystems durchführen, bevor er den Roboterarm bedient, dann alle anwendbaren Sicherheitsfunktionen auf den Roboter anwenden und den Roboterarm betreiben.
2) Wenn während des Initialisierungsprozesses ein Problem mit dem Roboter rauftriff oder der Roboterinstallationswinkel und die Lasteinstellung falsch sind, kann die Roboterinitialisierung fehlschlagen.
3) Im Play-Bildschirm bewegt sich der Roboter ohne Simulation. Bitte machen Sie sich mit der Verwendung vertraut.
4) Während des Betriebs ist es nicht möglich, zu anderen Bildschirmen wie Make- oder Setup-Bildschirm zu wechseln. Um den Bildschirm zu ändern, verwenden Sie ihn, nachdem das laufende Programm vollständig gestoppt wurde.
5) Es ist in Ordnung, die Tablet-UI während der Fahrt heranzuziehen.

© RAINBOW ROBOTICS Inc. All rights reserved.

8.2 Prüfung des Roboterstatus

Der Status des aktuell laufenden Roboterarms wird auf dem Ausführungsbildschirm angezeigt.



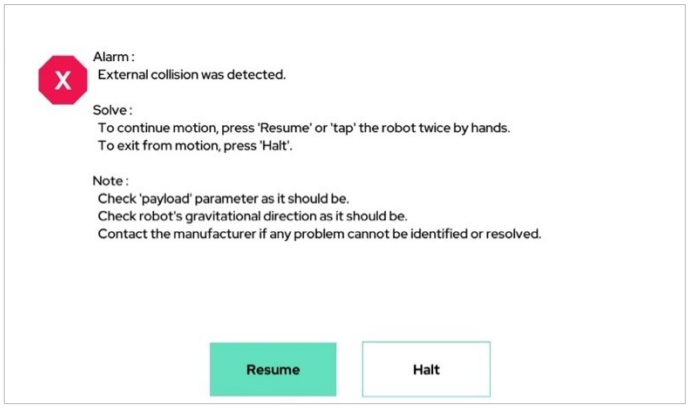
Nr.	Beschreibung
①	Viewer für Programmablauf
②	3D-Viewer
③	Die Perspektive des 3D-Viewer wird angepasst.
④	Statusinformation und Variablen des Systems werden angezeigt.
⑤	Wiedergabe-/Pause-/Geschwindigkeitssteuerleiste sind vorhanden.

© RAINBOW ROBOTICS Inc. All rights reserved.

8.3 Umgang mit ungewöhnlichen Ereignissen während des Betriebs

Während des Roboterbetriebs können verschiedene Problemsituationen auftreten, und es gibt entsprechende Gegenmaßnahmen.

1. Externe Kollision (Out Collision)



Wenn der Roboter während des Betriebs mit einem externen Objekt kollidiert, hält der Roboter den Betrieb an.

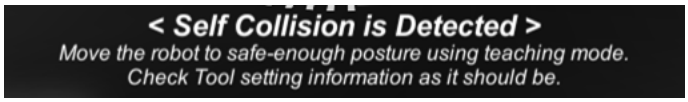
Zunächst entfernen Sie externe Konfliktfaktoren. Danach haben Sie zwei Möglichkeiten. Um die Arbeit des Roboters anzuhalten und Wartungsarbeiten durchzuführen, drücken Sie die Halt-Taste, um die Arbeit zu beenden. Um die Arbeit des Roboters fortzusetzen, drücken Sie die Resume-Taste, um die Arbeit fortzusetzen.

TOK TOK-Wiederherstellungsmethode

Wenn Sie aufgrund einer externen Kollision angehalten werden, tippen Sie zweimal leicht auf das distale Ende des Roboterarms, um den Pausezustand wiederherzustellen
(Gleiche Wirkung wie die Resume-Schaltfläche).

© RAINBOW ROBOTICS Inc. All rights reserved.

2. Selbstkollision (Self Collision)



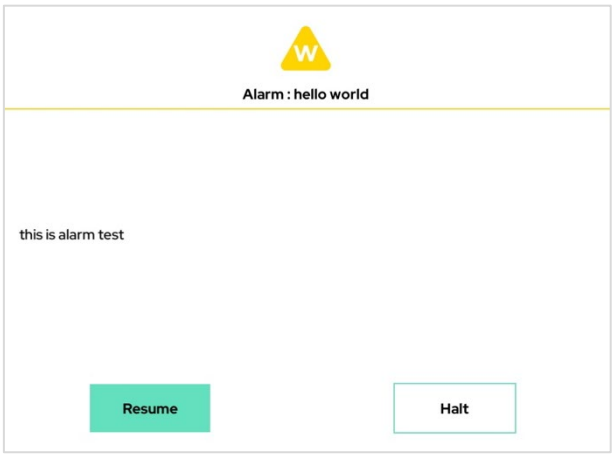
Der Roboter hört auf zu arbeiten, wenn sich ein Teil des Roboters außerhalb des angegebenen Betriebsbereichs befindet oder wenn sich der Roboter selbst einer Bewegung nähert, die ein Kollisionsrisiko darstellt.

Sie müssen die Teaching-Taste verwenden, um den Zustand des Roboters durch die Direkt-Teaching-Funktion zu normalisieren. Überprüfen Sie anschließend die eingelernten Bewegungs- und Betriebsbereichseinstellungen.

Wenn es im Simulationsmodus auf dem Make-Bildschirm auftritt, kann es auf folgende Weise wiederhergestellt werden.

- Den Roboter von Selbstkollision durch die bewegungsbezogenen Tasten des Roboters befreien
- Den Roboter vorübergehend in Real-Modus wechseln, um die Gelenkinformationen des echten Roboters zu empfangen und wiederherzustellen
- Die Teaching-Taste drücken, um die Gelenkinformationen des echten Roboters zu empfangen und wiederherzustellen

3. Alarmmeldung

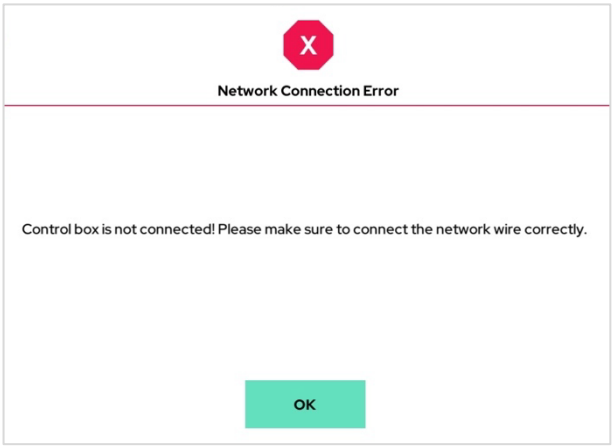


Wenn die Alarmmeldungsfunktion ausgeführt wird, hält der Roboter den Betrieb an, während das Programm ausgeführt wird.

Drücken Sie die Resume-Taste, um mit dem Vorgang fortzufahren, oder drücken Sie die Halt-Taste, um den Vorgang zu stoppen.

© RAINBOW ROBOTICS Inc. All rights reserved.

4. Trennen des Tablet-PCs



Wenn die Steuerbox und der Tablet-PC getrennt werden, tritt das Fenster auf.

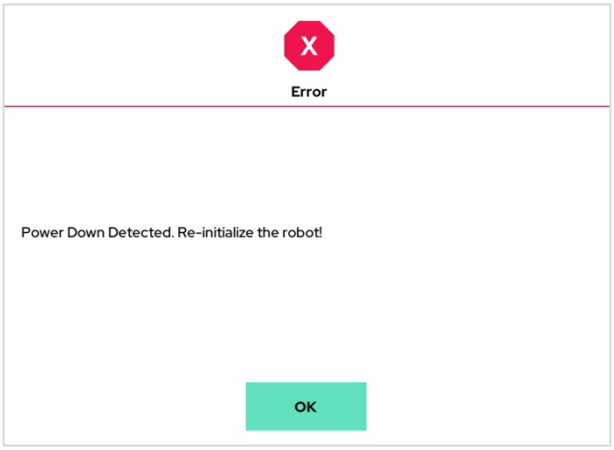
Wenn das USB-Kabel versehentlich getrennt wurde, schließen Sie das Kabel wieder an und befolgen Sie die Anweisungen, um den Tablet-PC wieder mit der Steuerbox zu verbinden.

Wenn das Kabel beschädigt ist, ersetzen Sie das Kabel und schließen Sie den Roboter gemäß dem Anleitungsfenster erneut an.

Wenn die Verbindung immer noch nicht hergestellt wird, versuchen Sie es mit einem anderen USB-Anschluss.

© RAINBOW ROBOTICS Inc. All rights reserved.

5. Unterbrechung der externen Stromversorgung



Wenn die Stromversorgung des Roboterarms unterbrochen wird, tritt das Fenster auf. Wenn der AC-Anschluss der Robotersteuerbox gewaltsam getrennt oder der Not-Aus-Schalter gedrückt wird, wird das Fenster angezeigt.

Nachdem die Stromversorgung der Steuerbox wiederhergestellt oder der Not-Aus-Schalter in den Normalzustand zurückversetzt und der Roboter erneut initialisiert wurde, kann der Roboter verwendet werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

6. Fehlererkennung des gemeinsamen Controllers

   Wenn die gemeinsame Steuerung feststellt, dass es schwierig ist, den Roboter zu steuern, wird der Roboter angehalten.

   Die folgenden Artikel sind vorhanden.

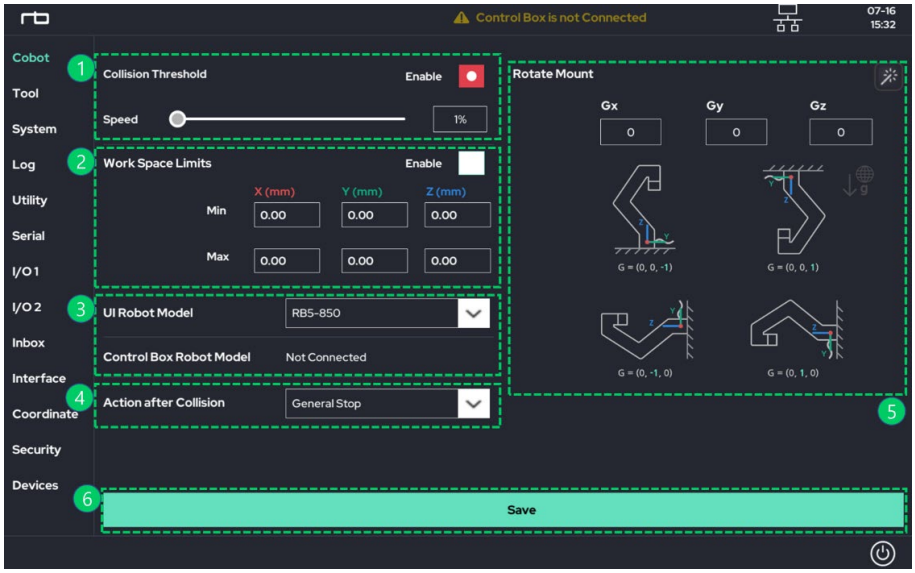
Big Error	Wenn die Differenz zwischen der Eingangsreferenz und dem Encoder einen bestimmten Wert überschreitet, tritt das auf.
Jam Error	Wenn sich der Encoderwert nicht ändert, aber ein Strom über einem bestimmten Wert für eine bestimmte Zeit oder länger fließt, tritt das auf
Overcurrent Error	Wenn ein Strom über einen bestimmten Wert über eine bestimmte Zeit fließt, tritt das auf
Temperature Error	Wenn die Temperatur der Steuerplatine einen bestimmten Wert überschreitet, tritt das auf
Mode Error	Wenn der von der Hauptsteuerung befohlene Steuerungsmodus nicht mit dem Modus der aktuellen gemeinsamen Steuerung übereinstimmt, tritt das auf

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel 9 Einstellungen

9.1 Set-up (Cobot)

Die Standardumgebung für den Roboterarm kann festgelegt werden.

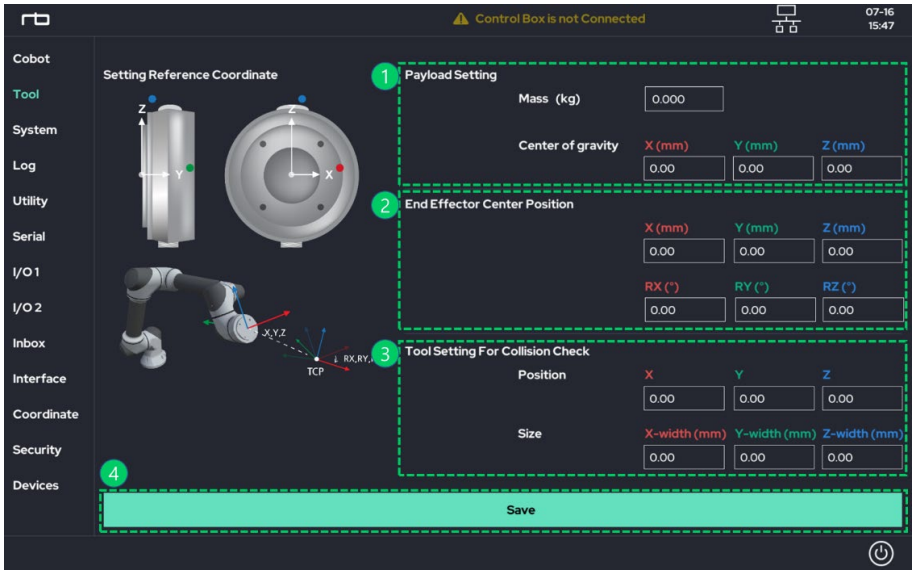


①	Die Kollisionsempfindlichkeit wird eingestellt. Ja niedriger die Empfindlichkeit, desto leichter Aufprall führt dem Roboter zum Stoppen. Kollisionserkennungsfunktion kann ein-/ausgeschaltet werden.
②	Der Betriebsbereich des Roboters wird festgelegt. Wenn ‚Enable Work Space' aktiviert ist, wird der Aktivitätsgrenzbereich in 3D angezeigt. Verlässt er den aktiven Bereich, wird dies als Selbstkollision erkannt und der Roboter stoppt. Diese Funktion kann ein-/ausgeschaltet werden.
③	Das derzeit in Betrieb befindliche UI-Robotermodell und das Robotermodell der angeschlossenen Steuerbox werden angezeigt.
④	Der Stoppmodus wird nach einem Absturz gewählt. Wählen Sie zwischen allgemeinem Stopp(General Stop) und Ausweichstopp(Evasion Stop).
⑤	Der Installationswinkel des Roboters wird eingestellt. Wenn eines der gezeigten Beispielbilder einen Installationswinkel hat, klicken Sie auf das Bild. Zur direkten Eingabe geben Sie den Richtungsvektor der Erdbeschleunigung basierend auf dem globalen Koordinatensystem ein.
⑥	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.2 Set-up (Tool)

Einstellungen für installierte Werkzeuge können konfiguriert werden.

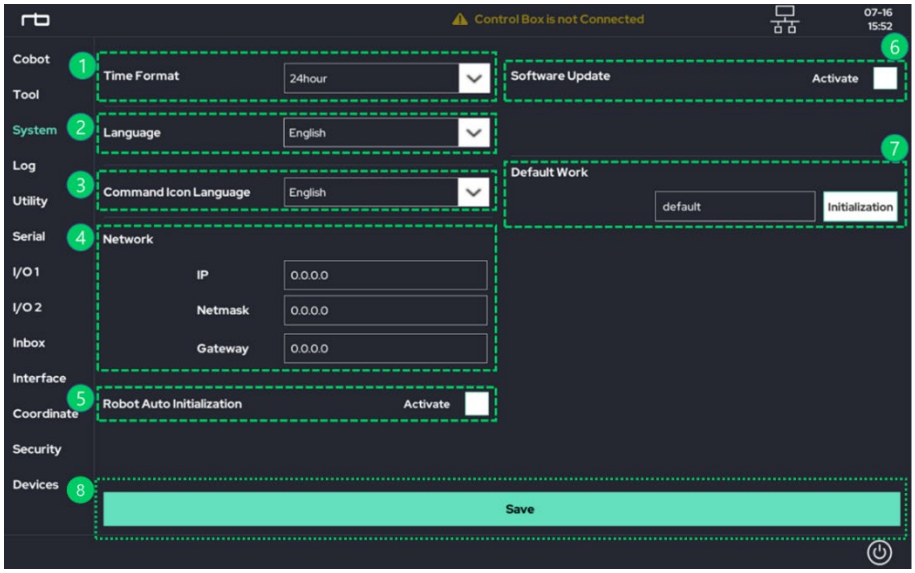


①	Geben Sie die Informationen der virtuellen Box ein, die am Werkzeugende montiert ist, um eine Selbstkollision zu verhindern. Geben Sie die Größe der Box und die Position des Mittelpunkts der Box ein Die obigen Einstellungen basieren auf dem Standard-Koordinatensystem des Herstellers, das auf dem Werkzeugflansch montiert ist.
②	Stellen Sie die auf den Roboter aufgebrachte Last ein. Die Eingabekategorie für die Masse unterscheidet sich je nach Nutzlast jedes Robotermodells in der Spalte ‚Mass' (kg). Geben Sie im 'Center of gravity'-Feld den Abstand zum Schwerpunkt (mm) ein.
③	Legen Sie die Position und Ausrichtung des Koordinatensystems des Endes vom Roboterarm (TCP) fest. Das Koordinaten des Endes vom Roboterarm legt fest, um wie viel in Euler-Winkeln im standardmäßigen Koordinatensystem des Herstellers gedreht werden soll. Die obigen Einstellungen basieren auf dem Standard-Koordinatensystem des Herstellers, das auf dem Werkzeugflansch montiert ist.
④	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.3 Set-up (System)

Anzeigeeinheiten, Datum und Uhrzeit, UI-Passwort, Systemaktualisierung und mehr werden eingestellt.

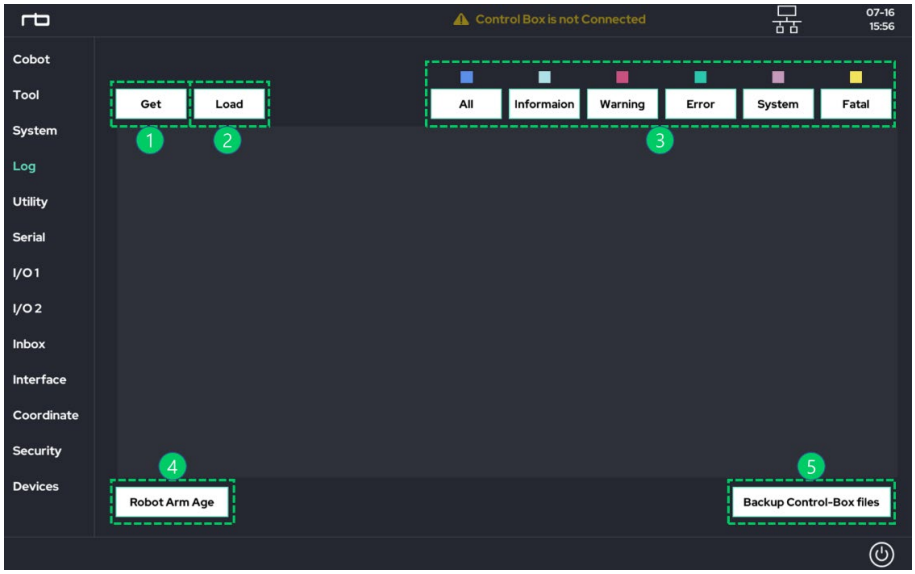


①	Die Datumsnotation für UI-Programme wird festgelegt.
②	Die Sprache wird eingestellt.
③	Die Sprache des Befehlsymbols wird festgelegt.
④	Die IP-Informationen der Steuerbox werden eingestellt. Die hier eingestellte IP wirkt sich auf die Socket-Funktion, die Modbus-Funktion und die externe Skript-Steuerungsfunktion aus.
⑤	Automatische Aktivierungsfunktion des Roboterarms. Wenn diese Funktion verwendet wird, fährt der Roboterarm automatisch mit der Aktivierung fort, wenn eine Aktivierung möglich ist. Der automatisch aktivierte Roboterarm ist der Standard-Real-Modus. Um diese Funktion zu verwenden, muss die Auto-Initialisierungstastenfunktion im digitalen Eingang der Steuerbox definiert werden und funktioniert, wenn eine Eingabe an der entsprechenden Taste vorhanden ist (Redundanz).
⑥	Diese Funktion dient zum Aktualisieren der Steuerbox-Software. Siehe Anhang (Software Update).
⑦	Legt das Projekt fest, für das das Steuerfeld und die Benutzeroberfläche beim ersten Start standardmäßig geöffnet (load) werden.
⑧	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.4 Set-up (Log)

Der Systemprotokollstatus des Roboterarms kann überprüft werden.

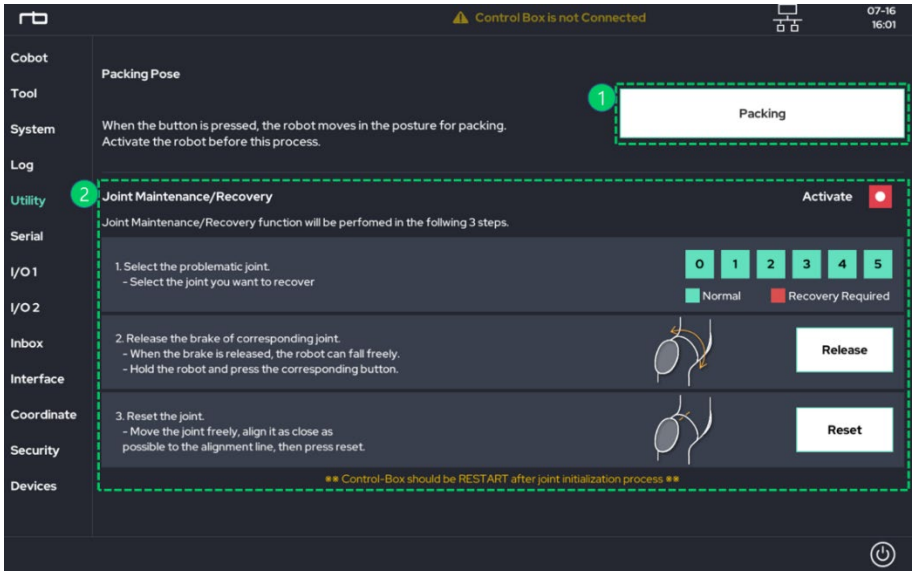


①	Protokolldateien(Log File) werden aus der Steuerbox auf den Tablet-PC kopiert. Je nach Größe der zu kopierenden Datei dauert das Kopieren einige Sekunden.
②	Die Funktion zum Öffnen der Protokolldatei, die durch den Prozess 1 kopiert wurde.
③	Eine Funktion zum Anzeigen von Protokollen nach Typ durch Filter. I : Information W : Warning E : Error S : System F : Fatal
④	Eine Funktion zum Überprüfen der Statusinformationen des Roboterarms
⑤	Mit dieser Funktion werden Programmdatei/ Protokolldateien/ Einstellungsdateien auf der Steuerbox auf den Tablet-PC kopiert. Die kopierten (gespeicherten) Dateien werden in einem bestimmten Pfad auf dem Tablet-PC gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.5 Set-up (Utility)

Funktionen im Zusammenhang mit dem Verpacken und Notfallwiederherstellung von Robotern werden geboten.



①	Sie können Haltung des Roboters für das Packen erstellen. Vor Verwendung dieser Funktion muss der Roboterarm aktiviert werden. Entfernen Sie alle Anbauteile vom Roboterarm, bevor Sie diese Funktion verwenden. Während die Taste gedrückt wird, bewegt sich der Roboterarm in die Gehsteigposition. Wenn die Gehrsteigposition erreicht ist, wird ein Alarm angezeigt, der anzeigt, dass die Gehsteigposition erreicht wurde.
②	Sie können den Bezugspunkt der Achse auf die Werkeinstellung zurücksetzen. Diese Funktion sollte mit Vorsicht verwendet werden, da es sich um eine Wiederherstellungsfunktion handelt, wenn der Roboter anormal arbeitet. Step1. Stellen Sie die zurückzusetzende Achse ein Step2. Drücken Sie die Release-Taste, um die Bremse zu lösen und freizugeben. Step3. Richten Sie die markierten Rillen auf der Achse aus und drücken Sie die Reset-Taste, um die Achse zurückzusetzen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Vorsicht

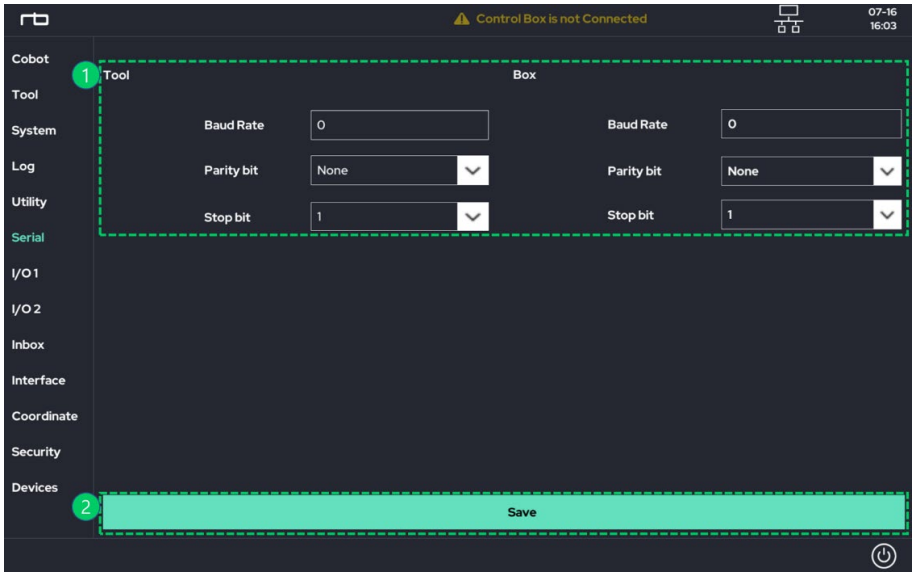
Vorsicht:

1) Bevor Sie den Emergency Joint Recovery-Modus verwenden, machen Sie sich mit den Eigenschaften und dem Betrieb dieser Funktion vertraut.
2) Beim Transportieren des Roboters muss dieser In der vom Hersteller bereitgestellten Verpackungsbox in die Verpackungsposition gebracht werden.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.6 Set-up (Serial)

Die serielle Kommunikation zwischen dem Roboterwerkzeug und der Steuerbox kann eingerichtet werden.

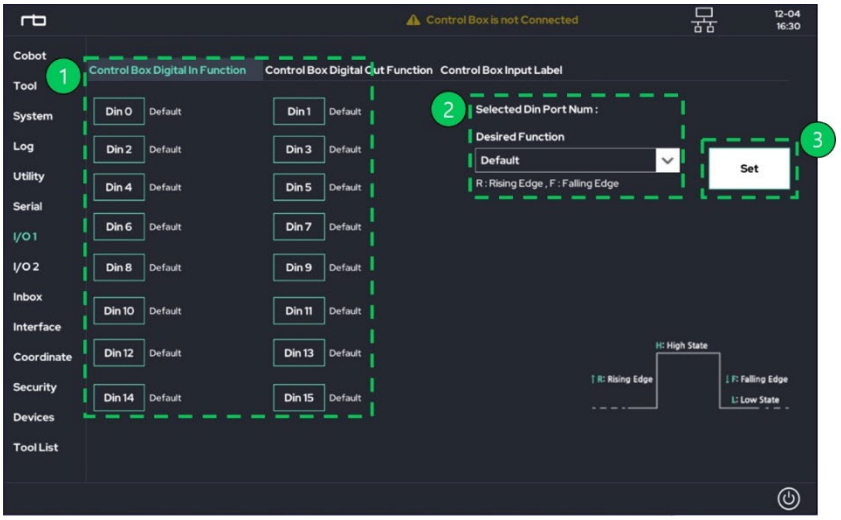


①	RS485-Kommunikationsoption wird eingestellt. Baudrate, Stoppbit und Parity bit werden festgelegt.
②	Die geänderten Einstellungen werden gespeichert.

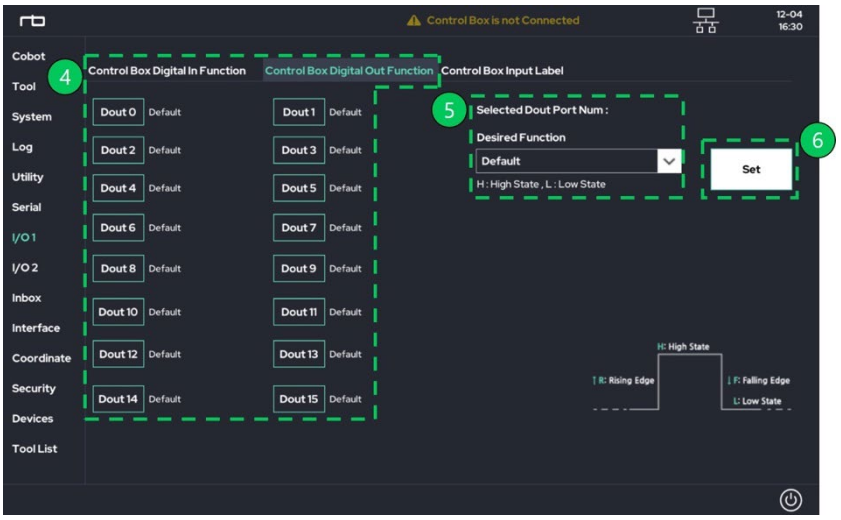
© RAINBOW ROBOTICS Inc. All rights reserved.

9.7 Set-up (Ein/Aus 1)

Die Funktion des GPIO-Anschlusses der Steuerbox kann eingestellt werden.

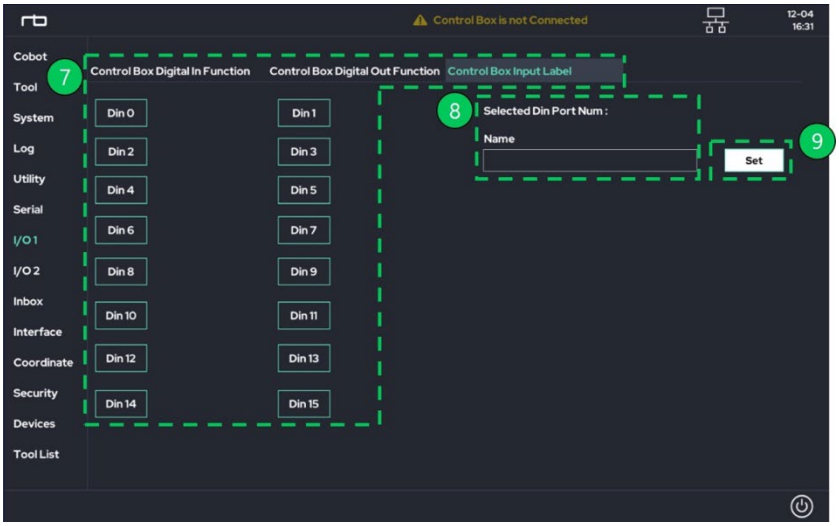


①	Wählen Sie den Eingangsanschluss aus, dessen Funktion Sie ändern möchten.
②	Wählen Sie, welche Funktion Sie ändern möchten. Beschreibungen der einzelnen Funktionsoptionen finden Sie auf den folgenden Seiten.
③	Ausgewählte Funktion werden gespeichert.

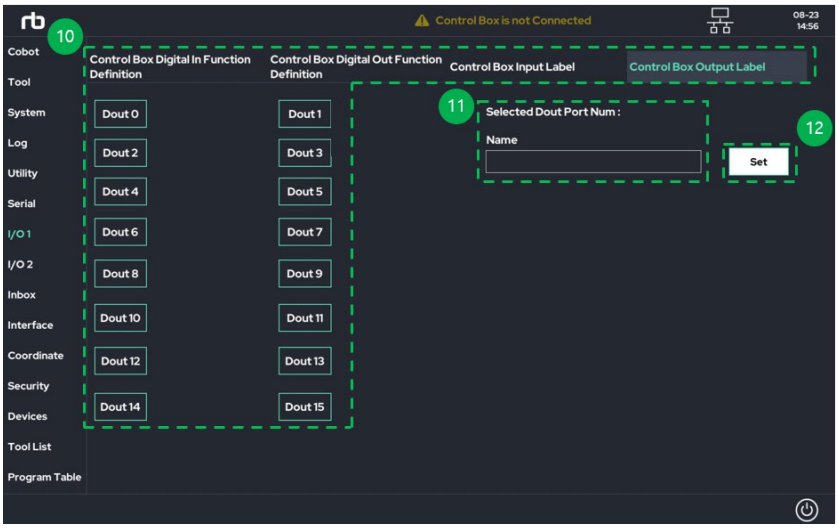


④	Wählen Sie den Ausgangsanschluss aus, dessen Funktion Sie ändern möchten.
⑤	Wählen Sie, welche Funktion Sie ändern möchten. Beschreibungen der einzelnen Funktionsoptionen finden Sie auf den folgenden Seiten.
⑥	Ausgewählte Funktion werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.



⑦	Wählen Sie den Eingangsanschluss aus, den Sie benennen möchten.
⑧	Geben Sie den Namen ein, den Sie verwenden möchten.
⑨	Speichern Sie unter einem Namen Ihrer Wahl.



⑩	Wählen Sie den Ausgangsanschluss aus, den Sie benennen möchten.
⑪	Geben Sie den Namen ein, den Sie verwenden möchten.
⑫	Speichern Sie unter einem Namen Ihrer Wahl.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Eingangsanschlussfunktion der Steuerbox

Die Funktion der Eingangsanschlüsse (Din 0 - Din 15) der Steuerbox können mit den folgenden Sonderfunktionen eingestellt werden (R = Rising Edge, F = Falling Edge, H = High State).

0. Basismodus (GPIO)
1. Einmaliger Programmstartmodus (Rising Edge)
2. Programm anhalten (Rising Edge)
3. Das Programm unterbrechen (Rising Edge)
4. R = Direkt-Teaching-Modus Ein / F = Direkt-Teaching.Modus Aus
5. R = Geschwindigkeit 100% / F = Geschwindigkeit 0%
6. R = zu Real-Modus / F = zu Simulation-Modus
7. R = Roboter initialisieren
8. H = Bei High State, Kollionserkennung Aus
9. H = Auto-Initialisierung-Taste (Auto-Initialization Key)
10. R = Pause wegen externer Kollision beenden
11. Point-Funktion zum UI-Programm hinzufügen (Rising Edge)
12. Programmstart-Wiederholungsmodus (Rising Edge)
13. R= zu Begin-Position / F=Bewegung anhalten
14. R = Programm fortsetzen
15. H = Quick Freedrive Change
16. R = Programmpause / F = Programm fortsetzen
17. F = Programmpause / R = Programm fortsetzen
18. H = Geschwindigkeitsleiste 100% / L = Geschwindigkeitsleiste 0%
19. R = Standard-Setup-Programm laden
20. F = Roboterarm ausschalten
21. R = Touch Sensing
22. F = Touch Sensing
23. H = No Arc
24. H = Program Start Block
25. R = Ext.Joint0 Plus/F=stop
26. R = Ext.Joint0 Minus/F=stop
27. R = Ext.Joint1 Plus/F=stop
28. R = Ext.Joint1 Minus/F=stop
29. R = Ext.Joint2 Plus/F=stop
30. R = Ext.Joint2 Minus/F=stop
31. H = Safety Speed

© RAINBOW ROBOTICS Inc. All rights reserved.

32. F = UserCoord0 ←TCP frame
33. F = UserCoord1 ←TCP frame
34. F = UserCoord2 ←TCP frame
35. F = Load & Run Program Table

Vorsicht

Vorsicht:

1) Bevor Sie den digitalen Eingang verwenden, machen Sie sich bitte vollständig mit den elektrischen Eigenschaften des vom Hersteller bereitgestellten Digitaleingangsanschlusses vertraut.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Ausgangsanschlussfunktion der Steuerbox

Die Funktion der Ausgangsports (Din 0 - Din 15) der Steuerbox können mit den folgenden Sonderfunktionen eingestellt werden (R = Rising Edge, F = Falling Edge, H = High State).

Der als Sonderfunktion eingestellte Ausgangsport kann im Teaching-Programm nicht für allgemeine Zwecke verwendet werden.

0. Basismodus (GPIO)
1. H = Programm/Roboter in Betrieb / L = Idle-Zustand
2. L = Programm/Roboter in Betrieb / H = Idle-Zustand
3. H = Externe Kollision erkannt
4. H = Ausführungsstatus von Direkt-Teaching-Modus
5. Digitales Eingangssignal mit der Gleichen Nummer wie der ausgewählte Anschluss leiten
6. Signalübertragung vom Werkzeugflansch-Eingangsanschluss 0
7. Signalübertragung vom Werkzeugflansch-Eingangsanschluss 1
8. H = Roboter aktiviert / L = Roboter nicht aktiviert
9. H = in Real-Modus / L = in Simulation-Modus
10. H = Roboter in Betrieb / L = Idle-Zustand
11. L = Roboter in Betrieb / H = Idle-Zustand
12. H = Roboterinitialisierung (Aktivierung) misserfolgt
13. H = Roboterarm ein / L = Roboterarm aus
14. H = Kollisionserkennung an / L = Kollisionserkennung aus
15. H = Pause
16. H = Trap in Inbox 0 aufgetreten
17. H = Trap in Inbox 1 aufgetreten
18. Als PWM-Modul verwendet
19. H = Programmierhandgerät angeschlossen
20. H = Programm auf MAKE-Bildschirm ausgeführt
21. H = Programm auf PLAY-Bildschrim ausgeführt
22. H = Förderbetrieb-Modus erfolgt
23. H = Booten der Steuerbox abgeschlossen
24. H = Kraftsteuerungsmodus erfolgt
25. PC Alive Pulse
26. H = Geschwindigkeitsleiste 100%
27. H = Erfolgreiches Laden der letzten Programme
28. H = TCP is in InBox 0
29. H = TCP is in InBox 1

© RAINBOW ROBOTICS Inc. All rights reserved.

30. H = Is Alarm
31. H = Robot posture is Begin posture
32. H = Emergency Teaching Enable
33. H = Prog. Run in Sub.P area

Vorsicht

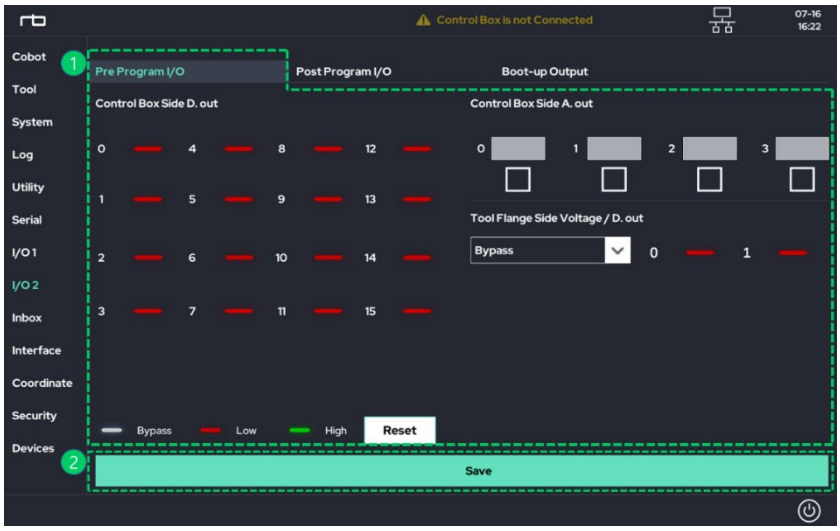
Vorsicht:

1) Bevor Sie den Digitalausgang verwenden, machen Sie sich bitte mit den vom Hersteller bereitgestellten elektrischen Eigenschaften des Digitalausgangsports vertraut.

© RAINBOW ROBOTICS Inc. All rights reserved.

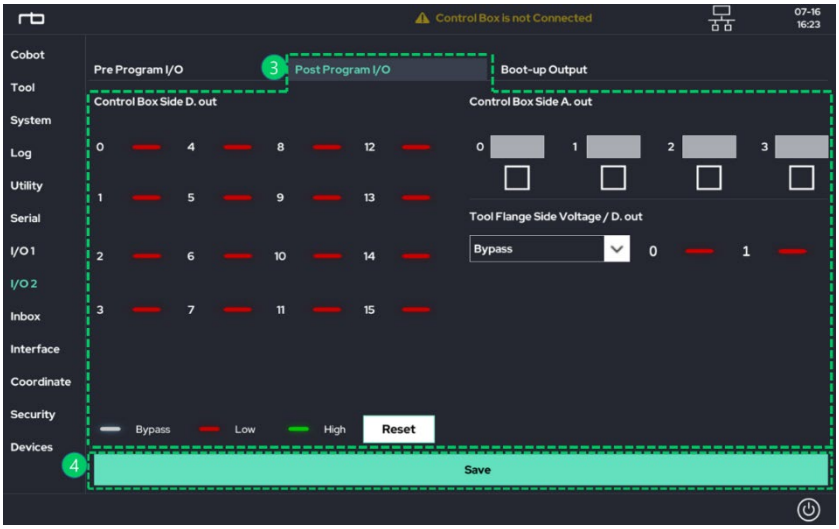
9.8 Set-up (Ein/Aus 2)

Der Ein/Aus-Wert wird so eingestellt, dass er immer vor/nach dem Programmbetrieb ausgeführt wird.

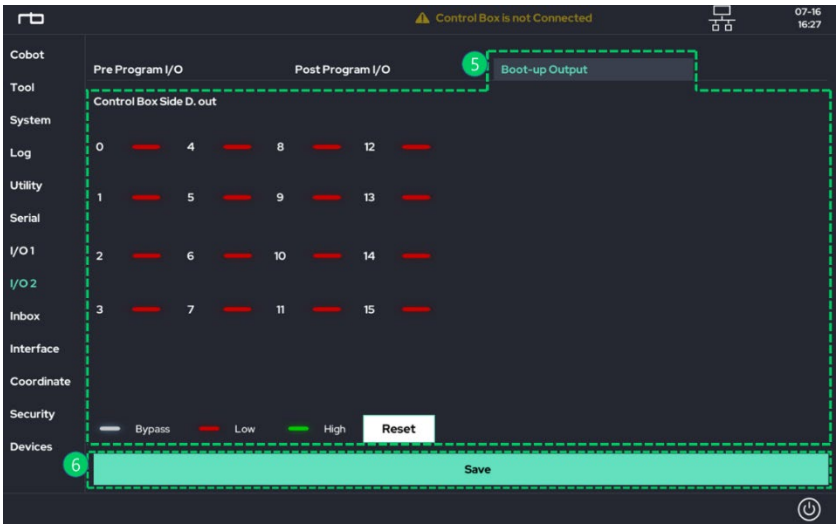


①	Ein/Aus-Übertragung wird so eingestellt, dass sie vor dem Programmstart (Wiedergabe) durchgeführt wird. Durch hier eingestellte Anschlüsse werden die entsprechenden eingestellten Werte ausgegeben, sobald das Programm startet.
②	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.



③	Ein/Aus-Übertragung wird eingestellt, die nach dem Programmende durchgeführt werden soll. Durch hier eingestellte Anschlüsse werden, die entsprechenden eingestellten Werte ausgegeben, sobald das Programm endet.
④	Die geänderten Einstellungen werden gespeichert.

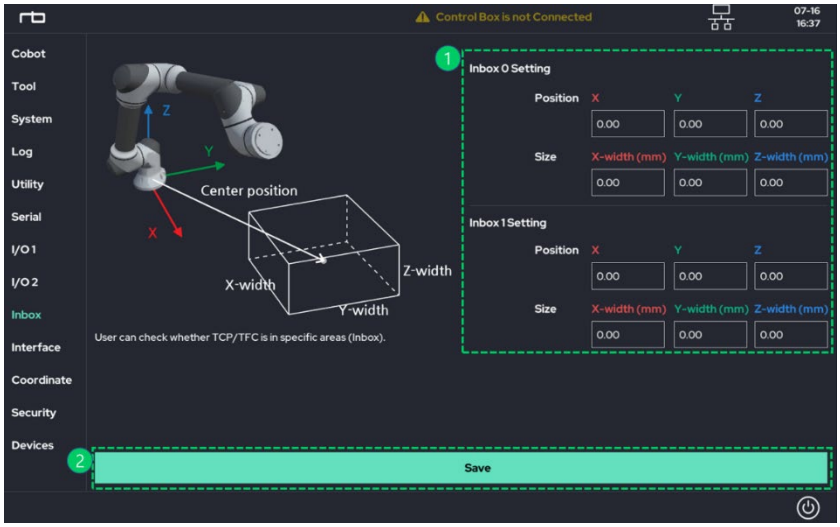


⑤	Nachdem die Steuerbox erst hochgefahren ist, wählen Sie die Digitalausgangsoption für die Steuerbox.
⑥	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.9 Set-up (Inbox)

Um die Inbox-Funktion zu verwenden, können die Größe und die Standortinformationen von Inbox eingestellt werden.

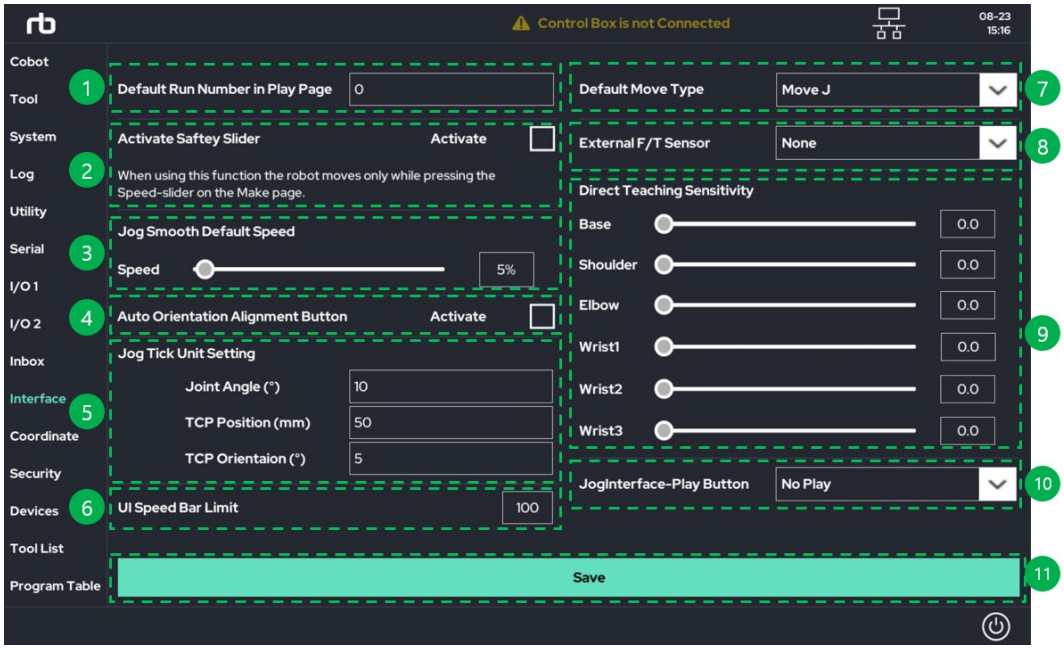


①	Die Position und Größe des Mittelpunkts von Inbox 0 und 1 werden eingegeben. Das Bezugskoordinatensystem der Einstellung ist das Basiskoordinatensystem des Herstellers.
②	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.10 Set-up (Schnittstelle)

Die erforderlichen Einstellungen werden vorgenommen, damit der Benutzer den Roboter und die Benutzeroberfläche bedienen kann.

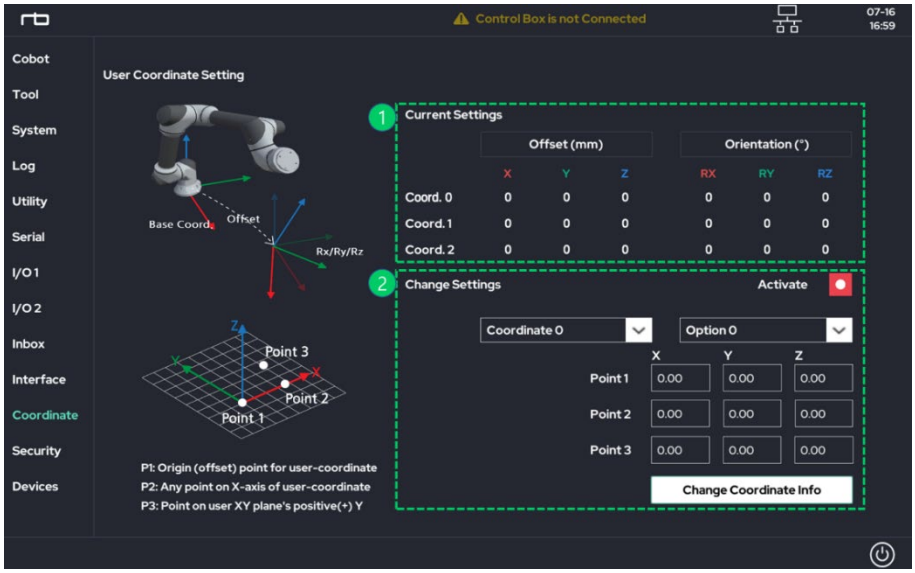


①	Die Anzahl der Wiederholungen werden auf dem ‘Play’-Bildschirm eingegeben.
②	Auf dem ‘Make’-Bildschirm wird ausgewählt, ob Sie die Sicherheitsschieber-Funktion aktivieren möchten Bei Verwendung des Sicherheitsschiebers kehrt die Geschwindigkeitsleiste auf 0 zurück, wenn der Benutzer den Bildschirm nicht gedrückt hält. Bei Deaktivierung wird die Geschwindigkeitsleiste auf dem von Benutzer festgelegten Niveaus gehalten, auch wenn der Benutzer den Bildschirm nicht gedrückt hält.
③	Geschwindigkeit für die Option ‘Smooth‘ wird eingestellt, während ‚Jog‘ verwendet wird.
④	Am unteren Rand des Jogs auf dem ‘Make’-Bildschirms wird eingestellt, ob die Funktionsschaltfläche ‚Auto Rotate Alignment’ angezeigt werden soll.
⑤	Bewegungsmenge pro ‘Tick’ bei Verwendung von ‚Jog‘ wird angegeben.
⑥	Der obere Grenzwert der Geschwindigkeitsleiste auf der Benutzeroberfläche wird begrenzt.
⑦	Die Betriebseigenschaft wird ausgewählt, die beim Erstellen von ‘Move‘ auf dem ‘Make’-Bildschirm standardmäßig erstellt wird.
⑧	Ob der externe F/T-Sensor verwendet wird, wird ausgewählt.
⑨	Ob die Funktionsschaltfläche für ‘Auto Rotate Alignment’ am unteren Rand des ‘Make’-Bildschirms angezeigt wird, wird ausgewählt.
⑩	Bei Verwendung des von Rainbow Robotics bereitgestellten dedizierten Jog-/Not-Aus-Schnittstellengeräts wird die Rolle der Wiedergabetaste auf dem Gerät definiert.
⑪	Die geänderten Einstellungen werden gespeichert.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.11 Set-up (Coordinate)

Benutzerdefiniertes Koordinatensystem (Benutzerkoordinate) wird eingerichtet.

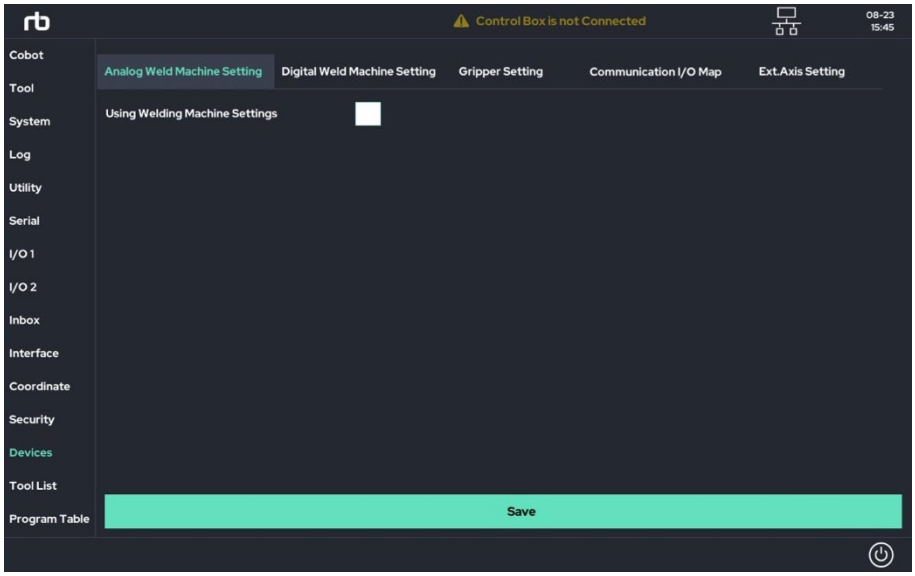


①	Dies sind die aktuell eingestellten Informationen des benutzerdefinierten Koordinatensystems. Die angezeigten Informationen basieren auf dem Roboterbasis-Koordinatensystem des Herstellers.
②	Die Einstellung eines benutzerdefinierten Koordinatensystems wird geändert. Über 3-Punkt-Einstellungen wird das Koordinatensystem eingestellt.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.12 Set-up (Devices)

Zusätzlich mit dem Roboter verbundene Geräte (Devices) werden eingerichtet

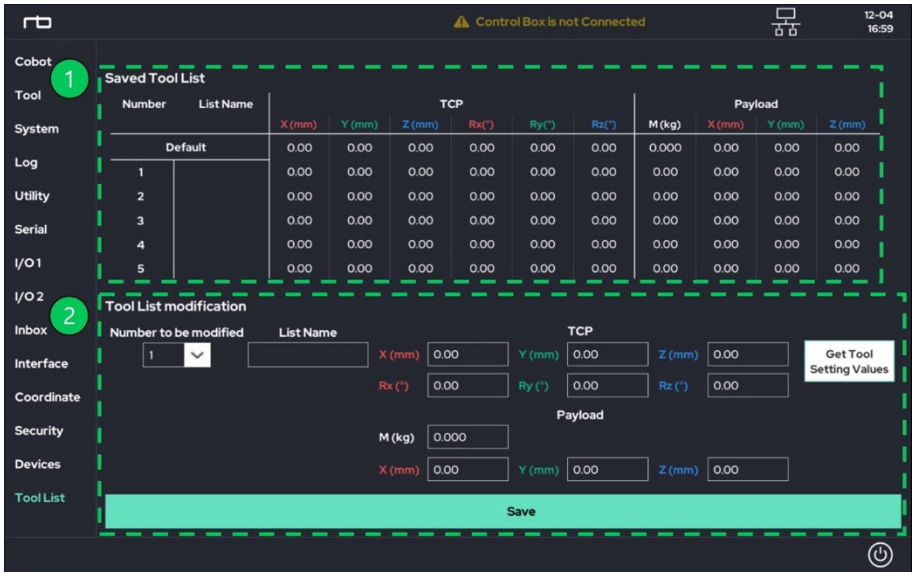


Einstellungen von an dem Roboter angeschossenen Geräte werden fortgefahren.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.13 Set-up (Tool List)

TCP-Werkzeugliste (Tool List) wird eingerichtet.



①	Der aktuell eingestellte TCP-Wert wird angezeigt.
②	Wählen Sie die zu ändernde TCP-Nummer aus. Nachdem Sie den Namen, den TCP-Standort und den Schwerpunkt festgelegt haben, speichern Sie sie.

© RAINBOW ROBOTICS Inc. All rights reserved.

9.14 Set-up (Program Table)

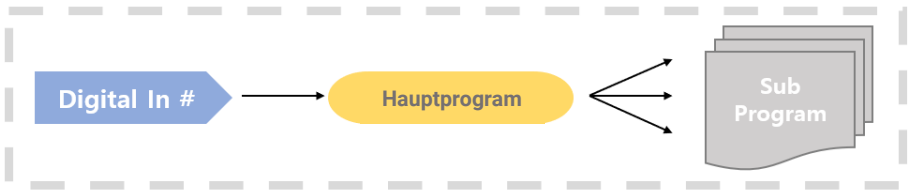
Das Programm, das über jeden digitalen Eingang geladen werden soll, wird eingestellt.

①	Zuvor erstellte Einstellungen werden importiert.
②	Welcher digitale Eingangsanschluss verwendet werden soll, wird angegeben.
③	Die Funktion, die in ‘Load’, ‘Load + Play (Once)’, ‘Load + Play (Repeat)’ unterteilt ist, wird ausgewählt.
④	Das Projekt, das über den digitalen Eingang verwendet werden soll, wird ausgewählt.
⑤	Die geänderten Einstellungen werden gespeichert.

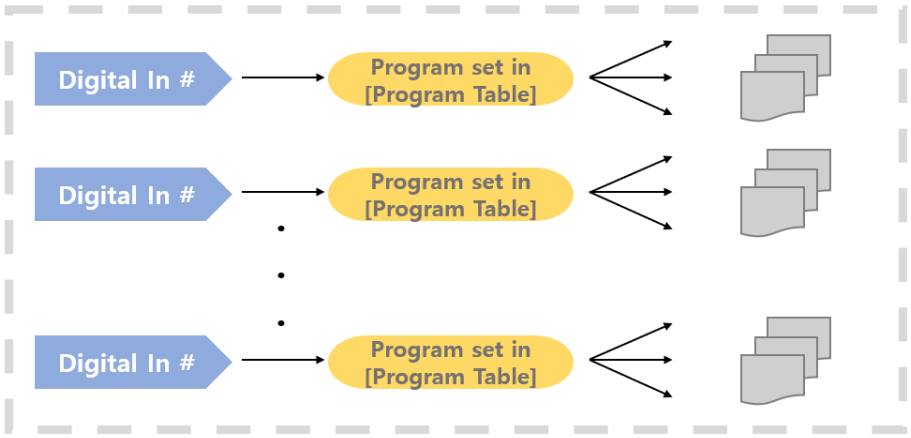
© RAINBOW ROBOTICS Inc. All rights reserved.

Die Funktion ‚Start Program', die als ‚Definition der Digitaleingangsfunktion der Steuerbox' unter 'Setup > I/O 1' verwendet wird, kann nur einen Digitaleingang verwenden, und das Programm kann nur das aktuell in die Steuerbox hochgeladene Hauptprogramm ausführen. Unter 'Setup > Program Table' können verschiedene Programme in verschiedene digitale Eingänge geladen und Programme ausgeführt werden.

Die Abbildung unten zeigt den Unterschied zwischen den beiden Funktionen.



SetUp > I/O1



SetUp > Program Table

© RAINBOW ROBOTICS Inc. All rights reserved.

Kapitel10 Wartung

10.1 Checkliste und Zyklus

Um den Roboter über einen langen Zeitraum in besten Zustand zu halten, ist eine kontinuierliche Inspektion notwendig. Die für die Inspektion verantwortliche Person erstellt und führt einen Inspektionsplan aus. Für die folgenden Punkte ist eine regelmäßige Inspektion erforderlich. Sollten Sie bei der Inspektion ein Problem feststellen, das Sie nicht selbst lösen können, wenden Sie sich bitte an den Hersteller.

Inspektions-ziel	Checkliste		Zyklus
Roboterarm	Roboter	Stellen Sie sicher, dass der Roboter sich gemäß dem darauf angewendeten Programm in die vorgesehene Position bewegt.	täglich
		Stellen Sie sicher, dass der Roboter die Position zwischen Ein- und Ausschalten beibehält.	
		Entfernen Sie Verunreinigungen wie Flecken und Staub.	Alle 3 Monate
	Motor	Überprüfen Sie, ob Sie auf ungewöhnliche Hitze oder Geräusche erzeugt.	täglich
	Schrauben	Überprüfen Sie, ob alle Schrauben am Gehäuse festgezogen sind.	Alle 3 Monate
Steuerbox	Kabel	Überprüfen Sie die Kabelverbindungen.	Alle 6 Monate
	Innerhalb der Box	Entfernen Sie den Staub im Inneren.	

Vorsicht

Vorsicht:

1) Schalten Sie bei der Wartung die an das System angeschlossene Stromversorgung (Steuerbox und Roboterarm) aus.
2) Entfernen Sie bei Modellen mit pneumatischer / elektrischer Leitungsdurchführung die angeschlossene Energiequelle (pneumatisch / elektrisch).

© RAINBOW ROBOTICS Inc. All rights reserved.

10.2 Wartung des Roboterarms

- Inspektionszyklus

  Der Roboterarm muss mindestens alle 1 Jahr einer Inspektion unterzogen werden. Je nach Verschleiß kann das Inspektionszyklus abweichen.

- Wartung und Reinigung des Roboterarm

  1. Bewegen Sie den Roboterarm in die Ausgangsposition(Home).
  2. Schalten Sie die Robotersteuerbox aus.
  3. Überprüfen Sie Folgendes:
     ① Ob das Kabel zwischen der Steuerbox und dem Roboterarm anormal ist
     ② Ob die äußerlich freiliegenden Schrauben nicht gelockert sind
     ③ Ob es Anomalien bei Motor, Bremse, Untersetzungsgetriebe usw. gibt
  4. Entfernen Sie alle Verunreinigungen wie Flecken oder Schmutz.

© RAINBOW ROBOTICS Inc. All rights reserved.

10.3 Wartung der Robotersteuerbox

Staub in der Steuerbox des Roboters kann zu elektrostatischer Aufladung oder Überhitzung führen, die möglicherweise die Steuerbox beschädigen können. Überprüfen Sie regelmäßig das Innere der Steuerbox, um Staub zu entfernen. Prüfen Sie, ob die intern angeschlossenen Kabel richtig verbunden sind.

- Inspektionszyklus

  Die Robotersteuerbox muss mindestens alle 6 Monate einer Inspektion unterzogen werden. Je nach Installationsort kann das Inspektionszyklus abweichen.

- Überprüfen und Reinigen der Robotersteuerbox

  1. Schalten Sie die Steuerbox aus.
  2. Öffnen Sie die Decke der Steuerbox.
  3. Prüfen Sie, ob sich Staub im Inneren angesammelt hat.
  4. Entfernen Sie eventuell vorhandenen Staub vorsichtig mit einem Staubsauger
  5. Prüfen Sie, ob die Anschlusskabel nicht getrennt und richtig angeschlossen sind.

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang A Technische Spezifikationen des Systems

Roboterarm

	Spezifikationen
Nutzlast	RB5-850E Serie: 5 kg / 11 lbs RB3-1200E Serie: 3 kg / 6,6 lbs RB10-1300E Serie: 10 kg / 22 lbs
Gewicht	RB5-850E Serie, RB3-1200E Serie: 22 kg / 48,5 lbs RB10-1300E Serie: 33 kg / 72,8 lbs
Reichweite	RB5-850E Serie: 850 mm / 33,5 in RB3-1200E Serie: 1.200 mm / 47,2 in RB10-1300E Serie: 1.300 mm / 51,1 in
Freiheitsgrad	6 Achsen
Reichweite des Gelenks	± 360°(Ellbogen: ± 165°)
Geschwindigkeit	Gelenke: Max. 180°/s, Werkzeug: 1m/s
Wiederholbarkeit	±0.1 mm
Montagefläche	RB5-850E/RB3-1200E Serie: Φ173 mm RB10-1300E Serie: Φ196 mm
Anschlusstyp für Werkzeug (Endeffektor)	M10 12-Pin
Werkzeugflansch-Ein/Aus	Non-E-Version: Digitaler Eingang 2, Digitaler Ausgang 2, Analogeingang 2 E-Version: Digitaler Eingang 6, Digitaler Ausgang 2
Kommunikation des Werkzeugflansches	RS485
Netzteil des Werkzeugflansches	12V/24V, 2A
IP-Rate	IP66
Temperatur / Geräusch	0 ~ 50 ˚C / <65dB
Material	Aluminium, Stahl.
Kabellänge	Netzkabel, Roboterarm – Anschlusskabel der Steuerbox, Not-Aus /Jog-Schnittstellenkabel : 5m

© RAINBOW ROBOTICS Inc. All rights reserved.

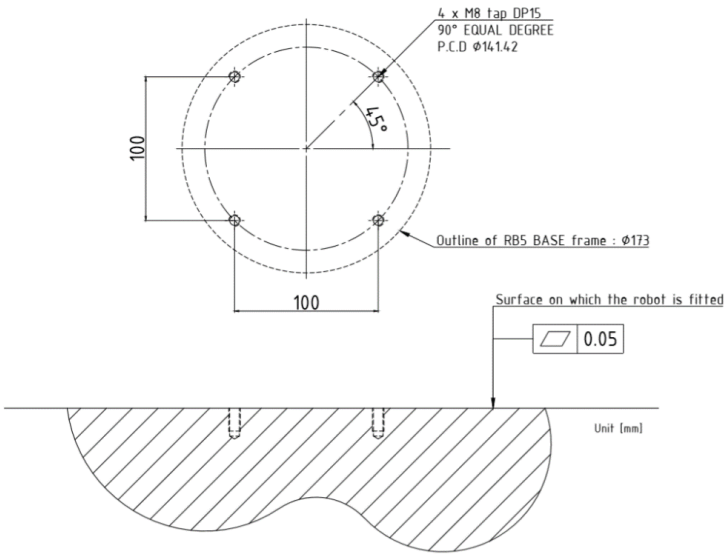
Standsteuerbox

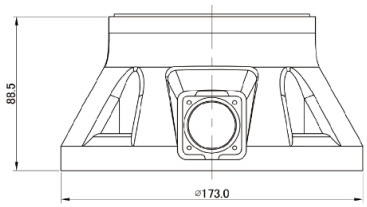
	Spezifikationen
Gewicht	CB04: 17,3 kg / 38,14 lbs
Größe(B x H x T)	454 x 240 x 416,2 mm
Ein/Aus-Anschluss	Digitaler Eingang 16, Digitaler Ausgang 16 Analogeingang 4, Analogausgang 4
Kommunikation	Ethernet, TCP/IP
Netzteil	100 ~ 240 VAC, 50 ~ 60 Hz
Material	EGI

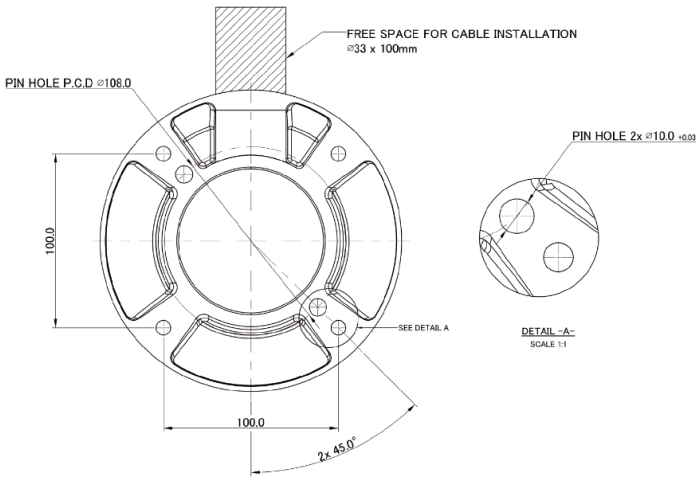
© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang B Abmessungen der Roboterbefestigungsfläche

- Abmessungen der Montagefläche des Roboters RB5-850E / RB3-1200E Serie
  - P.C.D: Teilkreisdurchmesser (Pitch Circle Diameter)
  - DP: Tiefe (Depth)

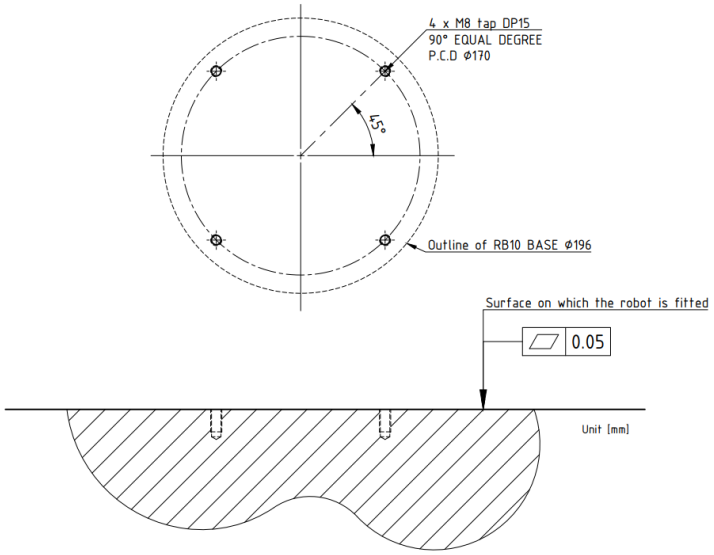


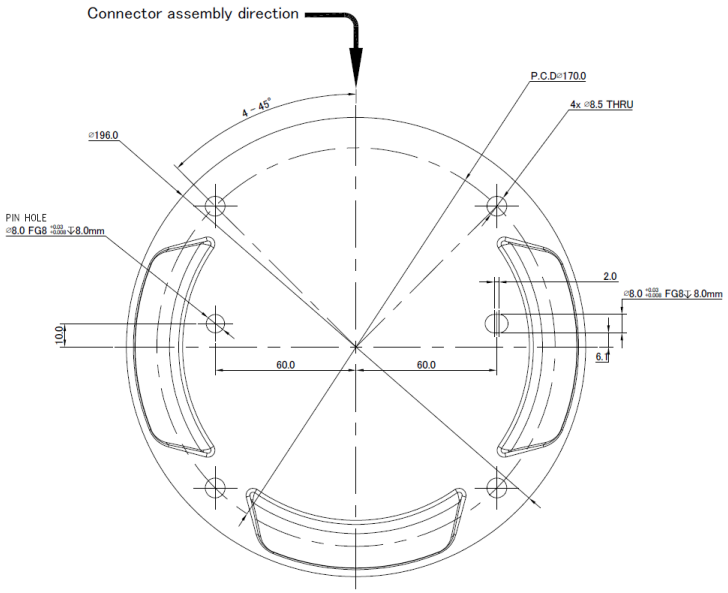




© RAINBOW ROBOTICS Inc. All rights reserved.

- Abmessungen der Montagefläche des Roboters RB10-1300E Serie
  - P.C.D: Teilkreisdurchmesser (Pitch Circle Diameter)
  - DP: Tiefe (Depth)

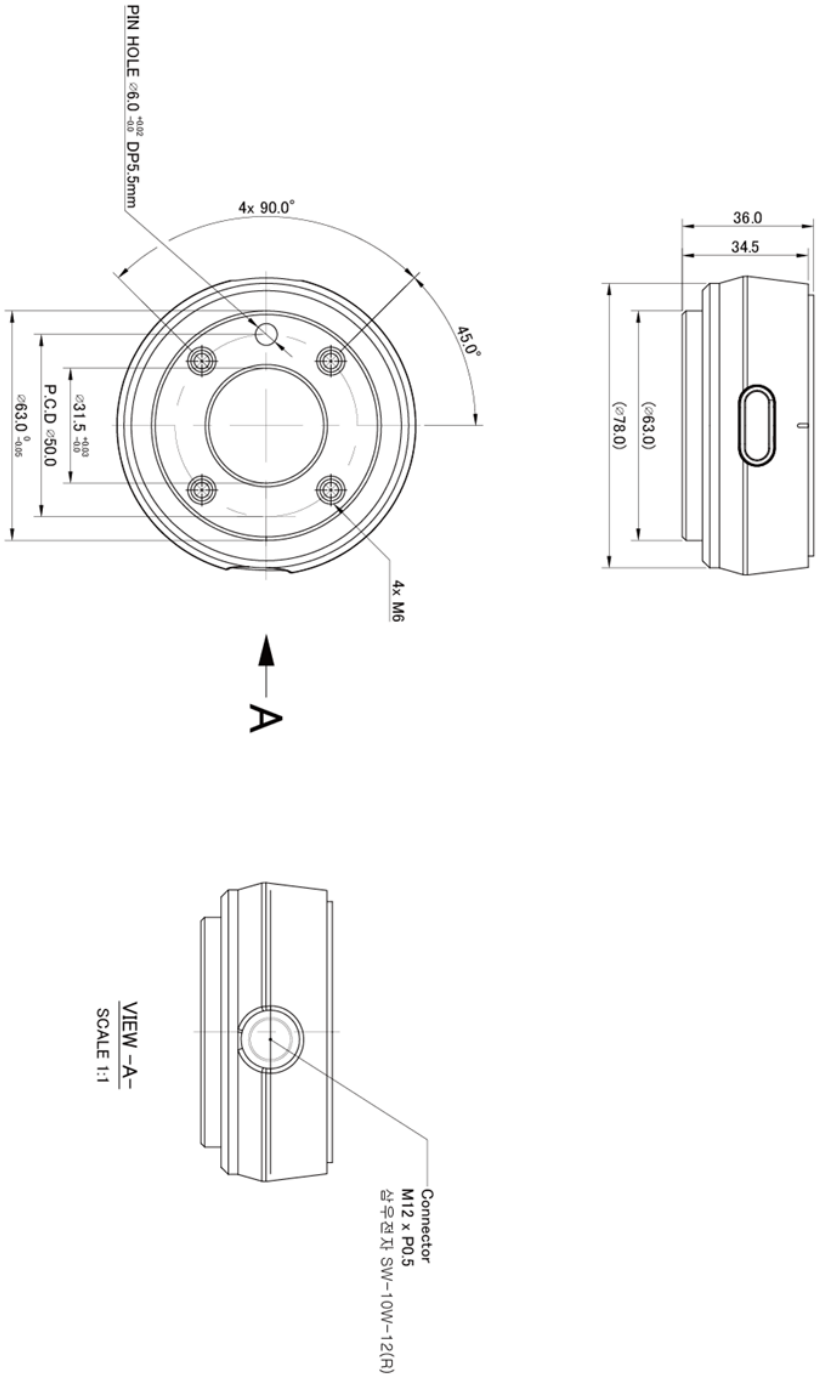




© RAINBOW ROBOTICS Inc. All rights reserved.

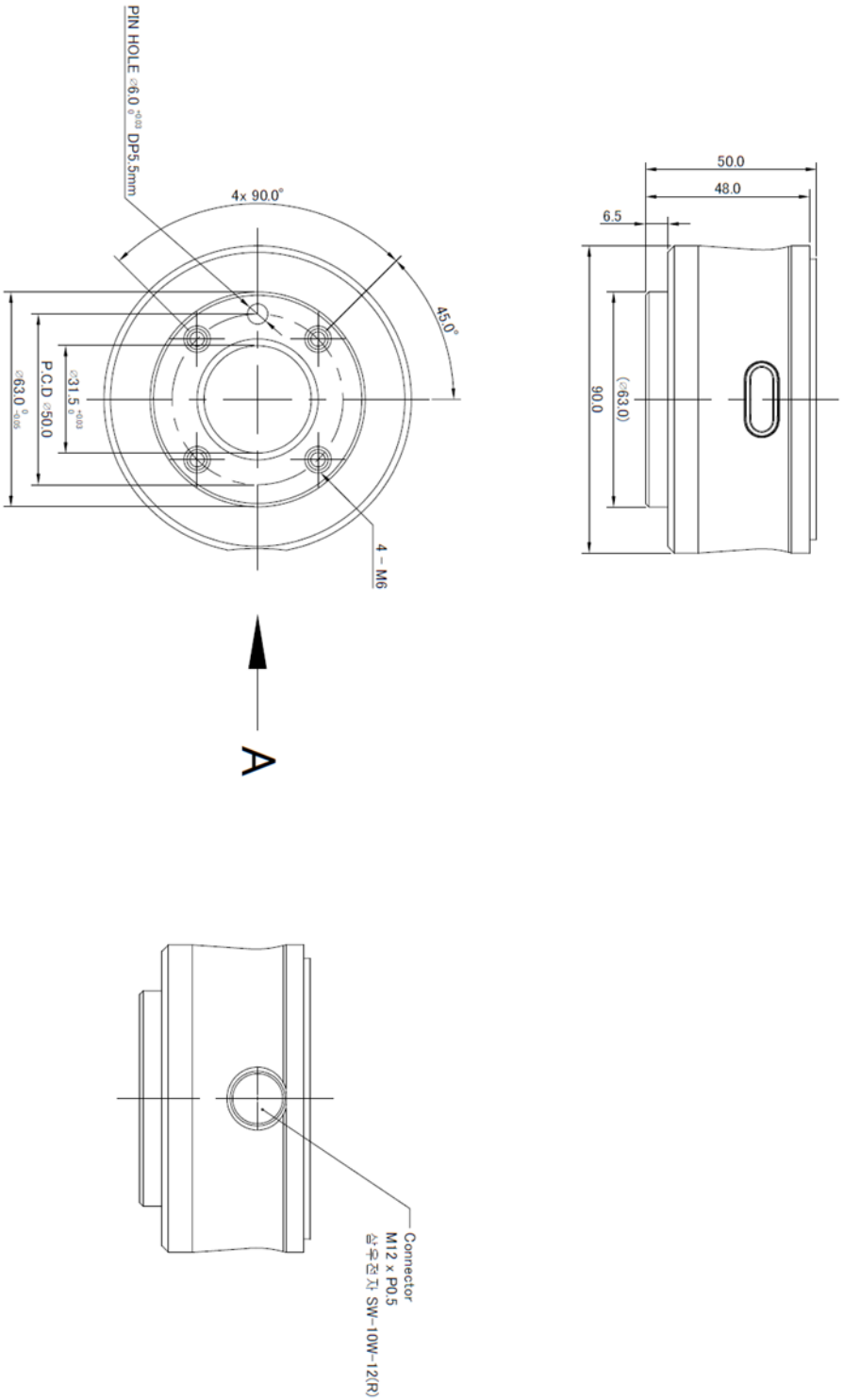
Anhang C Querschnitt von Werkzeugflansch

- Querschnitt von Werkzeugflansch RB5-850E / RB3-1200E Serie
  - P.C.D: Teilkreisdurchmesser (Pitch Circle Diameter)
  - DP: Tiefe (Depth)



© RAINBOW ROBOTICS Inc. All rights reserved.

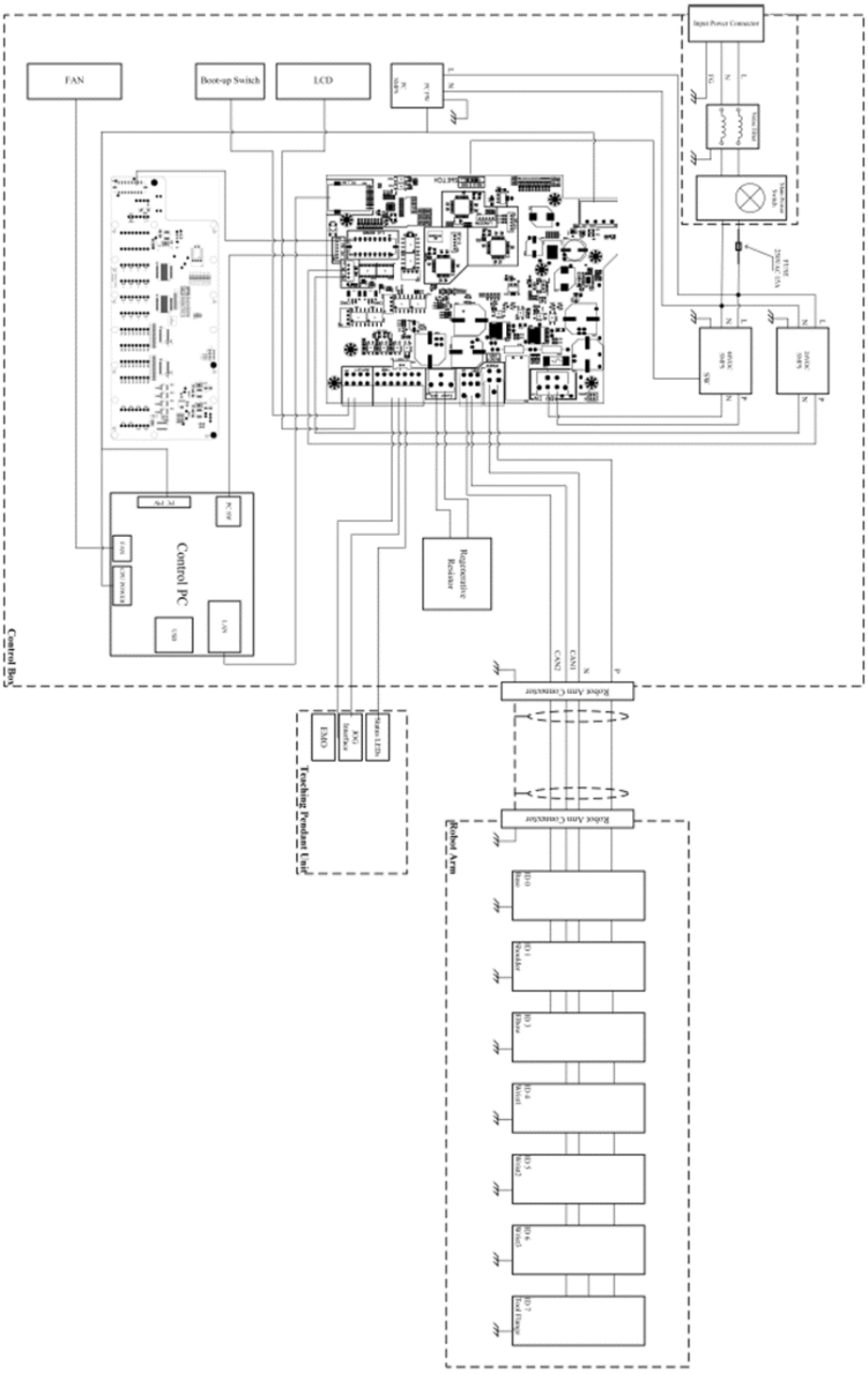
- Querschnitt von Werkzeugflansch RB10-1300E Serie
  - P.C.D: Teilkreisdurchmesser (Pitch Circle Diameter)
  - DP: Tiefe (Depth)



© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang D Schaltschema von Steuerbox

- Schaltschema von Standsteuerbox

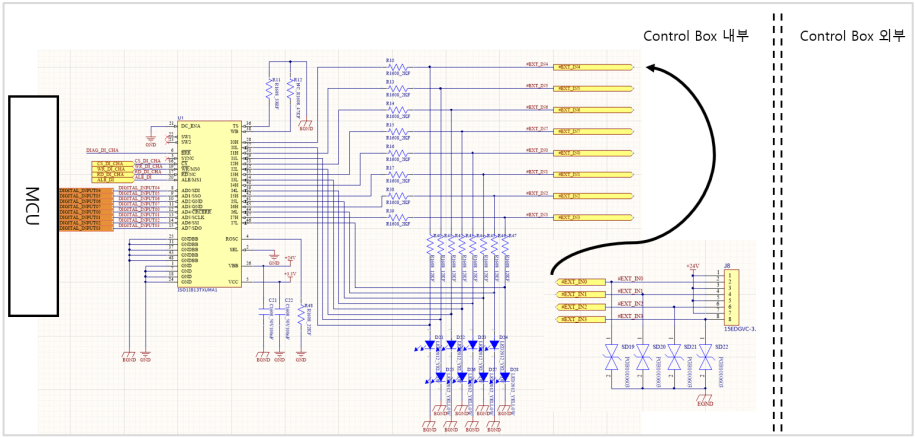


© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang D-1 Digitaler Eingang für Steuerbox

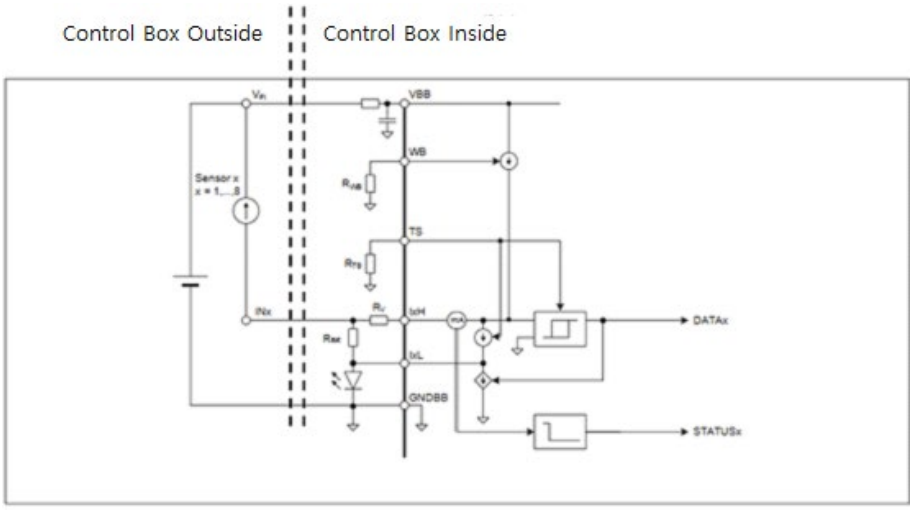
- Vorsicht
  Vor dem Verdrahten des digitalen Eingangsanschlusses der Steuerbox muss die Stromversorgung unterbrochen werden.

1. Interner Schaltplan des Digitaleingangs [DI00 ~ DI15]



Dies ist die Gerätekonfiguration, die den Digitaleingang [DI00-DI15] der Steuerbox empfängt.
Es gibt eine interne 24V-Versorgungsklemme und eine Fehlfunktion tritt auf, wenn externe 24V zugeführt werden.

2. Verwendung der Digitaleingangsgeräte [DI00 ~ DI15]



© RAINBOW ROBOTICS Inc. All rights reserved.

So verwenden Sie das digitale Eingangsgerät der RB Steuerbox [DI00-DI15].

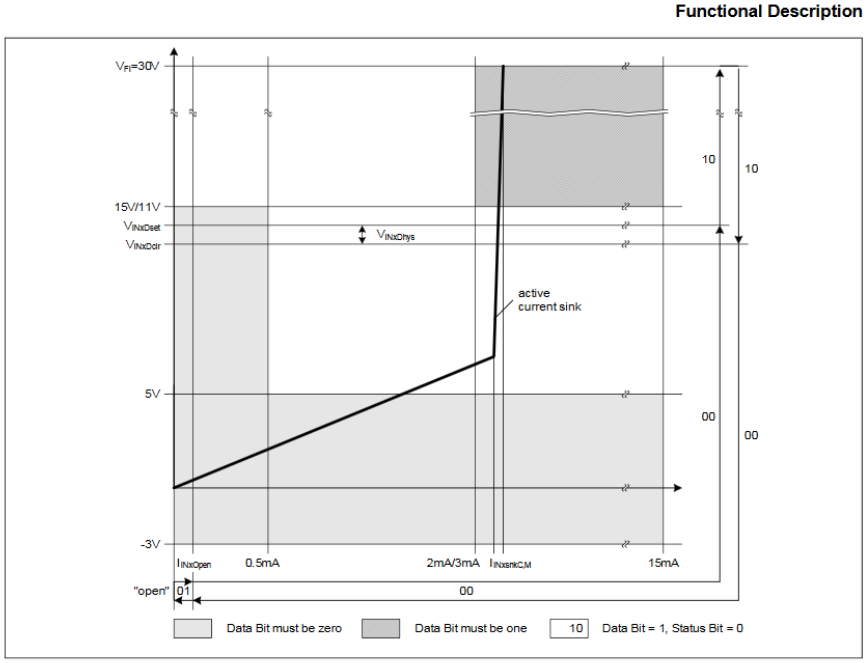


Figure 11 Sensor Input Characteristics

Spannungs-/Stromkennlinie des Digitaleingangssignals.

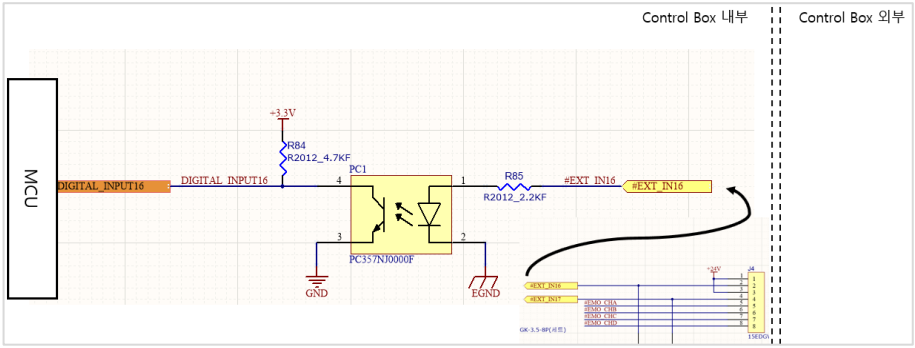
3. Eigenschaften des Digitaleingangs [DI00 ~ DI15]

Terminals	Parameter	Min	Typ	Max	Unit
[DI00 – DI15]	Voltage	-3	-	30	V
[DI00 – DI15]	OFF region	-3	-	5	V
[DI00 – DI15]	ON region	11	-	30	V
[DI00 – DI15]	Current(11-30V)	2	-	15	mA
[DI00 – DI15]	Function	-	PNP+	-	Type
[DI00 – DI15]	IEC 61131-2	-	1	-	Type

Diese Spezifikation gilt nur für Digitaleingang 0 bist Digitaleingang 15.

© RAINBOW ROBOTICS Inc. All rights reserved.

4. Interner Schaltplan des Digitaleingangs [DI16-DI17]



Dies ist die Gerätekonfiguration, die den Digitaleingang (DI16-DI17) der Steuerbox empfängt.

Es gibt eine interne 24V-Versorgungsklemme und eine Fehlfunktion tritt auf, wenn externe 24V zugeführt werden.

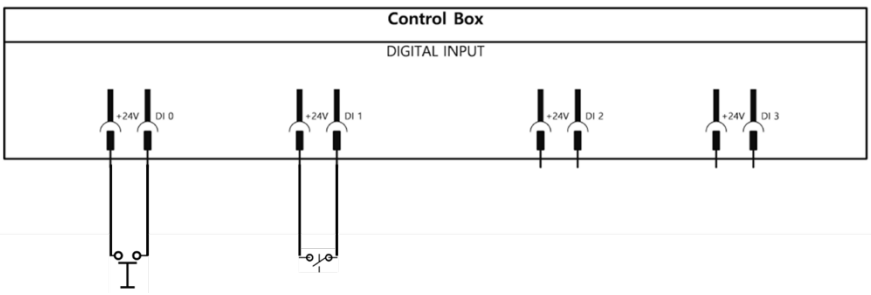
5. Eigenschaften des Digitaleingangs [DI16-DI17]

Terminals	Parameter	Min	Typ	Max	Unit
[DI16 – DI17]	Voltage	0	-	25	V
[DI16 – DI17]	OFF region	0	-	7	V
[DI16 – DI17]	ON region	7	-	25	V
[DI16 – DI17]	Function	-	PNP+	-	Type

Diese Spezifikation gilt nur für Digitaleingang 16 und 17.

6. Testumgebung

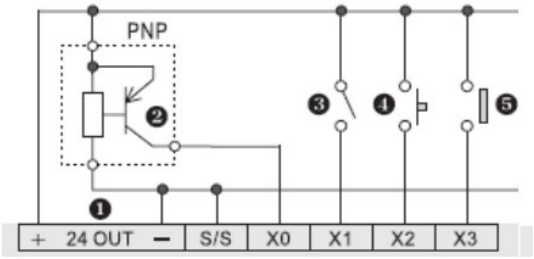
Der Test des digitalen Eingangsgeräts wurde unter Verwendung des Kippschalters durchgeführt, und mit der folgenden Konfiguration durchgeführt.



© RAINBOW ROBOTICS Inc. All rights reserved.

7. Verwendung des PNP-Sensors

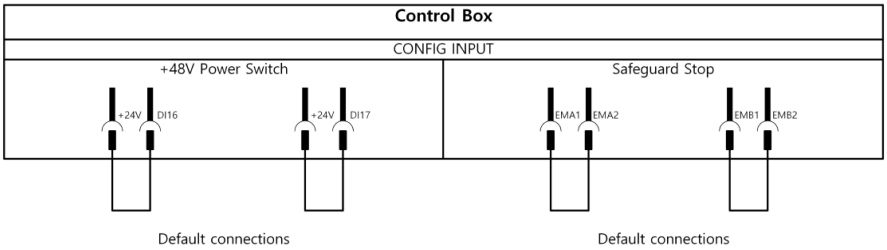
Quelle: https://blog.naver.com/mjg5080/97380010



PNP-Sensor kann auf die gleiche Weise wie oben verwendet werden.

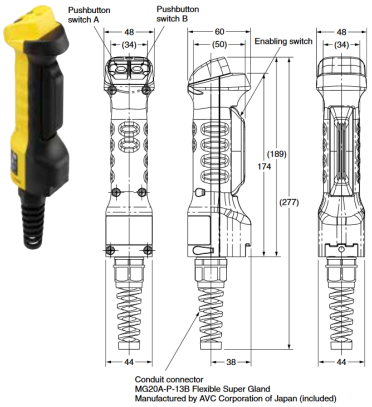
Diese Spezifikation gilt für alle Digitaleingänge

8. Verdrahtungsmethode des 3-Positionen-Zustimmgeräts

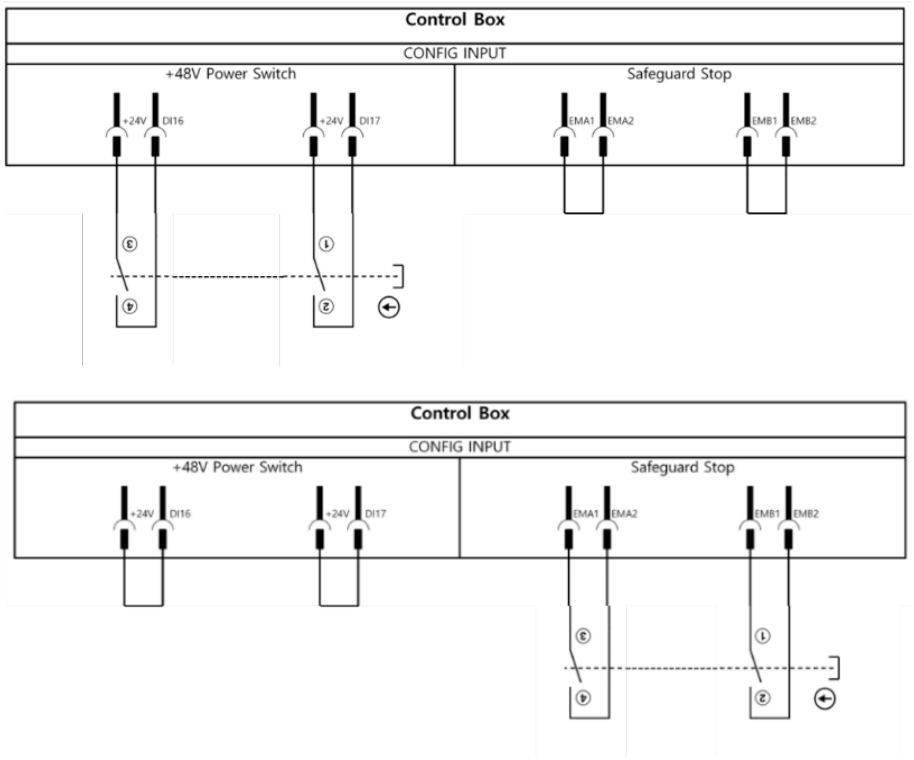


Der anfängliche Zustand ist wie oben, und es ist möglich, eine betriebserlaubende Vorrichtung gemäß den Sicherheitsmaßnahmen zu installieren.

Quelle: https://www.motionsolutions.com



© RAINBOW ROBOTICS Inc. All rights reserved.



Diese Spezifikation gilt für Zustimmeinrichtungen nach ISO 10218 und IEC 60204-1.

9. Verdrahtungsmethode der Sicherheitsgeräte

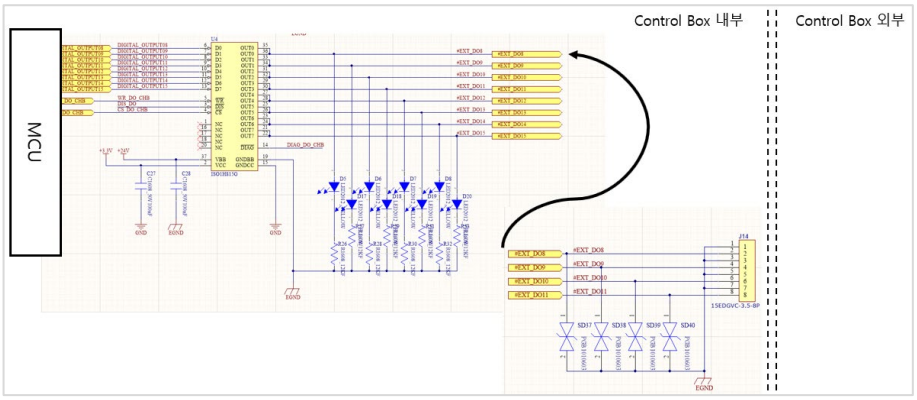
Die Verdrahtung der Sicherheitsgeräte mit PNP-Sensor und Zustimmgerät wie Lichtvorhang und Sicherheitstürsensor ist die gleiche wie oben

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang D-2 Digitaler Ausgang für Steuerbox

- Vorsicht
  Vor dem Verdrahten des digitalen Ausgangsanschlusses der Steuerbox muss die Stromversorgung unterbrochen werden.

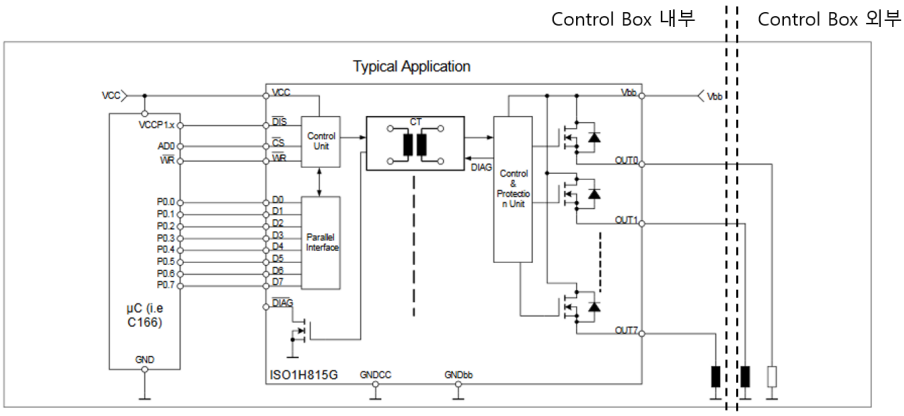
1. Interner Schaltplan des Digitaleingangs [DO00-DO15]



Dies ist die Gerätekonfiguration, die den Digitalausgang [DO00-DO15] der Steuerbox empfängt.

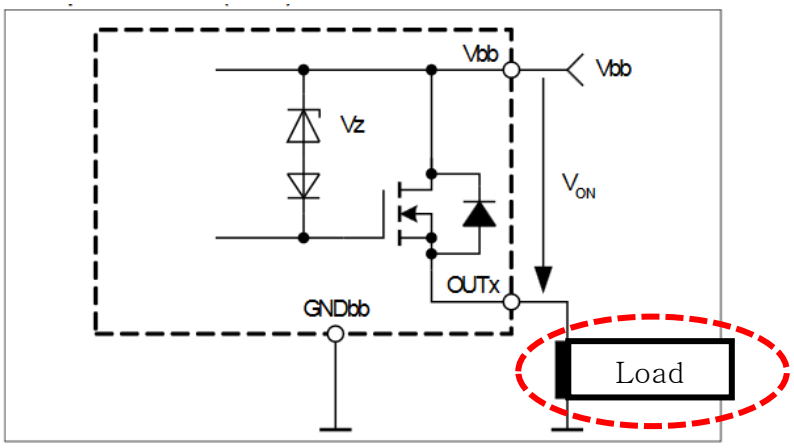
Es gibt einen internen GND-Anschluss, der mit dem GND des externen Sensors und der anzuschließenden Ausrüstung verbunden werden muss.

2. Verwendung der Digitalausgangsgeräte [DI00 ~ DI15]



So verwenden Sie das digitale Ausgangsgerät der RB Steuerbox [DI00-DI15].

© RAINBOW ROBOTICS Inc. All rights reserved.



So verwenden Sie einen einzelnen digitalen Ausgang.

Vbb-Strom wird innerhalb der Steuerbox zugeführt und als Quelle ausgegeben.

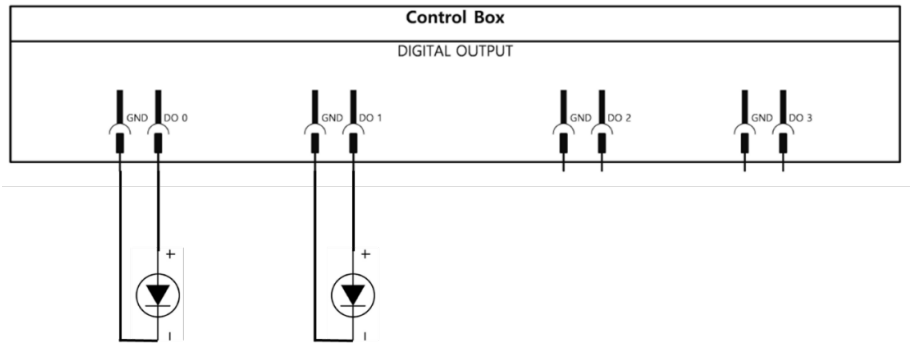
3. Eigenschaften des Digitalausgangs [DO00-DO15]

Terminals	Parameter	Min	Typ	Max	Unit
[DO00 – DO15]	Voltage	-	24	-	V
[DO00 – DO15]	CURRENT	0	-	1	A
[DO00 – DO15]	Function	-	PNP	-	Type

Ein Einzelkanal 1A kann verwendet werden, aber die Gesamtstromstärke aller Kanäle muss weniger als 2A betragen.

4. Testumgebung

Der Test des digitalen Ausgangsgeräts wurde unter Verwendung des 24V DC LEDs durchgeführt, und mit der folgenden Konfiguration durchgeführt.

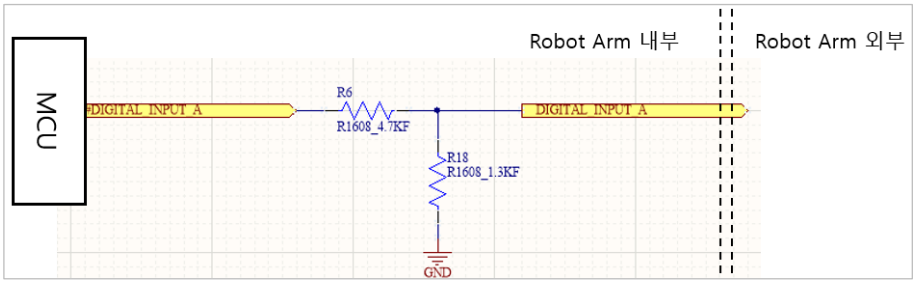


© RAINBOW ROBOTICS Inc. All rights reserved.

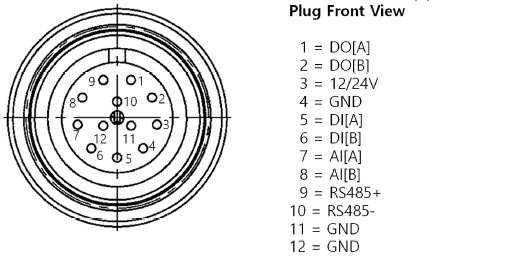
Anhang D-3 Digitaler Eingang für Werkzeugflansch

- Vorsicht
  Vor dem Verdrahten des Ein/Aus-Eingangsanschlusses von Werkzeugflansch der RB Serie muss die Stromversorgung unterbrochen werden.
- Das nachstehende elektrische Schaltbild gilt nur für den Nicht-E-Typ.

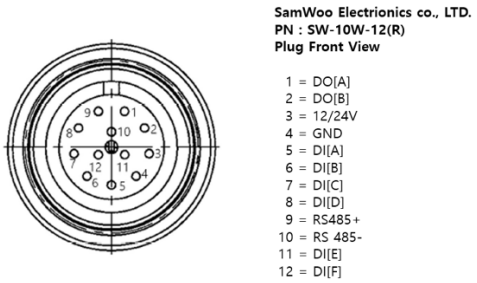
1. Interner Schaltplan des Digitaleingangs



Dies ist die Gerätekonfiguration für den Digitaleingang von Werkzeugflansch.



(1) Nicht-E-Version-Roboter



(2) E-Version-Roboter

Dies ist ein extern freiliegender Anschlussschaltplan. Je nach Roboterversion ist der Schaltplan wie oben dargestellt aufgeteilt.

© RAINBOW ROBOTICS Inc. All rights reserved.

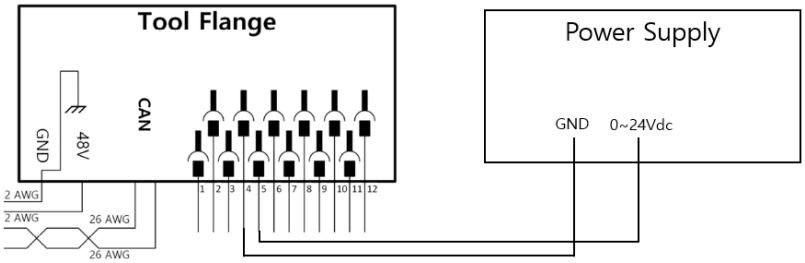
2. Eigenschaften des Digitaleingangs

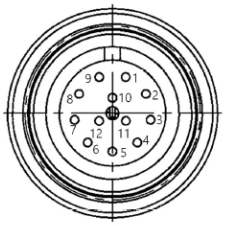
Terminals	Parameter	Min	Typ	Max	Unit
[DIA, ... , DIF]	Voltage	0	-	24	V
[DIA, ... , DIF]	OFF region	0	-	9	V
[DIA, ... , DIF]	ON region	10	-	24	V

Diese Spezifikation gilt nur für den Digitaleingang von Werkzeugflansch (Für Nicht-E-Version-Roboter gelten nur DIA und DIB.)

3. Testumgebung

Der Test des digitalen Eingangsgeräts wurde unter Verwendung einer Stromversorgung durchgeführt, und mit der folgenden Konfiguration durchgeführt.

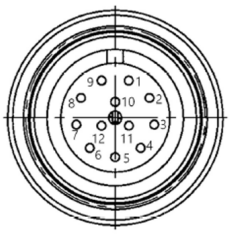




SamWoo Electronics co., LTD.
PN : SW-10W-12(R)
Plug Front View

1 = DO[A]
2 = DO[B]
3 = 12/24V
4 = GND
5 = DI[A]
6 = DI[B]
7 = AI[A]
8 = AI[B]
9 = RS485+
10 = RS485-
11 = GND
12 = GND

Nicht-E-Version



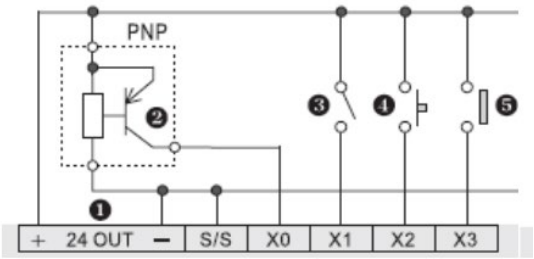
SamWoo Electronics co., LTD.
PN : SW-10W-12(R)
Plug Front View

1 = DO[A]
2 = DO[B]
3 = 12/24V
4 = GND
5 = DI[A]
6 = DI[B]
7 = DI[C]
8 = DI[D]
9 = RS485+
10 = RS 485-
11 = DI[E]
12 = DI[F]

E-Version

© RAINBOW ROBOTICS Inc. All rights reserved.

4. Verwendung des PNP-Sensors
Quelle: https://blog.naver.com/mjg5080/97380010



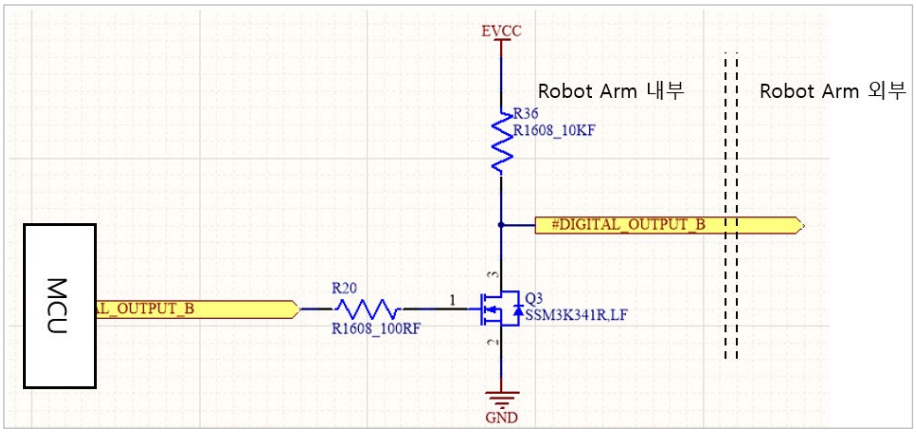
PNP-Sensor kann auf die gleiche Weise wie oben verwendet werden.
Diese Spezifikation gilt gleichermaßen für den Digitaleingang der Steuerbox.

© RAINBOW ROBOTICS Inc. All rights reserved.

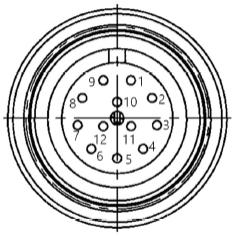
Anhang D-4 Digitaler Ausgang für Werkzeugflansch

- Vorsicht
  Vor dem Verdrahten des Ein/Aus-Ausgangsanschlusses von Werkzeugflansch muss die Stromversorgung unterbrochen werden.
- Das nachstehende elektrische Schaltbild gilt nur für den Nicht-E-Typ.

1. Interner Schaltplan des Digitalausgangs



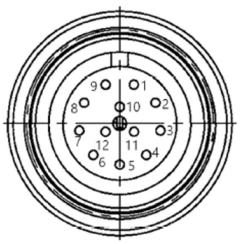
Dies ist die Gerätekonfiguration für den Digitalausgang von Werkzeugflansch [DOA, DOB].



SamWoo Electronics co., LTD.
PN : SW-10W-12(R)
Plug Front View

1 = DO[A]
2 = DO[B]
3 = 12/24V
4 = GND
5 = DI[A]
6 = DI[B]
7 = AI[A]
8 = AI[B]
9 = RS485+
10 = RS485-
11 = GND
12 = GND

(1) Nicht-E-Version-Roboter



SamWoo Electronics co., LTD.
PN : SW-10W-12(R)
Plug Front View

1 = DO[A]
2 = DO[B]
3 = 12/24V
4 = GND
5 = DI[A]
6 = DI[B]
7 = DI[C]
8 = DI[D]
9 = RS485+
10 = RS 485-
11 = DI[E]
12 = DI[F]

(2) E-Version-Roboter

Dies ist ein extern freiliegender Anschlussschaltplan. Je nach Roboterversion ist der Schaltplan wie oben dargestellt aufgeteilt.

© RAINBOW ROBOTICS Inc. All rights reserved.

2. Eigenschaften des Digitalausgangs

Terminals	Parameter	Min	Typ	Max	Unit
[DOA, DOB]	Voltage	0	12/24	24	V
[DOA, DOB]	CURRENT Ver 1.	0	150	700*	mA
[DOA, DOB]	CURRENT Ver 2.	0	2000	2000	mA

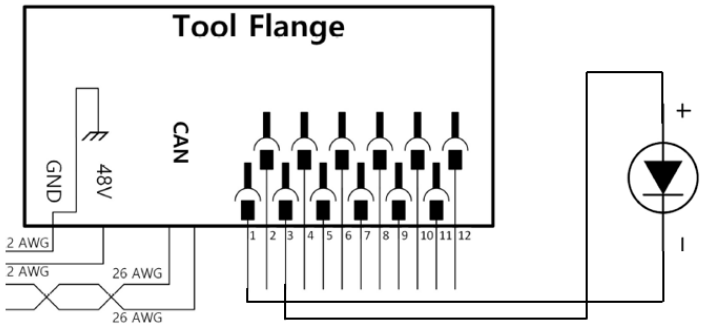
*Tsp=25°C; pulsed; tp≤10μs

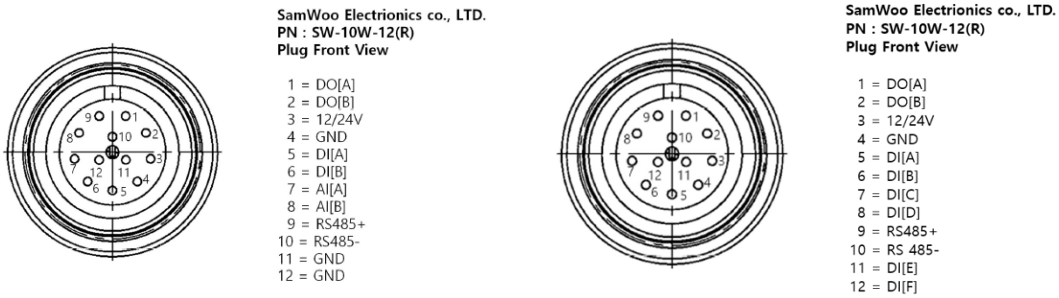
Diese Spezifikation gilt nur für den Digitalausgang A und B von Werkzeugflansch.

Der ab 24.07.2019 ausgelieferte kollaborative Roboter ist Current Ver 1.

3. Testumgebung

Der Test des digitalen Ausgangsgeräts wurde unter Verwendung des 24V DC LEDs durchgeführt, und mit der folgenden Konfiguration durchgeführt.



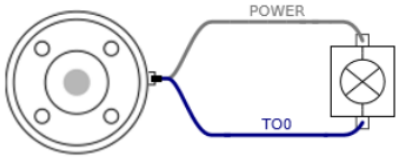


(1) Non-E Version (2) E Version

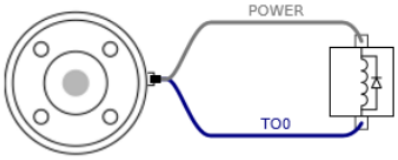
Die folgenden Beispiele sind in unserer Bedienungsanleitung definiert.

© RAINBOW ROBOTICS Inc. All rights reserved.

■ Um den digitalen Ausgang des Werkzeugs zu verwenden, wird eine 12V- oder 24V-Stromversorgung zum Einschalten der Last wie unten gezeigt verwendet. Sie können die Ausgangsspannung im Tool Out-Block definieren.



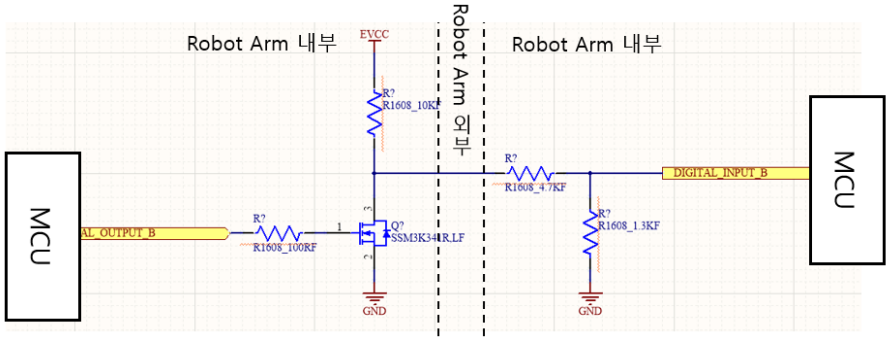
※ Es wird dringend empfohlen, eine Schutzdiode für induktive Lasten zu verwenden, wie im Diagramm unten gezeigt.



4. Vorsichtsmaßnahmen während des Gebrauchs

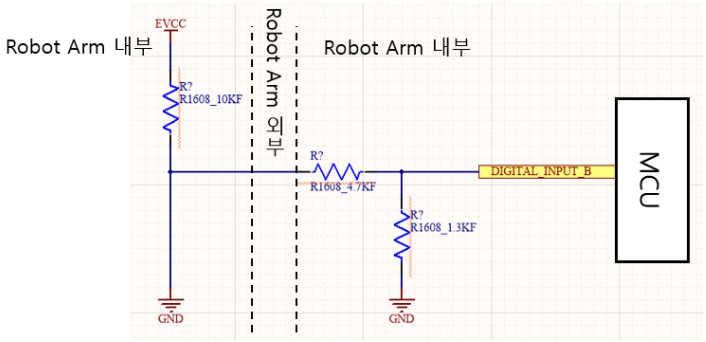
Das digitale Ausgangsgerät gehört zum NPN-Typ, hat aber einen internen 10K-Pullup-Widerstand.

Die meisten Geräte (LED, Magnetventile, Relais) können in der 3. Testumgebung oder mit der Digitalsignal-Anlegefunktion an handelsüblichen Greifern. Sie funktionieren jedoch möglicherweise nicht in Umgebungen mit derselben Spannungsverteilung wie die digitalen Werkzeugflansch-Eingangsgeräte von Rainbow Robotics.



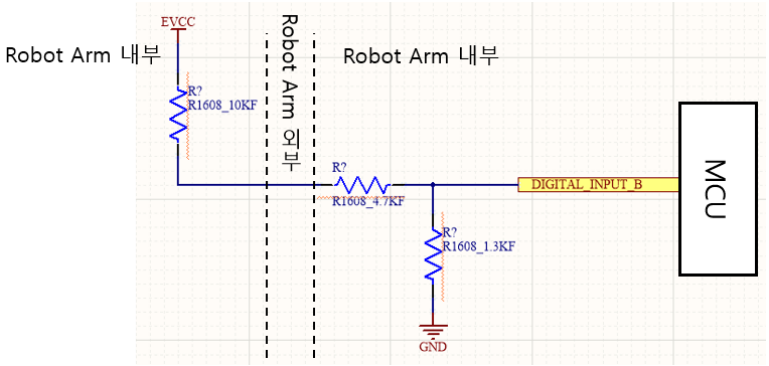
Wenn unser Digitalausgang mit dem Digitaleingang verbunden ist

© RAINBOW ROBOTICS Inc. All rights reserved.



Low Digitalausgang

Der Digitaleingang wird bei 0V Eingang ‘Low‘ ausgegeben.



High Digitalausgang

Je nach Widerstand wird der Digitaleingang möglicherweise nicht erkannt.

Wenn die an die MCU angelegte Spannung im obigen Fall EVCC 24V DC ist, werden etwa 2V angelegt, was als ‚Low‘ erkannt wird.

Wenn Benutzer wie im obigen Diagramm betreiben MÜSSEN, ist eine Anpassung des Widerstandsverhältnisses der digitalen Eingangsstufe erforderlich.

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang E Externe Skriptsteuerungs-API

E.0 Begriff

Die kollaborativen Roboter der RB-Serie können für verschiedene Umgebungen und Zwecke betrieben werden. Es kann in Verbindung mit mehreren RB-Serien oder anderen Systemen verwendet werden. In Verbindung mit dem Vision-System können die Bewegungskoordinaten in Echtzeit geändert werden oder der Benutzer kann es als Teil des bestehenden Systems verwenden.

Obwohl der Roboter mit der gegebenen Tablet-UI gesteuert werden kann, wird eine Möglichkeit geboten, den Roboter von einer beliebigen externen Steuerung aus zu steuern, um den Benutzerkomfort zu erhöhen oder den Betrieb zu optimieren.

Die RB-Serie empfängt grundsätzlich Script-Befehle und führt sie aus. Wenn es sich um eine allgemeine Betriebsmethode handelt, werden Aufgabenbewegungen über Tablet-UI geschrieben und die Skripte in der Datei nacheinander ausgeführt. Die in diesem Dokument erläuternde Methode ist, dass ein Befehlsskript von einem anderen externen Gerät empfangen, interpretiert und verschoben wird.

Steueranweisungen wie IF-, REPEAT-Anweisung, die in der Tablet-UI bereitgestellt werden, werden vom Benutzer direkt auf einem externen Steuergerät implementiert, und die Betriebsbefehle des Roboters und Ein/Aus-Steuerbefehle werden entsprechend der Situation gesendet, um den Roboter zu betreiben.

In diesem Dokument wird ein Beispiel für das Antrieben eines Roboters mit dem obigen Konzept beschrieben.

© RAINBOW ROBOTICS Inc. All rights reserved.

E.1 Externe Skriptsteuerungs-API

Die Beschreibung des in diesem Dokument bereitgestellten Skripts ähnelt dem Skript im ".wsl"-Arbeitsdokument, das unter Verwendung eins Tablets als dediziertes Skript für die externe Steuerung erstellt wurde. Aber die beiden unterscheiden sich. Da das Arbeitsdokument Anweisungen enthält, die den Ablauf steuern, wie "repeat", "if-else" und "break", steht der Abschluss einer Anweisung nicht in direktem Zusammenhang mit einer Aktion, sondern der obere Satz der Anweisung muss auch vollbeendet sein.

Wenn beispielsweise Point im Move-Befehl vorhanden sind, kann dies wie folgt ausgedrückt werden.

```
1)
move joint {
   point ( ) absolute 0.4, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
      point ( ) absolute 0.4, 0.1, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0
}
```

```
2)
move joint {
   point ( ) absolute 0.4, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
      point ( ) absolute 0.4, 0.1, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0
```

Der Unterschied zwischen 1) und 2) ist das Vorhandensein oder Fehlen von "}" am Ende. In beiden Fällen ist der point-Satz vollständig. Im Gegensatz zu 1) ist 2) jedoch eine Syntax, die nicht ausgeführt werden kann, weil die move-Anweisung, die die übergeordnete Anweisung von point-Anweisung ist, nicht abgeschlossen ist. Dann wartet der Parser auf die Beendigung der Anweisung.

```
3)
folder( ) {
      move joint {
      point ( ) absolute 0.4, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
            point ( ) absolute 0.4, 0.1, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0
      }
```

Nach der Gleichen Logik funktioniert der Parser selbst im Fall 3) nicht, während er darauf wartet, dass die Anweisung abgeschlossen wird.

Wie oben ist das Verfahren, das das Vervollständigen mehrerer Anweisungszeilen erfordert, nicht für externe Steuerung geeignet. Wenn ein Benutzer einen Befehl durch externe Steuerung sendet, analysiert er den Befehl und erwartet, dass er ausgeführt wird. Benutzer möchten nicht mehrere Befehle senden und diese Befehle mit mehreren Satzzeilen wie 3) vervollständigen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Die externe Steuerung ist also prägnant aufgebaut, sodass alle Befehle in einen Satz eingepackt werden können. Stattdessen bietet es keine Syntax oder andere unnötige Funktionen zur Steuerung des Flusses. Es ist strukturell und logisch sinnvoller und prägnanter, die Anweisungen wie “repeat”, “if-else”, “break” oder “wait” im Arbeitsdokument durch die externe Steuerungssoftware zu lösen.

Das Skript, das zuerst erklärt wird, ist Aktionsbefehle, die den Roboter tatsächlich bewegen.

Es gibt fünf Hauptbetriebsbefehle.

1) jointall
2) movetcp
3) movecircle
4) blend_jnt
5) blend_tcp

1) jointall

Befehl	jointall
Skriptcode	jointall spd, acc, joint1, joint2, joint3, joint4, joint5, joint6
Beschreibung	Dieser Befehl bewegt jedes Gelenk des Roboters im ’Joint Space’. joint1 bis joint6 sind die absoluten Winkelwert der Zielgelenke der base(Basis-), shoulder(Schulter-), elbow(Ellbogen-), wrist1(Hand-1), wrist2(Hand-2), wrist3(Hand-3)-Gelenke in dieser Reihenfolge. Geben Sie eine reelle Konstante in Grad ein. spd und acc repräsentieren Geschwindigkeits- bzw. Beschleunigungswerte. Geben Sie eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn Sie -1 eingeben, bewegt es sich mit den im System eingestellten Standardgeschwindigkeits- und Beschleunigungswerten. Falls zuvor ein anderer Aktionsbefehl ausgeführt wurde, wird dieser Befehl ignoriert.
Beispiel	“jointall 0.4, 0.1, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0”

© RAINBOW ROBOTICS Inc. All rights reserved.

2) movetcp

Befehl	movetcp
Skriptcode	movetcp spd, acc, x, y, z, rx, ry, rz
Beschreibung	Dieser Befehl bewegt den Roboter im kartesischen Raum und zielt auf die eingegebene TCP-Position und den Rotationswinkel ab. x, y, z sind reelle Konstante in mm als Wert, der die Position der TCP angibt. rx, ry und rz sind die in Gradeinheiten von Euler-Winkeln in der ‚Yaw-Pitch-Roll'-Reihenfolge ausgedrückten Werte zur Bestimmung der TCP-Richtung. spd und acc repräsentieren Geschwindigkeits- bzw. Beschleunigungswerte. Geben Sie eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn Sie -1 eingeben, bewegt es sich mit den im System eingestellten Standardgeschwindigkeits- und Beschleunigungswerte. Wenn zuvor ein anderer Aktionsbefehl ausgeführt wird, wird dieser Befehl ignoriert.
Beispiel	“movetcp 0.4, 0.1, 100.0, 100.0, 300.0, 0.0, 90.0, 0.0”

3) movecircle

Befehl	movecircle(3-Punkt-Modus)
Skriptcode	movecircle threepoints orientation_option spd, acc, x1, y1, z1, rx1, ry1, rz1, x2, y2, z2, rx2, ry2, rz2
Beschreibung	Dieser Befehl ist eine Aktion zum Zeichnen eines Kreises im kartesischen Raum unter Verwendung der eingegebenen Zwischen- und Endpositionen, beginnend mit der aktuellen Position der TCP. orientation_option ermöglicht, 3 Optionen auszuwählen, um die TCP-Ausrichtung beim Zeichen eines Kreises zu bestimmen. “intended” zeichnet einen Kreis unter Beibehaltung aller Richtungen (rx1, ry1, rz1 und rx2, ry2, rz2), die an den Zwischen- und Endpositionen eingegeben wurden. “constant” zeichnet einen Kreis unter Beibehaltung der aktuellen Richtung des Startpunkts TCP. “radial” ändert die Richtung des TCP relative zum Mittelpunkt des zu zeichnenden Kreises. Die Basis der Richtung ist die Richtung des aktuellen TCP. x1, y1, z1 sind reelle Konstante in mm als Wert, der die Zwischenposition der TCP angibt. rx1, ry1, rz1 stellen die Euler-Winkel in der Yaw-Pitch-Roll-Reihenfolge dar, um die TCP-Richtung an der Zwischenposition des Kreises zu bestimmten, und geben eine reelle Konstante in Gradeinheit ein.

© RAINBOW ROBOTICS Inc. All rights reserved.

	x2, y2, z2 sind reelle Konstante in mm als Wert, der die Endposition der TCP angibt. rx2, ry2, rz2 stellen die Euler-Winkel in der Yaw-Pitch-Roll-Reihenfolge dar, um die TCP-Richtung an der Endposition des Kreises zu bestimmten, und geben eine reelle Konstante in Gradeinheit ein. spd und acc repräsentieren Geschwindigkeits- bzw. Beschleunigungswerte. Geben Sie eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn Sie -1 eingeben, bewegt es sich mit den im System eingestellten Standardgeschwindigkeits- und Beschleunigungswerte. Wenn zuvor ein anderer Aktionsbefehl ausgeführt wird, wird dieser Befehl ignoriert.
Beispiel	“movecircle threepoints intended 0.4, 0.1, 100.0, 100.0, 300.0, 0.0, 90.0, 0.0, 200.0, 200.0, 200.0, 0.0, 90.0, 45.0” “movecircle threepoints constant 0.4, 0.1, 100.0, 100.0, 300.0, 0.0, 90.0, 0.0, 200.0, 200.0, 200.0, 0.0, 90.0, 45.0” “movecircle threepoints radial 0.4, 0.1, 100.0, 100.0, 300.0, 0.0, 90.0, 0.0, 200.0, 200.0, 200.0, 0.0, 90.0, 45.0”

Befehl	movecircle(Achsenmodus)
Skriptcode	movecircle axis orientation_option spd, acc, rot_angle, cx, cy, cz, ax, ay, az
Beschreibung	Dieser Befehl startet an der aktuellen Position des TCP und zeichnet einen Kreis in Höhe des Eingabewinkels um die Eingaberotationsachse. Mit der ‘orientation_option’ können Sie 3 Optionen auswählen, die die Ausrichtung des TCP beim Zeichnen eines Kreises bestimmen. “intended” ist im Folgenden gleichbedeutend mit “constant”. “constant” zeichnet einen Kreis unter Beibehaltung der aktuellen Richtung des Startpunks TCP. “radial” ist die Richtung des Mittelpunkts des zu zeichnenden Kreises, und die TCP-Richtung ändert sich relativ. Die Basis der Richtung ist die Richtung des aktuellen TCP. cx, cy, cz sind die Positionswerte des Drehzentrums in mm. ax, ay, az sind Vektoren, die die Drehachse darstellen. rot_angle ist ein Wert, der den zu drehenden Betrag in Grad dargestellt spd und acc repräsentieren Geschwindigkeits- bzw. Beschleunigungswerte. Geben Sie eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn Sie -1 eingeben, bewegt es sich mit den im System eingestellten Standardgeschwindigkeits- und Beschleunigungswerte. Wenn zuvor ein anderer Aktionsbefehl ausgeführt wird, wird dieser Befehl ignoriert.

© RAINBOW ROBOTICS Inc. All rights reserved.

Beispiel	“movecircle axis constant 0.4, 0.1, 180.0, 200.0, 200.0, 200.0, 1.0, 0.0, 0.0” “movecircle axis radial 0.4, 0.1, 180.0, 200.0, 200.0, 200.0, 1.0, 0.0, 0.0”

4) blend_jnt

Befehl	blend_jnt
Skriptcode	blend_jnt clear_pt
Beschreibung	Dieser Befehl ist zum Löschen der gespeicherten Zielwerte unter den Bewegungsbefehlen, die aufeinanderfolgende Zielgelenkwinkel im Joint Space verbinden. Verwenden Sie den Befehl als Vorbewegung, bevor Sie die neuen aufeinanderfolgenden Gelenkwinkel eingeben.
Beispiel	“blend_jnt clear_pt”

Befehl	blend_jnt
Skriptcode	blend_jnt add_pt spd, acc, joint1, joint2, joint3, joint4, joint5, joint6
Beschreibung	Dieser Befehl ist zum Eingeben des Zielgelenkwinkels unter den Bewegungsbefehlen, die aufeinanderfolgende Zielgelenkwinkel im Joint Space verbinden. joint1 bis joint6 sind die absoluten Winkelwerte der Zielgelenke der Basis-(base), Schulter-(shoulder), Ellbogeb-(elbow), Hand-1(wrist1), Hand-2(wrist2), Hand-3(wrist3)-Gelenke in dieser Reihenfolge. Geben Sie eine reelle Konstante in Grad ein. spd und acc repräsentieren Geschwindigkeits- bzw. Beschleunigungswerte. Geben Sie eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn Sie -1 eingeben, bewegt es sich mit den im System eingestellten Standardgeschwindigkeits- und Beschleunigungswerte. Die letzten Eingaben spd und acc bestimmen die Geschwindigkeit und Beschleunigung des gesamten “blend_jnt”-Befehls.
Beispiel	“blend_jnt add_pt 0.4, 0.1, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0”

Befehl	blend_jnt
Skriptcode	blend_jnt move_pt
Beschreibung	Dieser Befehl ist zum Starten einer Bewegung unter den Bewegungsbefehlen, die aufeinanderfolgende Zielgelenkwinkel im Joint Space verbinden. Bewegungen werden nacheinander mit den zuvor mit “blend_jnt add_pt” eingegebenen Gelenkwerten ausgeführt.
Beispiel	“blend_jnt move_pt”

© RAINBOW ROBOTICS Inc. All rights reserved.

5) blend_tcp

Befehl	blend_tcp
Skriptcode	blend_tcp clear_pt
Beschreibung	Dieser Befehl ist zum Löschen der gespeicherten Zielwerte unter den Bewegungsbefehlen, die aufeinanderfolgenden Zielorte im Cartesian Space verbinden. Verwenden Sie den Befehl als Vorbewegung, bevor Sie die neuen aufeinanderfolgenden Zielorte eingeben.
Beispiel	“blend_tcp clear_pt”

Befehl	blend_tcp
Skriptcode	blend_tcp add_pt spd, acc, radius, x, y, z, rx, ry, rz
Beschreibung	Dieser Befehl ist zum Eingeben der Zielorte unter den Bewegungsbefehlen, die aufeinanderfolgenden Zielorte im Cartesian Space verbinden. radius‘ ist ein Wert in mm, der die zu bewegende Distanz zur nächsten Zielposition angibt. Indem eine Kurve gezeichnet wird, wenn die Zielposition nacheinander bewegt werden. Der Maximalwert des ‚radius‘ ist die Hälfte der Länge des Liniensegments, das sich von einem Ziel zum nächsten bewegt. Wenn Sie einen größeren Wert eingeben, wird der Wert intern reduziert verwendet. x, y, z sind reelle Konstante in mm als Wert, der die Position der TCP angibt. rx, ry, rz stellen Euler-Winkel in der Yaw-Pitch-Roll-Reihenfolge dar, um die TCP-Richtung zu bestimmen, und geben eine reelle Konstante in Gradeinheit ein. spd und acc repräsentieren Geschwindigkeits- bzw. Beschleunigungswerte. Geben Sie eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn Sie -1 eingeben, bewegt es sich mit den im System eingestellten Standardgeschwindigkeits- und Beschleunigungswerte. Die letzten Eingaben spd und acc bestimmen die Geschwindigkeit und Beschleunigung des gesamten “blend_tcp”-Befehls.
Beispiel	“blend_tcp add_pt 0.4, 0.1, 30.0, 100.0, 100.0, 300.0, 0.0, 90.0, 0.0”

Befehl	blend_tcp
Skriptcode	blend_tcp move_pt
Beschreibung	Dieser Befehl ist zum Starten der Bewegung unter den Bewegungsbefehlen, die aufeinanderfolgenden Zielorte im Cartesian Space verbinden. Bewegungen werden sequenziell zu den zuvor mit “blend_tcp add_pt” eingegebenen Zielpositionen ausgeführt.
Beispiel	“blend_tcp move_pt”

© RAINBOW ROBOTICS Inc. All rights reserved.

Die folgenden Skripte sind Befehle zur Steuerung der Ausgabewerte der digitalen und analogen Anschlüsse von Schalttafeln und Werkzeugflanschen. Es gibt drei Befehle.

1) digital_out
2) analog_out
3) tool_out

1) digital_out

Skriptcode	digital_out d0, d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15
Beschreibung	Dieser Befehl ist zum Senden eines Signals über den digitalen Ausgangsanschlusses des Bedienfelds. d0 bis d15 zeigen Ein/Aus jeweiligen digitalen Anschlusses an. Ein Wert von 1 bedeutet an/ein, 0 Aus. Zusätzlich zu 0 und 1 kann -1 auch für d0 bis d15 eingegeben werden. In diesem Fall wird der vorhandene Ausgabewert dieses Anschlusses beibehalten.
Beispiel	“digital_out 1, 1, 1, 1, 0, 0, 0, 0, -1, -1, -1, -1, -1, -1, -1, -1”

2) analog_out

Skriptcode	analog_out a0, a1, a2, a3
Beschreibung	Dieser Befehl ist zum Senden eines Signals über den analogen Ausgangsanschlusses des Bedienfelds. a0 - a3 geben den Ausgangswert jeweiligen analogen Anschlusses an. Der Wert reicht von 0 bis 10 und wird in V angegeben. Sie können -1 für a0 - a3 eingeben. In diesem Fall wird der vorhandene Ausgabewert dieses Anschlusses beibehalten.
Beispiel	“analog_out 5.0, 5.0, -1, -1”

3) tool_out

Skriptcode	tool_out volt, d0, d1
Beschreibung	Dieser Befehl stellt den Spannungspegel des Werkzeugflansches ein und sendet ein Signal über den digitalen Ausgangsanschluss. volt stellt den Spannungspegel des digitalen Ausgangs des Werkzeugflansches ein. Mögliche Werte sind 0, 12, 24, die in V angegeben sind. Andere Eingabe als diese drei Werte werden ignoriert. Sie können -1 für volt eingeben. In diesem Fall wird die alte Spannungseinstellung beibehalten. d0, d1 zeigen Ein/Aus jeweiligen digitalen Anschlusses an. Ein Wert von 1 ist an, 0 ist aus.

© RAINBOW ROBOTICS Inc. All rights reserved.

	Zusätzlich zu 0 und 1 kann -1 für d0 und d1 auch eingegeben werden. In diesem Fall wird der vorhandene Ausgabewert dieses Anschlusses beibehalten.
Beispiel	“tool_out 12, 1, 0”

Die folgenden Befehle dienen der Initialisierung, Beendigung und Änderung von Betriebsmodus und -geschwindigkeit.

1) mc
2) shutdown
3) pgmode
4) sdw

1) mc

Skriptcode	mc jall init
Beschreibung	Die Roboterhardware wird initialisiert.
Beispiel	“mc jall init”

2) shutdown

Skriptcode	shutdown
Beschreibung	Die Bewegung des Roboters wird beendet und der Roboter wird abgeschaltet.
Beispiel	“shutdown”

3) pgmode

Skriptcode	pgmode mode_type
Beschreibung	Der Betriebsmodus des Roboters wird geändert. Über ‘mode_type’ wird der Betriebsmodus gewählt. Es gibt die Betriebsmodi “Real” und “Simulation”. Bei “Real”-Modus bewegt sich der Roboter, wenn Bewegungsbefehl von Benutzer angegeben wird. Bei “Simulation”-Modus wird nur die interne Referenz geändert, wenn Bewegungsbefehl angegeben wird. Der Befehl bewegt den Roboter nicht. Wenn der Roboter initialisiert wird, ist der Standardmodus “Simulation”-Modus.
Beispiel	“pgmode real” “pgmode simulation”

© RAINBOW ROBOTICS Inc. All rights reserved.

4) sdw (shared data write)

Skriptcode	sdw default_speed spd
Beschreibung	Dieser Befehl legt den Standardgeschwindigkeitsmultiplikator für alle Bewegungen des Roboters fest. Geben Sie für spd eine reelle Konstante zwischen 0 und 1 ein. Näher an 0 zeigt einen langsameren Wert an. Wenn dieser Wert 0 ist, bewegt sich er Roboter nicht. Auch die Referenz ändert sich nicht. Bitte beachten Sie, dass dieser Geschwindigkeitswert auch vom Programmierhandgerät gesteuert wird, wenn das Gerät angeschlossen ist, insbesondere wenn der “Make”-Modus aufgerufen wird. Es gibt keine getrennte Priorität zwischen der externen Steuerung und dem Programmierhandgerät. Der zuletzt eingegebene Wert ist der Endwert.
Beispiel	“sdw default_speed 0.5”

Das letzte Skript ist das Task-Skript

task

Skriptcode	task load work_file_name
Beschreibung	Dieser Befehl ist zum Laden einer vorhandenen Arbeitsdatei. Für work_file_name wird der Name der “.wsl”-Datei verwendet. In diesem Fall lassen Sie “.wsl” weg und geben Sie nur den relativen Pfad und Namen der Datei ein. Diese Datei sollte keine im Programmierhandgerät vorhandene Datei sein, sondern eine Datei, die geladen oder gespeichert wurde, indem mindestens einmal über das Programmierhandgerät eine Verbindung zur Zentrale hergestellt wurde. Deshalb können Sie diese Datei auch dann laden, wenn das Programmierhandgerät nicht angebracht ist.
Beispiel	“task load test_file”

Skriptcode	task play option
Beschreibung	Dieser Befehl ist zum Starten der geladenen Arbeitsdatei. option kann keinen Wert enthalten oder “once” kann darin eingegeben werden. Wenn option kein Wert gegeben wird, wird die Arbeitsdatei so oft wie eingestellt ausgeführt. Wenn Sie bei der option “once” eingeben, wird die Arbeitsdatei nur einmal ausgeführt.
Beispiel	“task play” “task play once”

© RAINBOW ROBOTICS Inc. All rights reserved.

Skriptcode	task repeat num
Beschreibung	Dieser Befehl gibt die Anzahl der Iterationen beim Ausführen der Arbeitsdatei an. Geben Sie in num die Anzahl der Iterationen als ganze Zahl ein. Wenn Sie -1 für num eingeben, bedeutet dies unendlich. Die durch diesen Befehl eingestellte Anzahl von Wiederholungen wir beibehalten, bis die Stromversorgung der Schaltanlage neu gestartet wird. Danach wird es durch den Wert ersetzt, der vom Programmierhandgerät festgelegt wurde.
Beispiel	“task repeat 5” “task repeat -1”

Skriptcode	task pause
Beschreibung	Dieser Befehl hält die laufende Roboterbewegung an. Es kann sowohl für den Aktionsbefehl durch externe Steuerung als auch für die Arbeitsdatei verwendet werden, die aufgrund des Befehls “task play” ausgeführt wurde. Im angehaltenen Zustand kann die Operation durch den Befehl “task resume_a” wieder aufgenommen werden. Wenn sich der Roboter in einem angehaltenen Zustand befindet, wird der Vorgang nicht als beendet angesehen, sodass er ignoriert wird, selbst wenn ein weiterer Betriebsbefehl durch eine externe Steuerung gegeben wird.
Beispiel	“task pause”

Skriptcode	task stop
Beschreibung	Dieser Befehl stoppt die laufende Roboterbewegung. Es kann sowohl für den Aktionsbefehl durch externe Steuerung als auch für die Arbeitsdatei verwendet werden, die aufgrund des Befehls“task play” ausgeführt wurde. Wenn eine Aktion ausgesetzt wird, wird die Aktion nicht mit “task resume_a” fortgesetzt. Die Aktion ist vollständig beendet. Wenn Sie einen Stoppbefehl geben, kann der Roboter plötzlich anhalten, während er sich schnell bewegt. Es wird empfohlen, den Befehl “task stop” nach “task Pause” auszuführen.
Beispiel	“task stop”

© RAINBOW ROBOTICS Inc. All rights reserved.

Skriptcode	task resume_a
Beschreibung	Dieser Befehl setzt den Betrieb des Roboters fort, der durch “task Pause” oder durch den Alarm- oder Debug-Befehl in der Arbeitsdatei angehalten wurde.
Beispiel	“task resume_a”

Skriptcode	task resume_b
Beschreibung	Dieser Befehl setzt die Bewegung des Roboters fort, der aufgrund einer externen Kollisionserkennung angehalten wurde.
Beispiel	“task resume_b”

Um die externe Steuerung zu verwenden, muss der externe Computer an die Steuerbox angeschlossen werden. Die Verbindung nutzt TCP/IP-Kommunikation und die entsprechende IP-Adresse kann im Programmierhandgerät eingestellt werden. Das Ergebnis wird auf dem Bildschirm des Bedienfelds angezeigt. Die Port 5000 und 5001 sind für die externe Steuerung geöffnet. Port 5000 ist ein Anschluss zum Empfangen von Befehlen, und Port 5001 ist ein Anschluss zum Anfordern und Senden von Daten, die den Roboterstatus angeben. Einfachheit halber wird Port 5000 als Befehlsport und Port 5001 als Datenport bezeichnet.

Der Benutzer kann den oben beschriebenen Skriptbefehl an den Befehlsport senden. Der Befehlsport hat einen Filter für den ersten Befehl, wenn also der Start kein Skriptbefehl wie oben beschrieben ist, wie z. B. “jointall”, “movetcp”, “mc”, “pgmode” usw., zeigt die Antwort “The command is not allowed”. Wenn der Befehl mit einem normalen Befehl beginnt und die Eingabeanweisung an den Parser übergibt, zeigt die Antwort “The command was executed”.

Wenn der Befehl “reqdata” an den Datenport gesendet wird, werden als Antwort Roboterstatusinformationen an den Datenport gesendet. Das Format der Daten ist unten gezeigt.

Header (4 Byte)				Daten (n Byte)
0x24	Size&0xFF	(Size>>8)&0xFF	0x03	Data

© RAINBOW ROBOTICS Inc. All rights reserved.

Das Format der Daten ist unten gezeigt. Je nach Systemversion kann die Größe der Daten unterschiedlich sein. Die Reihenfolge ist jedoch konsistent, siehe Tabelle unten.

Offset	Typ	Beschreibung
0	Float	Verstrichene Aufgabenzeit (in Sekunde – zu Beginn der Aufgabe zurückgesetzt)
1	Float	Bezugswinkel des Basisgelenks (in Grad)
2	Float	Bezugswinkel des Schultergelenks (in Grad)
3	Float	Bezugswinkel des Ellbogengelenks (in Grad)
4	Float	Bezugswinkel des Handgelenks 1 (in Grad)
5	Float	Bezugswinkel des Handgelenks 2 (in Grad)
6	Float	Bezugswinkel des Handgelenks 3 (in Grad)
7	Float	Encoderwinkel des Basisgelenk (in Grad)
8	Float	Encoderwinkel des Schultergelenks (in Grad)
9	Float	Encoderwinkel des Ellbogengelenks (in Grad)
10	Float	Encoderwinkel des Handgelenks 1 (in Grad)
11	Float	Encoderwinkel des Handgelenks 2 (in Grad)
12	Float	Encoderwinkel des Handgelenks 3 (in Grad)
13	Float	Stromwert des Basisgelenks (in Ampere)
14	Float	Stromwert des Schultergelenks (in Grad)
15	Float	Stromwert des Ellbogengelenks (in Grad)
16	Float	Stromwert des Handgelenks 1 (in Grad)
17	Float	Stromwert des Handgelenks 2 (in Grad)
18	Float	Stromwert des Handgelenks 3 (in Grad)
19	Float	Referenzposition des TCP in X-Richtung (in mm)
20	Float	Referenzposition des TCP in Y-Richtung (in mm)
21	Float	Referenzposition des TCP in Z-Richtung (in mm)
22	Float	Referenzausrichtungswinkel von TCP RX-Wert (in Grad)
23	Float	Referenzausrichtungswinkel von TCP RY-Wert (in Grad)
24	Float	Referenzausrichtungswinkel von TCP RZ-Wert (in Grad)
25	Float	Gleich wie 19
26	Float	Gleich wie 20
27	Float	Gleich wie 21
28	Float	Gleich wie 22
29	Float	Gleich wie 23
30	Float	Gleich wie 24
31	Float	Spannung am analogen Eingangsanschluss Nr. 0 (in Voltage)
32	Float	Spannung am analogen Eingangsanschluss Nr. 1 (in Voltage)
33	Float	Spannung am analogen Eingangsanschluss Nr. 2 (in Voltage)
34	Float	Spannung am analogen Eingangsanschluss Nr. 3 (in Voltage)
35	Float	Spannung am analogen Ausgangsanschluss Nr. 0 (in Voltage)
36	Float	Spannung am analogen Ausgangsanschluss Nr. 1 (in Voltage)
37	Float	Spannung am analogen Ausgangsanschluss Nr. 2 (in Voltage)
38	Float	Spannung am analogen Ausgangsanschluss Nr. 3 (in Voltage)
39	Int	Spannung am digitalen Eingangsanschluss Nr. 0 (on:1 / off:0)
40	Int	Spannung am digitalen Eingangsanschluss Nr. 1 (on:1 / off:0)
41	Int	Spannung am digitalen Eingangsanschluss Nr. 2 (on:1 / off:0)
42	Int	Spannung am digitalen Eingangsanschluss Nr. 3 (on:1 / off:0)
43	Int	Spannung am digitalen Eingangsanschluss Nr. 4 (on:1 / off:0)

© RAINBOW ROBOTICS Inc. All rights reserved.

44	Int	Spannung am digitalen Eingangsanschluss Nr. 5 (on:1 / off:0)
45	Int	Spannung am digitalen Eingangsanschluss Nr. 6 (on:1 / off:0)
46	Int	Spannung am digitalen Eingangsanschluss Nr. 7 (on:1 / off:0)
47	Int	Spannung am digitalen Eingangsanschluss Nr. 8 (on:1 / off:0)
48	Int	Spannung am digitalen Eingangsanschluss Nr. 9 (on:1 / off:0)
49	Int	Spannung am digitalen Eingangsanschluss Nr. 10 (on:1 / off:0)
50	Int	Spannung am digitalen Eingangsanschluss Nr. 11 (on:1 / off:0)
51	Int	Spannung am digitalen Eingangsanschluss Nr. 12 (on:1 / off:0)
52	Int	Spannung am digitalen Eingangsanschluss Nr. 13 (on:1 / off:0)
53	Int	Spannung am digitalen Eingangsanschluss Nr. 14 (on:1 / off:0)
54	Int	Spannung am digitalen Eingangsanschluss Nr. 15 (on:1 / off:0)
55	Int	Spannung am digitalen Ausgangsanschluss Nr. 0 (on:1 / off:0)
56	Int	Spannung am digitalen Ausgangsanschluss Nr. 1 (on:1 / off:0)
57	Int	Spannung am digitalen Ausgangsanschluss Nr. 2 (on:1 / off:0)
58	Int	Spannung am digitalen Ausgangsanschluss Nr. 3 (on:1 / off:0)
59	Int	Spannung am digitalen Ausgangsanschluss Nr. 4 (on:1 / off:0)
60	Int	Spannung am digitalen Ausgangsanschluss Nr. 5 (on:1 / off:0)
61	Int	Spannung am digitalen Ausgangsanschluss Nr. 6 (on:1 / off:0)
62	Int	Spannung am digitalen Ausgangsanschluss Nr. 7 (on:1 / off:0)
63	Int	Spannung am digitalen Ausgangsanschluss Nr. 8 (on:1 / off:0)
64	Int	Spannung am digitalen Ausgangsanschluss Nr. 9 (on:1 / off:0)
65	Int	Spannung am digitalen Ausgangsanschluss Nr. 10 (on:1 / off:0)
66	Int	Spannung am digitalen Ausgangsanschluss Nr. 11 (on:1 / off:0)
67	Int	Spannung am digitalen Ausgangsanschluss Nr. 12 (on:1 / off:0)
68	Int	Spannung am digitalen Ausgangsanschluss Nr. 13 (on:1 / off:0)
69	Int	Spannung am digitalen Ausgangsanschluss Nr. 14 (on:1 / off:0)
70	Int	Spannung am digitalen Ausgangsanschluss Nr. 15 (on:1 / off:0)
71	Float	Temperatur Motorantrieb am Basisgelenk (in Celsius)
72	Float	Temperatur Motorantrieb am Schultergelenk (in Celsius)
73	Float	Temperatur Motorantrieb am Ellbogengelenk (in Celsius)
74	Float	Temperatur Motorantrieb am Handgelenk 1 (in Celsius)
75	Float	Temperatur Motorantrieb am Handgelenk 2 (in Celsius)
76	Float	Temperatur Motorantrieb am Handgelenk 3 (in Celsius)
77	Int	Speicherort des Programmzählers in der Aufgabe (Der Speicherort, an dem der Step-Befehl ausgeführt wird)
78	Int	Gewünschte Anzahl an Wiederholungen
79	Int	Aktuelle Aktionsnummer der Aufgabe
80	Int	Aktuelle Anzahl der Wiederholungen
81	Float	Verstrichene Aufgabenzeit (in Sekunde – zu Beginn der Aufgabe nicht zurückgesetzt)
82	Int	Aufgabenstatus (1: Idle, 2: Paused, 3: Run)
83	Float	Bewegungsgeschwindigkeit (0~1)
84	Float	Roboterstatus (1: gestoppt, 3: in Betrieb)
85	Float	Status der Leistung in Bezug auf den LSB Offset 0: 48V-Eingang 1: 48V-Ausgang 2: 24V-Status 3: Not-Aus-Status 4: Status des PC-Schalters 5: Status von Motorsteuer
86	Float	Not used

© RAINBOW ROBOTICS Inc. All rights reserved.

87	Float	Not used
88	Float	Not used
89	Float	Not used
90	Float	Not used
91	Float	Not used
92	Int	Status der Motorsteuerung am Basisgelenk
93	Int	Status der Motorsteuerung am Schultergelenk
94	Int	Status der Motorsteuerung am Ellbogengelenk
95	Int	Status der Motorsteuerung am Handgelenk 1
96	Int	Status der Motorsteuerung am Handgelenk 2
97	Int	Status der Motorsteuerung am Handgelenk 3
		Status der Motorsteuerung in Bezug auf den LSB Offset 0: FET 1: Positionkontorlle 2: Status der Initialisierung 3: Im Steuermodus 4: Nonius-Fehler 5: Low Battery 6: Im Kalibriermodus 7: Multiturn-Fehler 8: JAM-Fehler 9: Überstromfehler 10: Großer Fehler 11: Eingabefehler 12: FET-Treibfehler 13: Temperaturfehler 14: Positionsfehler (Low) 15: Positionsfehler (High)
98	Int	Kollisionserkennung (1: on, 0: off)
99	Int	Direkt-Teaching (1: on, 0: off)
100	Int	Betriebsmodus (1: Simulation mode, 0: real mode)
101	Int	Informationen zum Initialisierungsprozess 0: Standard 1: Spannungsprüfung 2: Geräteprüfung 3: Start der Lageregelung 4: Variablenprüfung 5: Kollisionsprüfung 6: Initialisierung abgeschlossen
102	Int	Fehlercode bei der Initialisierung 0: keine Fehler 1: SMPS-Fehler 2: Not-Aus-Schalter-Fehler 3: Leistungsumwandlungsfehler1 (in Steuerbox) 4: Leistungsumwandlungsfehler2 (in Steuerbox) 5: Verbindungsfehler 6: Initialisierungsfehler 7: Inspektionsfehler bei der Nutzlasteinstellung 8: Verbindungsfehler des Werkzeugflansches 9: Orientierungsfehler des Werkzeugflansches 10: Encoderfehler der Motorsteuerung 1 11: Encoderfehler der Motorsteuerung 2

© RAINBOW ROBOTICS Inc. All rights reserved.

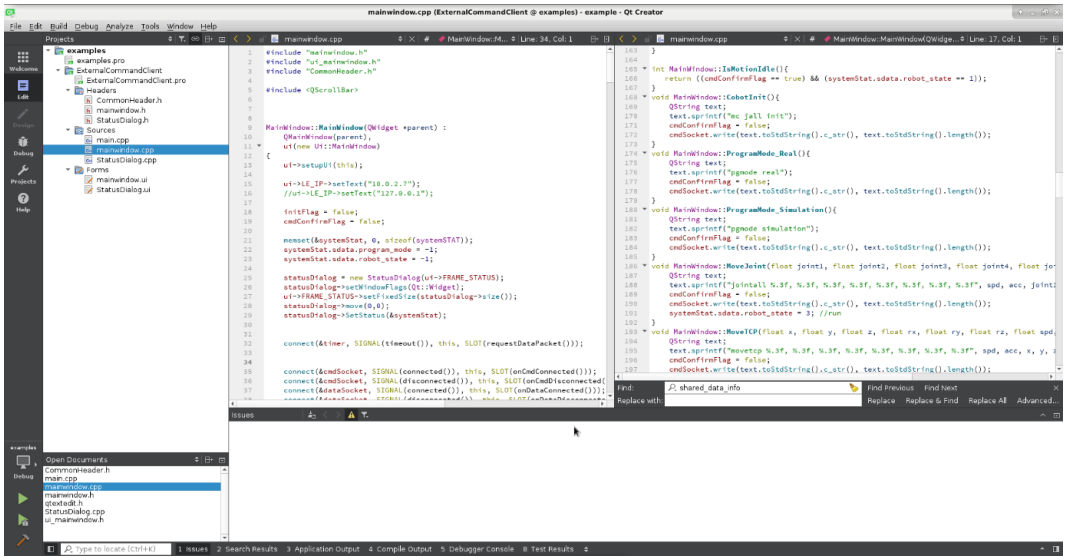
		12: Digitaler Eingang 16/17 Kurzschlussfehler 13: 48V-Schalterfehler 14: Fehler bei Direkt-Teaching-Taste
103	Float	Spannung des analogen Signalanschluss Nr. 0 am Werkzeugflansch (in voltage)
104	Float	Spannung des analogen Signalanschluss Nr. 1 am Werkzeugflansch (in voltage)
105	Int	Status des digitalen Eingangsanschlusses Nr. 0 am Werkzeugflansch (on:1 / off:0)
106	Int	Status des digitalen Eingangsanschlusses Nr. 1 am Werkzeugflansch (on:1 / off:0)
107	Int	Status des digitalen Ausgangsanschlusses Nr. 0 am Werkzeugflansch (on:1 / off:0)
108	Int	Status des digitalen Ausgangsanschlusses Nr. 1 am Werkzeugflansch (on:1 / off:0)
109	Float	Spannungsausgang am Werkzeugflansch (in voltage)
110	Int	Status der Kollisionserkennung (on:1 / off:0)
111	Int	Status von Gerätenfehlern 0: Keine Fehler 1: PVL-Fehler 2: CPU- Fehler 3: Großer Fehler 4: Input- Fehler 5: JAM- Fehler 6: Überstromfehler 7: Gelenkwinkelfehler 8: Fehler im Steuermodus 9: Offsetfehler zwischen Referenz und Encoder 10: Stromfehler an übergeordneter Steuerung 11: Temperaturfehler 12: Geschwindigkeitsfehler bei Direkt-Teaching
112	Int	Selbstkollisionserkennung (on:1 / off:0)
113	Int	Roboter pausiert (on:1 / off:0)
114	Int	Status von Bewegungsfehlern 0: keine Fehler 1: TCP-Bewegung befohlen, wenn der Roboter vollständig ausgestreckt ist 2: TCP-Befehl nicht erreichbar 3: Der gemeinsame Befehl hat die mechanische Grenze überschritten 4: Singularität des TCP-Befehls
115	Int	Spannung am digitalen Eingangsanschluss Nr. 16 (on:1 / off:0)
116	Int	Spannung am digitalen Eingangsanschluss Nr. 17 (on:1 / off:0)
117	Int	Inbox 0 Trap aufgetreten
118	Int	Inbox 1 Trap aufgetreten
119	Int	Inbox 0 Prüfmodus
120	Int	Inbox 1 Prüfmodus

© RAINBOW ROBOTICS Inc. All rights reserved.

E.2 Beispiel von Programmentwicklungsumgebung (Development Environment)

Dieses Beispiel wurde auf Debian 9.8 und Ubuntu 18.04 getestet. Es funktioniert möglicherweise auf ähnlichen Linux-Systemen. Es ist kein separater Kernel-Patch erforderlich.

Als integrierte Entwicklungsumgebung(DIE, Integrated Development Environment) zum Programmieren verwenden Sie Qt Version 5.8 (https://www.qt.io).



Vorsicht:

Qt-basierte C++ Beispiele, Visual Studio-basierte C# -Beispiele und mehr. Beispielprogramme sind beim Hersteller oder Vertreiber erhältlich

© RAINBOW ROBOTICS Inc. All rights reserved.

E.3 Programmiermethode

Dieses Beispiel enthält nicht alle Funktion, die von der Tablet-UI(User Interface) bereitgestellt werden. Nur die Informationen, die für die Überwachung durch den Benutzer nützlich sind, während der Roboter durch externe Steuerung bewegt wird, werden implementiert.

Das folgende Bild ist die Programmier-UI, wenn das Beispielprogramm ausgeführt wird.

Die Funktion jeweiligen Elements ist wie folgt.

1) Netzwerk verbinden

Stellen Sie über den LAN-Anschluss an der Steuerbox des RB-Serie Roboters eine Verbindung zur Hauptsteuerung des Roboters her. Die Standard-IP-Adresse für die externe Steuerung ist fest auf '10.0.2.7' eingestellt. Der Server zum Empfangen externer Steuerbefehle wird mit Prot 5000 verbunden, und der Server zum Anfordern und Empfangen von Roboterstatusinformationen wird mit Port 5001 verbunden. Für die Verbindung gibt es jeweils eine separate Schaltfläche. Wenn die Verbindung erfolgreich ist, ändert sich das Wort 'Connect' auf der Schaltfläche in 'Disconnect'. Das Gegenteil passiert, wenn die Verbindung unterbrochen wird.

© RAINBOW ROBOTICS Inc. All rights reserved.

2) Roboter initialisieren

Nachdem Sie sich mit der Hauptsteuerung des Roboters verbunden haben, drücken Sie die 'Cobot Init'-Taste, um den Initialisierungsprozess zu starten. Roboter durchlaufen eine Reihe von Prozessen namens 'Power Set', 'Device Set', 'System Set' und 'Robot Init'. Während der Initialisierungsprozess des Roboters fortgesetzt wird, wird das weiße Bearbeitungsfeld vor jedem Kurs gelb. Angeschlossene Prozesse werden grün und nicht durchgeführte Prozesse bleiben rot. Wenn alle vier Kästchen grün werden, ist der Initialisierungsprozess des Roboters abgeschlossen und einsatzbereit.

3) Roboterstatus

Der Status des Roboters kann aus dem vom Hauptprogramm in der Steuerbox empfangenen Daten bekannt sein. Diese Daten werden als Antwort auf eine Anfrage nach 'reqdata' auf Port 5001 an das Hauptprogramm gesendet. Das Format der Daten wird in Form der 'systemSTAT'-Struktur in 'CommonHeader.h' übergeben.

Betriebsmodus(run mode): Zeigt den Betriebsmodus des Roboters an. Es gibt Realmodus und Simulationsmodus. Im Real-Modus werden Bewegungsbefehle tatsächlich auf den Roboter angewendet und der Roboter bewegt sich. Im Simulationsmodus wird die Bewegung ausgeführt, aber der Befehl wird nicht an den Roboter gesendet. Das Programmierhandgerät zeigt, wie sich der durchscheinende Roboter bewegt. Der Roboterbetriebsmodus wird durch den Wert der 'program_mode'-Variablen in 'systemSTAT' dargestellt. Ein Wert von 0 für diese. Variable steht für Real-Modus und ein Wert von 1 für Simulation-Modus.

Roboterzustand(robot state): Zeigt an, ob sich der Roboter gerade bewegt oder in einem Zustand ist, in dem er Bewegungsbefehle empfangen kann. Der Roboterstatus kann aus dem Wert der 'robot_state'-Variablen von 'systemSTAT' ermittelt werden. Wenn dieser Wert 1 ist, kann der Ruhezustand Bewegungsbefehle empfange. Wenn der Wert 3 ist, bewegt sich der Roboter. Bewegungsbefehle werden ignoriert, während der Roboter in Bewegung ist. Wenn der Wert 2 ist, wird der Roboter aus unbestimmten Gründen angehalten oder durch den Pause-Befehl angehalten. In diesem Fall wird es in der 'status'-Splate als 'paused' angezeigt.

Status: Zeigt den aktuellen Sonderbetriebsstatus oder abnormalen Status des Roboters wie folgt an. Wenn Direkt-Teaching ausgeführt wird, ist 'teaching' möglich. Wenn sie durch externe Kollisionserkennung gestoppt wird, ist 'ext. collision' möglich. Wenn sie während des Betriebs kurz vor einer Selbstkollision stattfindet, ist 'self collision' möglich. Wenn sie durch einen Pausenbefehl gestoppt wird, ist 'paused' möglich. Wenn eine Eingabe ohne Lösung im Robotersteuerungsalgorithmus eingeht, ist 'ems' möglich. Ein Stromproblem oder ein Problem mit der Robotersteuerung ändert die Farbe des 'sos'-Bearbeitungsfenster. Dies wird durch Bezugnahme auf den Wert von 'op_stat_collision_occur', 'op_stat_sos_flag', 'op_stat_self_collision', 'op_stat_soft_estop_occur', 'op_stat_ems_flag' und 'robot_state' in'systemSTAT' angezeigt.

© RAINBOW ROBOTICS Inc. All rights reserved.

joint reference: zeigt den Referenzeingabewert für jedes Gelenks (in Grad).

joint encoder: zeigt den aktuellen Encoder-Wert jedes Gelenks an (in Grad).

TCP reference: zeigt den Referenzpositionswert des TCP an (in mm und Grad).

digital in: zeigt den digitalen Eingangswert der Steuerbox an.

digital out: zeigt den digitalen Ausgangswert der Steuerbox an.

analog in: zeigt den analogen Eingangswert der Steuerbox an (in V).

analog out: zeigt den analogen Ausgangswert der Steuerbox an (in V).

tool out voltage: zeigt die Ausgangsspannung der aktuell eingestellten Werkzeugflanschplatine an (0V, 12V oder 24V).

tool digital in: zeigt den digitalen Eingangswert der Werkzeugflanschplatine an.

tool digital out: zeigt den digitalen Ausgangswert der Werkzeugflanschplatine an.

tool analog in: zeigt den analogen Eingangswert der Werkzeugflanschplatine an.

4) Modus wechseln

Der Roboter kann zwei Betriebsmodi haben(Simulationsmodus und Realmodus). Im Simulationsmodus bewegt sich der Roboter nicht, aber der Wert der Eingangsreferenz kann geändert werden. Im Real-Modus bewegt sich der Roboter tatsächlich als Reaktion auf Benutzereingaben. Um den Betriebsmodus zu wechseln, drücken Sie die mit ‘Real’ und ‘Simulation’ gekennzeichneten Tasten. Unmittelbar nach dem Initialisierungsprozess befindet sich der Roboter im Simulationsmodus.

5) Geschwindigkeit ändern

Passen Sie die Gesamtgeschwindigkeit der Roboterbewegung an. Benutzer können den Schieberregler zwischen 0% und 100% verschieben. Diese Geschwindigkeit wird mit der Geschwindigkeit multipliziert, die dem Bewegungsbefehl des Roboters gegeben wird.

6) Bewegung anhalten und fortsetzen

Drücken Sie die ‘Motion Pause‘-Taste, um die Bewegung anzuhalten. Um die Bewegung vollständig zu stoppen, drücken Sie ‘Motion Halt’-Taste. Im Fall von ‘Motion Halt’ stoppt der Roboter abrupt, daher wird empfohlen, zuerst die Pause zu verwenden, um den Roboter stabil zu verwenden. Wenn sich der Roboter im angehaltenen Zustand befindet, wird er nicht betrieben, selbst wenn ein anderer Roboter einen Bewegungsbefehl erhält. Um die aktuelle Operation zu stoppen und eine andere Operation durchzuführen, muss die aktuelle Operation nach dem Drücken

© RAINBOW ROBOTICS Inc. All rights reserved.

der ‘Motion Pause’-Taste über die ‘Motion Halt’-Schaltfläche vollständig beendet werden.

Umgekehrt können Benutzer die angehaltene oder durch externe Kollisionserkennung gestoppte Bewegung fortsetzen. Drücken Sie die ‘Motion Resume’-Taste, um eine angehaltene Bewegung fortzusetzen, oder drücken Sie die ‘Collision Resume’-Taste, um eine Bewegung fortzusetzen, die aufgrund einer externen Kollisionserkennung gestoppt wurde.

7) Debugging-Meldungsbildschirm

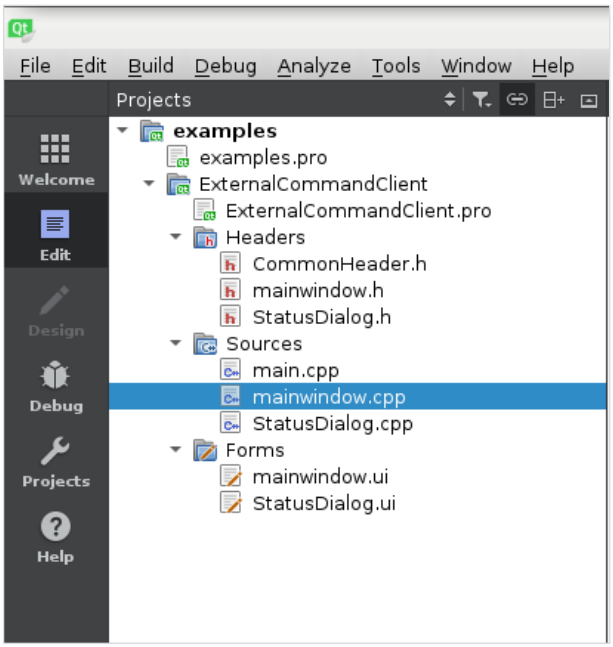
Dies ist das Fenster, in dem Benutzer Meldungen zum Debuggen anzeigen können.

8) Bewegung testen

Drücken Sie die ‘Motion Test’-Taste, um drei grundlegende Bewegungen nacheinander auszuführen. Bitte berücksichtigen Sie die Umgebung des Roboters.

Wenn der Benutzer auf die Schaltfläche ‘Get Joint and TCP’ klickt, werden der Referenzwinkel und der TCP-Wert des Gelenks des aktuellen Roboters mit dem Komma ‘,’ im Bearbeitungsfenster daneben ausgedrückt. Es ist hilfreich, diesen Wert beim Codieren des Roboterbewegungsablaufs in das Programm zu kopieren.

Dieses Beispiel ist ein einzelnes Prozessbeispiel mit einer GUI. In Qt können Benutzer die GUI einfach platzieren, Ereignissen wie Schaltflächenklicks generieren und sie mit dem Programm verknüpfen. Siehe ‘mainwindow.ui’.



© RAINBOW ROBOTICS Inc. All rights reserved.

Die Kerninhalte des Beispiels sind in ‘mainwindow.cpp’ und ‘mainwindow.h’ enthalten. Sie können das Format der Roboterstatusdaten in ‘CommonHeader.h’ finden.

```
    // Cobot Control API -------------------
    // <CobotInit>
    // : initialize Cobot
    void CobotInit();

    // <MoveJoint>
    // : move to target posture in joint coordinate
    // joint1~joint6 : target joint angle in deg unit
    // spd : speed parameter (0~1: user define   or   -1: default setting)
    // acc : acceleration parameter (0~1: user define   or   -1: default setting)
    void MoveJoint(float joint1, float joint2, float joint3, float joint4, float joint5, flo

    // <MoveTCP>
    // : move to target posture in cartesian coordinate
    // x, y, z : target TCP(tool center point) position in mm unit
    // rx, ry, rz : target TCP orientation (Yaw-Pitch-Roll Euler angle) in degree unit
    // spd : speed parameter (0~1: user define   or   -1: default setting)
    // acc : acceleration parameter (0~1: user define   or   -1: default setting)
    void MoveTCP(float x, float y, float z, float rx, float ry, float rz, float spd = -1, fl

    // <ControlBoxDigitalOut>
    // control digital out ports in control box
    // d0~d15 : digital out value (0 or 1)
    void ControlBoxDigitalOut(int d0, int d1, int d2, int d3, int d4, int d5, int d6, int d7

    // <ControlBoxAnalogOut>
    // control analog out ports in control box
    // a0~a3 : analog out value in voltage unit (0~10)
    void ControlBoxAnalogOut(float a0, float a1, float a2, float a3);

    // <ToolOut>
    // control digital out ports and voltage level in tool flange board
    // volt : reference voltage of tool flange board in voltage unit(0, 12, 24)
    // d0, d1 : digital out value (0 or 1)
    void ToolOut(int volt, int d0, int d1);

    // <ProgramMode_Real>
    // change to 'real robot' mode -- robot will move
    void ProgramMode_Real();

    // <ProgramMode_Simulation>
    // change to 'simulation' mode -- robot will not move except teaching
    void ProgramMode_Simulation();

    // <BaseSpeedChange>
    // change base speed -- base speed will be multiplied to motion velocity
    // spd : normalized base speed (0~1)
    void BaseSpeedChange(float spd);

    // <MotionPause>
    // pause the current motion
    void MotionPause();

    // <MotionHalt>
    // halt the current motion
    // !! CAUTION : user would better escape the motion sequence
    //              : if not, the next motion will be activated immediately
    void MotionHalt();

    // <MotionResume>
    // resume the paused motion
    void MotionResume();

    // <CollisionResume>
    // resume the motion which is paused due to external collision
    void CollisionResume();
    // -------------------------------------
```

© RAINBOW ROBOTICS Inc. All rights reserved.

Robotersteuerbefehle, die Benutzer verwenden können, sind wie oben in ‘mainwindow.h’ angegeben. Im Folgenden finden Sie ausführliche Erklärungen.

Funktion	CobotInit(void)
Skript	“mc jall init”
Beschreibung	Die Hardware des Roboters wird initialisiert. Der Initialisierungsprozess kann über ‘init_stat_info’ und ‘init_error’ von ‘systemSTAT’ gefunden werden.

Funktion	MoveJoint(float joint1, float joint2, float joint3, float joint4, float joint5, float joint6, float spd = -1, float acc = -1);
Skript	“jointall spd, acc, joint1, joint2, joint3, joint4, joint5, joint6”
Beschreibung	Der Roboter bewegt sich im Gelenkraum(Joint Space) mit dem eingegebenen Gelenkwert als Ziel. joint1 - joint6 sind die Zielgelenkwerte in Gradeinheiten. spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet.

Funktion	MoveTCP(float x, float y, float z, float rx, float ry, float rz, float spd = -1, float acc = -1);
Skript	“movetcp spd, acc, x, y, z, rx, ry, rz”
Beschreibung	Der Roboter bewegt sich im kartesischen Raum (Cartesian Space) mit dem Ziel der eingegebenen TCP-Position und dem Drehwinkelwert. x, y, z sind Positionswerte in TCP in mm . rx, ry und rz sind die in Gradeinheiten von Euler-Winkeln in der ‚Yaw-Pitch-Roll‘-Reihenfolge ausgedrückten Werte zur Bestimmung der TCP-Richtung.

© RAINBOW ROBOTICS Inc. All rights reserved.

	spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet.

Funktion	MoveCircle_ThreePoint(int type, float x1, float y1, float z1, float rx1, float ry1, float rz1, float x2, float y2, float z2, float rx2, float ry2, float rz2, float spd = -1, float acc = -1);
Skript	“movecircle threepoints intended spd, acc, x1, y1, z1, rx1, ry1, rz1, x2, y2, z2, rx2, ry2, rz2” “movecircle threepoints constant spd, acc, x1, y1, z1, rx1, ry1, rz1, x2, y2, z2, rx2, ry2, rz2” “movecircle threepoints radial spd, acc, x1, y1, z1, rx1, ry1, rz1, x2, y2, z2, rx2, ry2, rz2”
Beschreibung	Dies ist eine Aktionsbefehl zum Zeichnen eines Kreises unter Verwendung von drei Punkten der aktuellen Position, der Zwischenposition und der Endposition des TCP. Je nach Typ kann die TCP-Richtung auf drei Arten bestimmt werden. type=0: wird als ‘intended’ in Skripten verwendet. Der Roboter bewegt sich unter Beibehaltung der vom Benutzer eingegeben Richtung. type=1: wird als ‘constant’ in Skripten verwendet. Ein Kreis wird unter Beibehaltung der aktuellen Ausrichtung von TCP gezeichnet. type=2: wird als‘radial’ in Skripten verwendet. Die Richtung des TCP wird relative in Richtung der Mitte des durch die drei Positionen bestimmten Kreises geändert. Die Richtung basiert auf der aktuellen TCP-Richtung als Ausgangspunkt. x1, y1, z1 sind die Zwischenpositionswerte von TCP in mm. rx1, ry1 und rz1 sind die in Gradeinheiten von Euler-Winkeln in der ‚Yaw-Pitch-Roll‘-Reihenfolge, um die Richtung der Zwischenposition von TCP zu bestimmen. x2, y2, z2 sind die Endpositionswerte von TCP in mm.

© RAINBOW ROBOTICS Inc. All rights reserved.

	rx2, ry2 und rz2 sind die in Gradeinheiten von Euler-Winkeln in der ‚Yaw-Pitch-Roll'-Reihenfolge, um die Richtung der Endposition von TCP zu bestimmen. spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	MoveCircle_Axis(int type, float cx, float cy, float cz, float ax, float ay, float az, float rot_angle, float spd = -1, float acc = -1);
Skript	“movecircle axis intended spd, acc, rot_angle, cx, cy, cz, ax, ay, az” “movecircle axis constant spd, acc, rot_angle, cx, cy, cz, ax, ay, az” “movecircle axis radial spd, acc, rot_angle, cx, cy, cz, ax, ay, az”
Beschreibung	Dies ist ein Aktionsbefehl, der von der aktuellen Position des TCP ausgeht und einen Kreis mit dem eingegebenen Winkel um die Rotationsachse zeichnet. Je nach Typ kann die TCP-Richtung auf drei Arten bestimmt werden. type=0: wird als ‘intended’ in Skripten verwendet. Der Roboter bewegt sich unter Beibehaltung der vom Benutzer eingegeben Richtung. type=1: wird als ‘constant’ in Skripten verwendet. Ein Kreis wird unter Beibehaltung der aktuellen Ausrichtung von TCP gezeichnet. type=2: wird als‘radial’ in Skripten verwendet. Die Richtung des TCP wird relative in Richtung der Mitte des durch die drei Positionen bestimmten Kreises geändert. Die Richtung basiert auf der aktuellen TCP-Richtung als Ausgangspunkt. cx, cy, cz sind die Positionswerte des Drehzentrums in mm. ax, ay, az sind Vektoren, die die Drehachse darstellen. rot_angle ist ein Wert, der den zu drehenden Betrag in Grad dargestellt. spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet.

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	MoveJointBlend_Clear(void);
Skript	“blend_jnt clear_pt”
Beschreibung	Dies ist ein Befehl zum Löschen der gespeicherten Zielorte unter Aktionsbefehlen, die aufeinanderfolgende Zielorte im Joint Space verbinden.

Funktion	MoveJointBlend_AddPoint(float joint1, float joint2, float joint3, float joint4, float joint5, float joint6, float spd = -1, float acc = -1);
Skript	“blend_jnt add_pt spd, acc, joint1, joint2, joint3, joint4, joint5, joint6”
Beschreibung	Dies ist ein Befehl zum Eingeben der Zielorte unter Aktionsbefehlen, die aufeinanderfolgende Zielorte im Joint Space verbinden. joint1 - joint6 sind gemeinsame Zielwert in Grad. spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet. Der gesamte Befehl wird von den zuletzt eingegeben ‚spd‘- und ‚acc‘-Werte gesteuert.

Funktion	MoveJointBlend_MovePoint(void);
Skript	“blend_jnt move_pt”
Beschreibung	Dies ist ein Befehl zum Starten einer Bewegung unter Aktionsbefehlen, die aufeinanderfolgende Zielpositionen in Joint Space verbinden. Bewegungen werden sequenziell zu den mit MoveJointBlend_AddPoint eingetragenen Positionen durchgeführt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	MoveTCPBlend_Clear(void);
Skript	“blend_tcp clear_pt”
Beschreibung	Dies ist ein Befehl zum Löschen von gespeicherten Zielorten unter Aktionsbefehlen, die aufeinanderfolgende Zielorte in kartesischen Raum(Cartesian Space)verbinden.

Funktion	MoveTCPBlend_AddPoint(float radius, float x, float y, float z, float rx, float ry, float rz, float spd = -1, float acc = -1);
Skript	“blend_tcp add_pt spd, acc, radius, x, y, z, rx, ry, rz”
Beschreibung	Dies ist ein Befehl zum Eingeben von Zielorten unter Aktionsbefehlen, die aufeinanderfolgende Zielorte in kartesischen Raum(Cartesian Space)verbinden. ‚radius‘ ist ein Wert in mm, der die zu bewegende Distanz zur nächsten Zielposition angibt. Indem eine Kurve gezeichnet wird, wenn die Zielposition nacheinander bewegt werden. Der Maximalwert des ‚radius‘ ist die Hälfte der Länge des Liniensegments, das sich von einem Ziel zum nächsten bewegt. Wenn Sie einen größeren Wert eingeben, wird der Wert intern reduziert verwendet. x, y, z sind Positionswert von TCP in mm. rx, ry und rz sind die in Gradeinheiten von Euler-Winkeln in der ‚Yaw-Pitch-Roll‘-Reihenfolge ausgedrückten Werte zur Bestimmung der TCP-Richtung. spd und acc haben Werte, die von 0 bis 1 für normalisierte Geschwindigkeit bzw. Beschleunigung sind. Wenn Sie -1 eingeben, werden die Standardgeschwindigkeit und -beschleunigung verwendet. Der gesamte Befehl wird von den zuletzt eingegeben ‚spd‘- und ‚acc‘-Werte gesteuert.

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	MoveTCPBlend_MovePoint(void);
Skript	“blend_tcp move_pt”
Beschreibung	Dies ist ein Befehl zum Starten einer Bewegung unter Bewegungsbefehlen, die aufeinanderfolgende Zielort in kartesischen Raum(Cartesian Space) verbinden. Bewegungen werden sequenziell zu den mit ‚MoveTCPBlend_AddPoint’ eingetragenen Positionen ausgeführt.

Funktion	ControlBoxDigitalOut(int d0, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int d8, int d9, int d10, int d11, int d12, int d13, int d14, int d15)
Skript	“digital_out d0, d1, d2, d3, d4,d5, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15”
Beschreibung	Ein Signal wird über den digitalen Ausgangsanschluss der Steuerbox gesendet. d0 - d15 zeigen das Ein/Aus jeweiligen digitalen Anschlusses an. Ein Wert von 1 bedeutet ‚angeschaltet‘, 0 bedeutet ‚ausgeschaltet‘.

Funktion	ControlBoxAnalogOut(float a0, float a1, float a2, float a3)
Skript	“analog_out a0, a1, a2, a3”
Beschreibung	Durch den analogen Ausgangsanschluss der Steuerbox wird ein Signal gesendet. a0 bis a3 geben den Ausgangswert jeweiligen analogen Anschlusses an. Der Wert reicht von 0 bis 10 und wird in V(Voltage) angegeben

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	ToolOut(int volt, int d0, int d1)
Skript	“tool_out volt, d0, d1”
Beschreibung	Spannungspegel am Werkzeugflansch wird festgelegt und ein Signal wird über den digitalen Ausgangsanschluss gesendet. ‘volt’ stellt den Spannungspegel des digitalen Ausgangs des Werkzeugflansches ein. Der Wert kann 0, 12 oder 24 sein und wird in V(voltage) angegeben. Andere Angabe als diese drei Werte werden ignoriert. d0, d1 zeigen Ein/Aus jeweiligen digitalen Anschlusses an, Ein Wert von 1 bedeutet ‚angeschaltet‘, 0 bedeutet ‚ausgeschaltet‘.

Funktion	ProgramMode_Real(void)
Skript	“pgmode real”
Beschreibung	Der Bewegungszustand des Roboters wird in den Real-Modus geändert. Im Realmodus bewegt sich der Roboter, wenn Bewegungsbefehle gegeben werden.

Funktion	ProgramMode_Simulation(void)
Skript	“pgmode simulation”
Beschreibung	Der Bewegungszustand des Roboters wird in den Simulationsmodus geändert. Im Simulationsmodus bewegt sich der Roboter nicht, wenn ein Bewegungsbefehl gegeben wird.

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	BaseSpeedChange(float spd)
Skript	“sdw default_speed spd”
Beschreibung	Es funktioniert durch Multiplizieren der Eingangsgeschwindigkeit bei ‚spd‘ mit der aller Bewegungsbefehle. Das heißt, das ist ein Wert, der die Geschwindigkeit der Gesamtbewegung bestimmt. spd hat normalisierten Wert von 0 bis 1.

Funktion	MotionPause(void)
Skript	“task pause”
Beschreibung	Roboterbewegung wird temporär angehalten. Jeder andere Bewegungsbefehl wird ignoriert, bis die Bewegung mit MotionResume fortgesetzt wird oder die Bewegung mit MotionHalt vollständig beendet wird.

Funktion	MotionResume(void)
Skript	“task resume_a”
Beschreibung	Durch MotionPause angehaltene Roboterbewegung wird fortgesetzt. Aufgrund einer externen Kollision gestoppte Bewegung kann nicht fortgesetzt werden. In diesem Fall müssen Sie ein CollisionResume durchführen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Funktion	CollisionResume(void)
Skript	“task resume_b”
Beschreibung	Die aufgrund einer externen Kollision angehaltene Roboterbewegung wird wieder aufgenommen. Aufgrund der MotionPause gestoppte Bewegung kann nicht fortgesetzt werden. In diesem Fall müssen Sie ein MotionResume durchführen.

Funktion	MotionHalt(void)
Skript	“task stop”
Beschreibung	Die Bewegung des Roboters wird vollständig gestoppt. Die Bewegung eines Roboters, der aufgrund von MotionPause oder einer externen Kollision angehalten ist, wird auch vollständig gestoppt.

© RAINBOW ROBOTICS Inc. All rights reserved.

```
int test_flag = false;
int test_state = 0;
void MainWindow::onLogic(){

    // Motion Sequence ================================================
    if(test_flag == true){
        switch(test_state){
        case 0:
            if(IsMotionIdle()){
                print("test_state 0\n");
                MoveJoint(0,0,0,0,0,0);
                test_state = 1;
            }
            break;

        case 1:
            if(IsMotionIdle()){
                print("test_state 1\n");
                MoveJoint(50,50,50,50,50,50);
                test_state = 2;
            }
            break;
        case 2:
            if(IsMotionIdle()){
                print("test_state 2\n");
                MoveTCP(400,400,400,150,30,-100);
                test_state = 3;
            }
            break;
        case 3:
            if(IsMotionIdle()){
                print("test_state 3\n");
                test_flag = false;
                test_state = 0;
            }
            break;
        }
    }
    // ================================================================

}
```

Der obige Code ist ein Aktionscode, der nacheinander zwei gemeinsame Steuerbewegungen und eine TCP-Steuerbewegung ausführt. Es gibt ein ‘test_flag’, das entscheidet, ob die Bewegungssequenz ausgeführt wird, und wenn dieser Wert wahr ist, bewegt es sich sequenziell von der vorherigen Bewegung zur nächsten Bewegung gemäß dem ‘test_state’-Wert, der den Fortschritt der Sequenz anzeigt.

Überprüfen Sie an dieser Stelle, ob die vorherige Bewegung beendet ist oder nicht, und es gibt eine ‘IsMotionIdle’-Funktion, um es einfacher zu machen. Diese Funktion sendet Anweisungen an die Hauptsteuerung des Roboters.

© RAINBOW ROBOTICS Inc. All rights reserved.

```
165 int MainWindow::IsMotionIdle(){
166     return ((cmdConfirmFlag == true) && (systemStat.sdata.robot_state == 1));
167 }
```

Die Funktion ‘onLogic’, die eine Aktionssequenz enthält, ist mit einem von Qt bereitgestellten Timer verknüpft. In diesem Beispiel ist sie auf 10 ms-Intervalle eingestellt, und diese Funktion wird alle 10 ms ausgeführt.

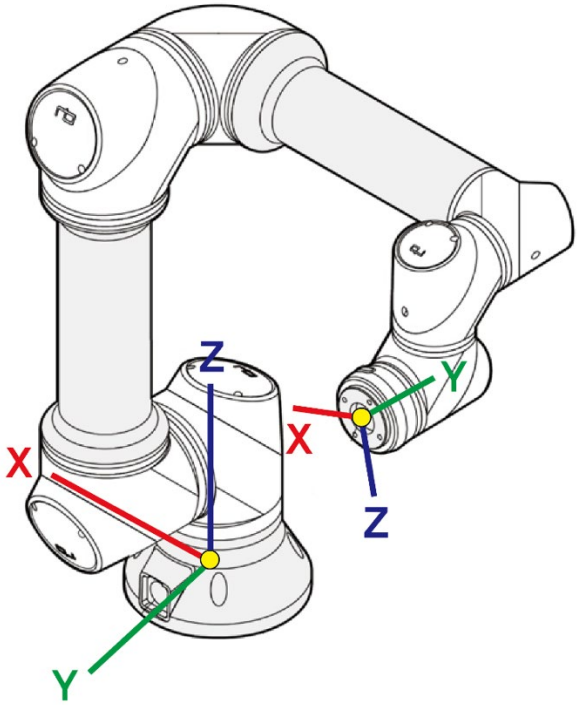
```
409
410 void MainWindow::on_BTN_TEST_clicked()
411 {
412     test_state = 0;
413     test_flag = true;
414 }
415
```

Das Ausführen von Roboterbewegungen ist einfach. Setzen Sie sich den ‘test_state’-Wert, der den Zustand der Bewegungssequenz darstellt, auf 0, den Startpunkt der Bewegung, und setzen Sie ‚true‘ als den ‘test_flag’-Wert, um die Bewegung zum Bewegen des Roboters auszuführen.

Das im bereitgestellten Beispielcode gezeigte Bewegung ist sehr einfach, aber auch frei von Strukturbeschränkungen. Benutzer können ihre eigene Anwendung basierend auf diesem Beispielcode erstellen oder eine separate Anwendung erstellen, indem sie nur das Skript verstehen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang F Einstellung von Koordinatensystem



- Globales Koordinatensystem

  Wenn das Koordinatensystem an der Basis des Roboters fixiert ist, ist das globale Koordinatensystem auch fixiert, sobald der Roboter fixiert ist.

  Als Ursprung wird der Mittelpunkt der Grundfläche genommen. Die Richtung des Roboters vom Ursprung ist die +Z-Richtung, und die Richtung des Verbinders vom Ursprung ist die +Y-Richtung.

- Locales Koordinatensystem

  Als am TCP(Tool Center Point) des Roboters fixiertes Koordinatensystem ändert sich die Richtung der Achse in Echtzeit, wenn der TCP-Offset gesetzt oder verschoben wird.

  Mit TCP als Ursprung wird die Richtung des Roboters vom Ursprung als +Y-Richtung festgelegt, und die Richtung der Teaching-Taste direkt vom Ursprung wird als +Z-Richtung festgelegt.

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang G Stoppzeit und Stoppdistanz

Beim kollaborativen Roboter der RB Serie werden die Roboterstoppzeit und Stoppdistanz von der Sicherheitsüberwachungsfunktion generiert.

Das folgende Diagramm zeigt die Stoppzeit und die Stoppdistanz der Stoppkategorie Joint 0(Basisachse), Joint 1(Schulterachse), Joint 2(Ellbogenachse).

Vorsicht

Je nach Situation kann die tatsächliche Stoppbewegung von den untenstehenden Ergebnissen abweichen. Bei Joint 0 ist es das Messergebnis bei horizontaler Bewegung und bei Joint 1und 2 das Messergebnis bei vertikaler Abwärtsbewegung. Für die Armlänge wurde der maximale Abstand verwendet.

RB5-850E Serie Basis (Joint 0)		
	Stoppdistanz (mm)	Stopzeit (sec)
Test 1	183.38	0.19
Test 2	160.1	0.24
Test 3	191.03	0.24
Maximum	191.03	0.24
Durchschnitt	178.17	0.22
Condition	Max. Reach / Max. Velocity / Horizontal Motion	

RB5-850E Serie Schulter (Joint 1)		
	Stoppdistanz (mm)	Stopzeit (sec)
Test 1	183.71	0.13
Test 2	177.53	0.13
Test 3	183.39	0.21
Maximum	183.71	0.21
Durchschnitt	181.54	0.16
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

RB5-850E Serie Ellbogen (Joint 2)		
	Stoppdistanz (mm)	Stopzeit (sec)
Test 1	90.935	0.14
Test 2	94.809	0.14
Test 3	81.987	0.14
Maximum	94.809	0.14
Durchschnitt	89.24	0.14
Condition	Max. Reach / Max. Velocity / Vertical Downward Motion	

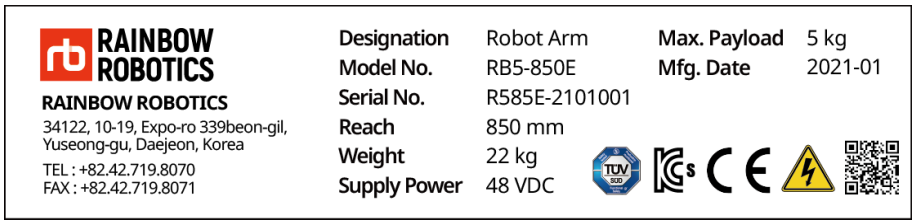
© RAINBOW ROBOTICS Inc. All rights reserved.

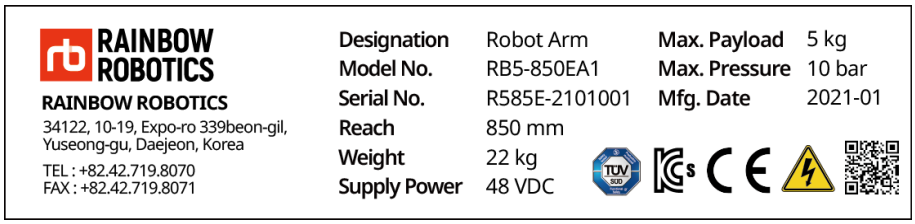
Anhang H Beschriftung

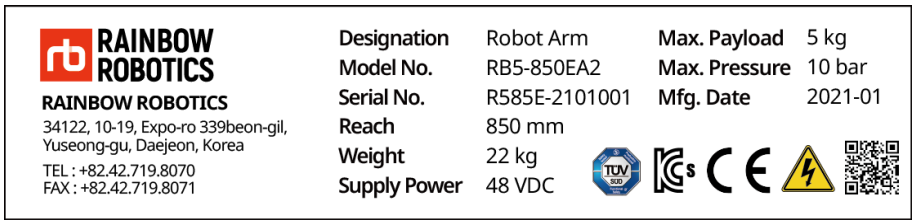
Die Beschriftung des Roboters ist separat für den Roboterarm und die Steuerbox wie folgt gekennzeichnet.

[Roboterarm]

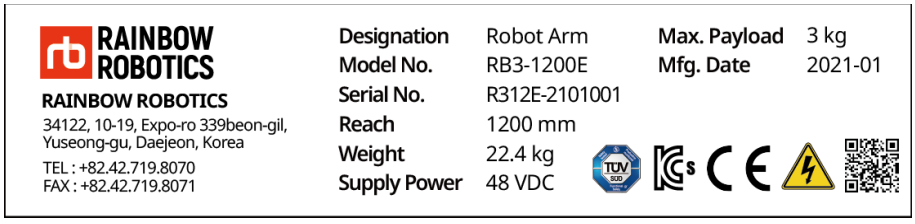
- RB5-850E Serie

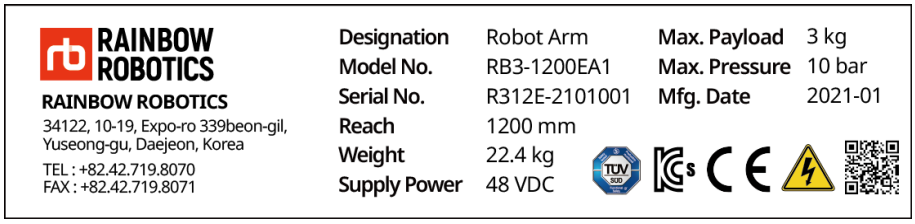






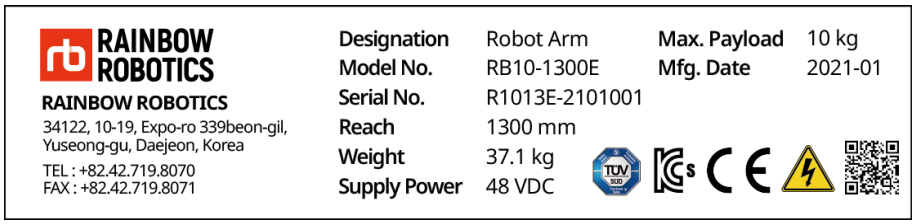
- RB3-1200E Serie

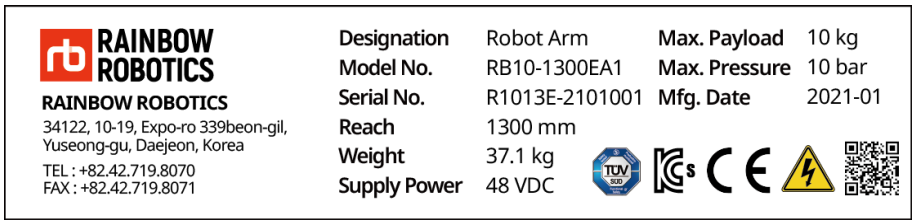


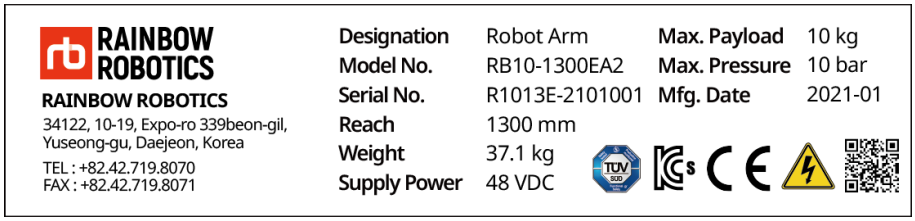


© RAINBOW ROBOTICS Inc. All rights reserved.

- RB10-1300E Serie

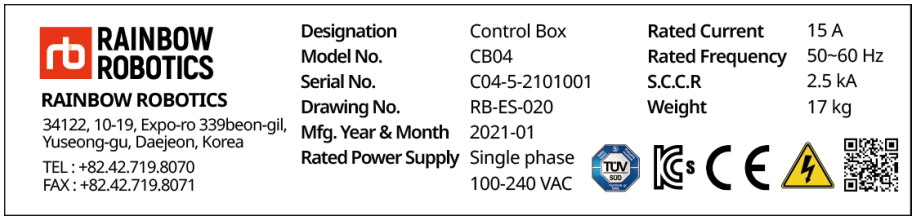




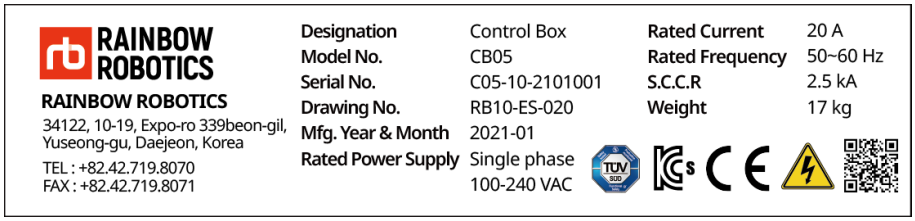


[Steuerbox]

- RB5-850E Serie, RB3-1200E Serie: Stand type(CB04)



- RB10-1300E Serie: Stand type(CB05)



© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang I Modbus-Server

- **Vorsicht**
  Diese Bedienungsanleitung behandelt den Modbus-Server (slave). Die Beschreibung der Modbus-Client-Funktion finden Sie in Kapitel 6.

1. Einleitung

Der Modbus TCP-Server(slave) der RB-Serie ist auf die Portnummer 502 festgelegt. Die IP-Adresse ändert sich entsprechend den Netzwerkeinstellungen über die Benutzeroberfläche (anfängliche IP-Adresse: 10.0.2.7).

Der Modbus-Server der RB-Serie ermöglicht die Verbindung mehrerer Clients und führt die folgenden Betriebsbefehle aus.

	Funktionscode	Funktionsname
Bit-Adresse	2	Read Discrete Inputs
	1	Read Coils
	5	Write Single Coil
	15	Write Multiple Coils
16-bit-(Word) Adresse	4	Read Input Registers
	3	Read Multiple Holding Registers
	6	Write Single Holding Register
	16	Write Multiple Holding Registers

2. Exception Code (Ausnahmecode)

Wenn auf eine falsche Adresse zugegriffen wird, der Wertebereich falsch ist oder ein falscher Befehl gesendet wird, wird die folgende Fehlermeldung zurückgegeben.

Exception Code	Exception Name
1	Illegal Function
2	Illegal Data Address
3	Illegal Value

© RAINBOW ROBOTICS Inc. All rights reserved.

3. Bit Address Map

Bit Address (Bit-Adresse)			
Addresse	Funktion	lesen	schreiben
0	Box digital input 0	o	x
1	Box digital input 1	o	x
2	Box digital input 2	o	x
3	Box digital input 3	o	x
4	Box digital input 4	o	x
5	Box digital input 5	o	x
6	Box digital input 6	o	x
7	Box digital input 7	o	x
8	Box digital input 8	o	x
9	Box digital input 9	o	x
10	Box digital input 10	o	x
11	Box digital input 11	o	x
12	Box digital input 12	o	x
13	Box digital input 13	o	x
14	Box digital input 14	o	x
15	Box digital input 15	o	x
16	Box digital output 0	o	o
17	Box digital output 1	o	o
18	Box digital output 2	o	o
19	Box digital output 3	o	o
20	Box digital output 4	o	o
21	Box digital output 5	o	o
22	Box digital output 6	o	o
23	Box digital output 7	o	o
24	Box digital output 8	o	o
25	Box digital output 9	o	o
26	Box digital output 10	o	o
27	Box digital output 11	o	o
28	Box digital output 12	o	o
29	Box digital output 13	o	o
30	Box digital output 14	o	o
31	Box digital output 15	o	o
32	Tool digital input 0	o	x
33	Tool digital input 1	o	x
34	Tool digital output 0	o	o
35	Tool digital output 1	o	o

© RAINBOW ROBOTICS Inc. All rights reserved.

4. Word(16 bit) Address Map (Adresszuordnung)

Word Address (Wortadresse)				
Addresse	Funktion	lesen	Schreiben	Bemerkungen
0	Box digital input 0~15	o	x	[BBBB BBBB BBBB BBBB]
1	Box digital output 0~15	o	o	[BBBB BBBB BBBB BBBB]
2	Box analog input 0	o	x	1mV unit
3	Box analog input 1	o	x	1mV unit
4	Box analog input 2	o	x	1mV unit
5	Box analog input 3	o	x	1mV unit
6	Box analog output 0	o	o	1mV unit
7	Box analog output 1	o	o	1mV unit
8	Box analog output 2	o	o	1mV unit
9	Box analog output 3	o	o	1mV unit
10	Extend digital input 0~15	o	x	[BBBB BBBB BBBB BBBB]
11	Extend digital output 0~15	o	x	[BBBB BBBB BBBB BBBB]
12	Extend analog input 0	o	x	1mV unit
13	Extend analog input 1	o	x	1mV unit
14	Extend analog input 2	o	x	1mV unit
15	Extend analog input 3	o	x	1mV unit
16	Extend analog output 0	o	o	1mV unit
17	Extend analog output 1	o	o	1mV unit
18	Extend analog output 2	o	o	1mV unit
19	Extend analog output 3	o	o	1mV unit
20~29	Reserved (Box IO)			
30	Tool output voltage	o	o	0, 12, 24
31	Tool digital input 0~1	o	x	[TTxx xxxx xxxx xxxx]
32	Tool digital output 0~1	o	o	[TTxx xxxx xxxx xxxx]
33	Tool analog input 0	o	x	1mV unit
34	Tool analog input 1	o	x	1mV unit
35~49	Reserved (Tool IO)			
50	Is Robot Activated	o	x	0 or 1
51	Is Real-mode	o	x	0 or 1
52	Is Collision Detected	o	x	0 or 1
53	Is Robot arm power engaged	o	x	0 or 1
54	Is Direct Teaching mode	o	x	0 or 1
55	Is Robot moving	o	x	0 or 1
56	Is Pause state	o	x	0 or 1
57	Is Teaching pendant is connected	o	x	0 or 1

© RAINBOW ROBOTICS Inc. All rights reserved.

58	Is Program Run	o	x	0 or 1
59	Is No-Arc mode is on	o	x	0 or 1
60	Is EMG button released	o	x	0 or 1
61	Is First Program Run	o	x	0 or 1
62~99	Reserved (Future System)			
100	Command: Start Program Once	o	o	Rising Edge is command
101	Command: Start Program Repeat	o	o	Rising Edge is command
102	Command: Pause Program	o	o	Rising Edge is command
103	Command: Stop Program	o	o	Rising Edge is command
104	Command: Resume from pause	o	o	Rising Edge is command
105	Command: Resume from collision	o	o	Rising Edge is command
106	Command: Load default Program	o	o	Rising Edge is command
107	Command: Robot Arm activation	o	o	Rising Edge is command
108	Command: Change to Real-mode	o	o	Rising Edge is command
109	Command: Power off the robot arm	o	o	Rising Edge is command
110~127	Reserved (Future System)			
128~255	User General Purpose Register	o	o	User Define Area
256	Joint reference 0	o	x	0.02deg unit / Signed
257	Joint reference 1	o	x	0.02deg unit / Signed
258	Joint reference 2	o	x	0.02deg unit / Signed
259	Joint reference 3	o	x	0.02deg unit / Signed
260	Joint reference 4	o	x	0.02deg unit / Signed
261	Joint reference 5	o	x	0.02deg unit / Signed
262	Joint angle 0	o	x	0.02deg unit / Signed
263	Joint angle 1	o	x	0.02deg unit / Signed
264	Joint angle 2	o	x	0.02deg unit / Signed
265	Joint angle 3	o	x	0.02deg unit / Signed
266	Joint angle 4	o	x	0.02deg unit / Signed
267	Joint angle 5	o	x	0.02deg unit / Signed
268	Joint current 0	o	x	10mA unit / Signed
269	Joint current 1	o	x	10mA unit / Signed
270	Joint current 2	o	x	10mA unit / Signed
271	Joint current 3	o	x	10mA unit / Signed
272	Joint current 4	o	x	10mA unit / Signed
273	Joint current 5	o	x	10mA unit / Signed
274	Joint information 0	o	x	

© RAINBOW ROBOTICS Inc. All rights reserved.

275	Joint information 1	o	x	
276	Joint information 2	o	x	
277	Joint information 3	o	x	
278	Joint information 4	o	x	
279	Joint information 5	o	x	
280	Joint temperature 0	o	x	celcius unit
281	Joint temperature 1	o	x	celcius unit
282	Joint temperature 2	o	x	celcius unit
283	Joint temperature 3	o	x	celcius unit
284	Joint temperature 4	o	x	celcius unit
285	Joint temperature 5	o	x	celcius unit
286	Joint 0 Estimated Current	o	x	10mA unit / Signed
287	Joint 1 Estimated Current	o	x	10mA unit / Signed
288	Joint 2 Estimated Current	o	x	10mA unit / Signed
289	Joint 3 Estimated Current	o	x	10mA unit / Signed
290	Joint 4 Estimated Current	o	x	10mA unit / Signed
291	Joint 5 Estimated Current	o	x	10mA unit / Signed
292	Joint 0 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
293	Joint 1 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
294	Joint 2 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
295	Joint 3 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
296	Joint 4 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
297	Joint 5 Gap(Esti.-Meas.) Current	o	x	10mA unit / Signed
298	Joint 0 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
299	Joint 1 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
300	Joint 2 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
301	Joint 3 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
302	Joint 4 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
303	Joint 5 Gap(Esti.-Meas.) Curr+LPF	o	x	10mA unit / Signed
304~329	Reserved (Joint Information)			
330	TCP reference X	o	x	0,1mm unit / Signed

© RAINBOW ROBOTICS Inc. All rights reserved.

331	TCP reference Y	o	x	0,1mm unit / Signed
332	TCP reference Z	o	x	0,1mm unit / Signed
333	TCP reference RX	o	x	0,02deg unit / Signed
334	TCP reference RY	o	x	0,02deg unit / Signed
335	TCP reference RZ	o	x	0,02deg unit / Signed
336	TCP position X	o	x	0,1mm unit / Signed
337	TCP position Y	o	x	0,1mm unit / Signed
338	TCP position Z	o	x	0,1mm unit / Signed
339	TCP position RX	o	x	0,02deg unit / Signed
340	TCP position RY	o	x	0,02deg unit / Signed
341	TCP position RZ	o	x	0,02deg unit / Signed
342~389	Reserved (TCP Information)			

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang J System Update (Systemaktualisierung)

- **Vorsicht**
  Es wird empfohlen, die Programmdateien (.wsl) in der Tablet-Benutzeroberfläche vor der Systemaktualisierung zu sichern.

1. Einleitung

Die Systemaktualisierung besteht aus zwei Hauptschritten.

UI-Aktualisierung durch APK-Installation → Systemsoftwareaktualisierung

2. Programmdatei-Backup

Verbinden Sie das Tablet und den individuellen/geschäftlichen PC, und rufen Sie die Programmdatei (.wsl) aus dem untenstehenden Pfad ab und sichern Sie sie.

Tablet → Android → data → com.rainbow.cobot → files → work→ Get .wsl-Dateien

(※ Es wird empfohlen, die erfassten Dateien zu sichern, bevor Sie mit dem nächsten Schritt fortfahren.)

3. Aktualisierung des UI-Programms

Tablet-UI/Hauptsystemprogramm wird in Format von APK vertireben.

Dies ist eine Installationsdatei in der Gleichen Format wie eine normale Android-Anwendung. Wenn daher Installations-APK-Datei auf das Tablet verschoben wird, wird das UI-Programm aktualisiert.

(※ Rainbow Robotics empfielt, die vorhandene Anwendung zu deinstallieren und erneuert die Anwendung zu installieren.)

(※ Da beim Löschen der bestehenden Anwendung auch die Programmdatei (.wsl) gelöscht wird, wird empfohlen, dieses Verfahren nach dem Sichern der Programmdatei in Schritt 1 durchführen.)

Die bereitgestellte APK-Datei auf das Tablet verschieben → APK installieren

© RAINBOW ROBOTICS Inc. All rights reserved.

4. Tablet an die Steuerbox anschließen

Verbinden Sie das Tablet mit der Steuerbox und greifen Sie auf das UI-Programm zu. Verbinden Sie nach dem Anschließen die Kommunikation zwischen dem Tablet und der Steuerbox.

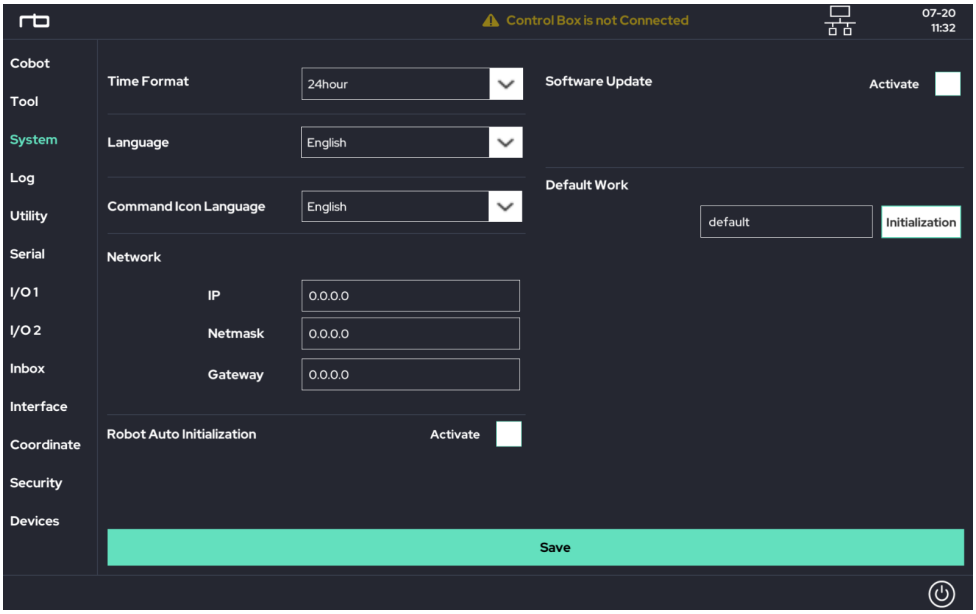
> UI-Startbildschirm → Make → unten links auf ‘State’ klicken → Connect

Wenn die Kommunikation zwischen dem Tablet und der Steuerbox hergestellt ist, leuchtet die erste Spalte blau. Aus Sicherheitsgründen wird empfohlen, den Roboter nicht zu initialisieren.

5. Zum Systemsoftware-Update gehen und aktivieren

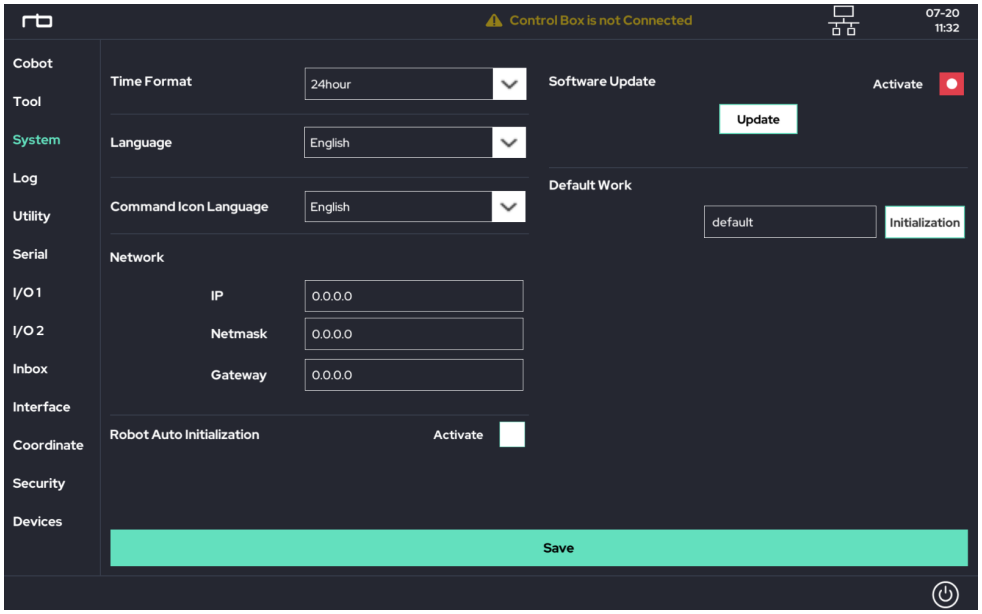
Navigieren Sie wie unten gezeigt zum Systemsoftwarepfad.

> UI-Startbildschirm → Make → Unten Page auswählen → Setup → zu System-Tab gehen



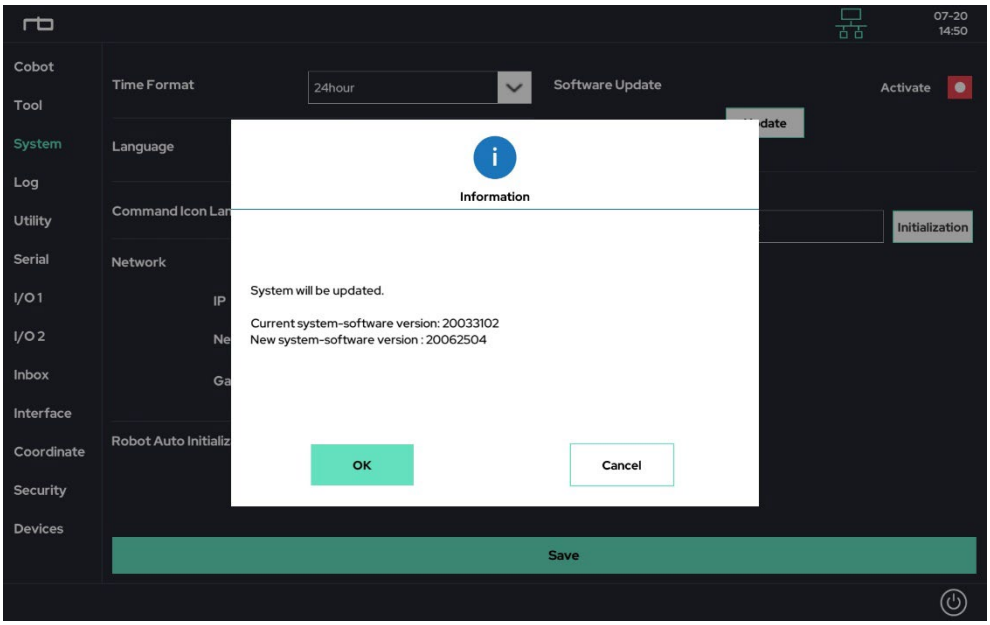
Klicken Sie rechts im Abschnitt “Software Update” auf das Kontrollkästchen ‘Activate(aktivieren)’.

© RAINBOW ROBOTICS Inc. All rights reserved.



6. System-Software aktualisieren

“Update”-Schaltfläche erscheint, und wenn Sie auf diese Schaltfläche klicken, erscheint noch ein Popup-Fenster.



Drücken Sie die “OK”-Schaltfläche und warten Sie, bis die Software aktualisiert wird.

© RAINBOW ROBOTICS Inc. All rights reserved.

Wenn das Update nach dem Klick auf die OK-Schaltfläche problemlos abgeschlossen ist, startet der PC (das Programm) der Steuerbox (Steuerung) innerhalb von 5 bis 15 Sekunden automatisch neu.

Während des Neustarts wird “Please Wait…” auf dem LCD-Bildschirm der Steuerbox vorübergehend angezeigt. Dadurch wird mitgeteilt, dass die Steuerbox neu gestartet wird.

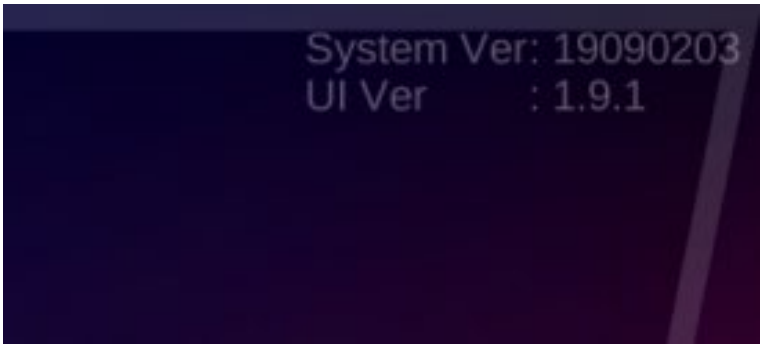
Wenn der Neustart endgültig abgeschlossen ist, wird “Normal Operation” auf dem LCD der Steuerbox angezeigt.

7. Nach Software-Update suchen

Verbinden Sie das UI-Tablet und die Steuerbox erneut.

UI-Startbildschirm → Make → unten links auf ‘State’ klicken → Connect

Wenn Sie zum Startbildschirm der UI zurückkehren, wird die Softwareversion in der oberen rechten (oder unteren linken) Ecke angezeigt. Überprüfen Sie, ob es auf die richtige Version aktualisiert wurde.

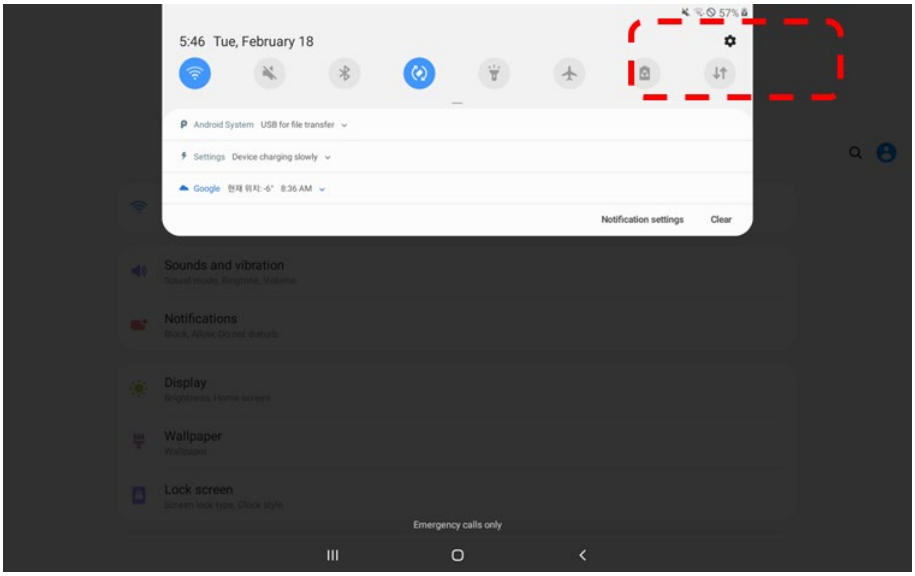


© RAINBOW ROBOTICS Inc. All rights reserved.

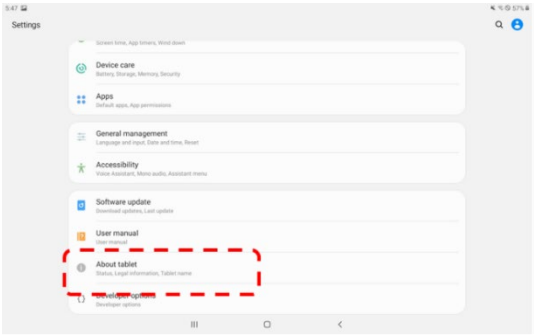
Anhang K Android-Tablet-Einstellungen

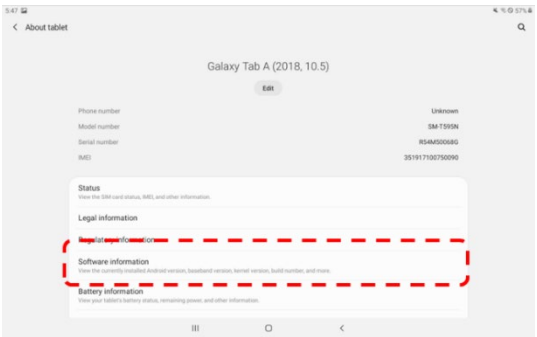
Vor der Verwendung des UI-Programms sind die folgenden Tablet-Einstellung erforderlich.

1. **Gehen Sie zu “Einstellungen” von Android.**



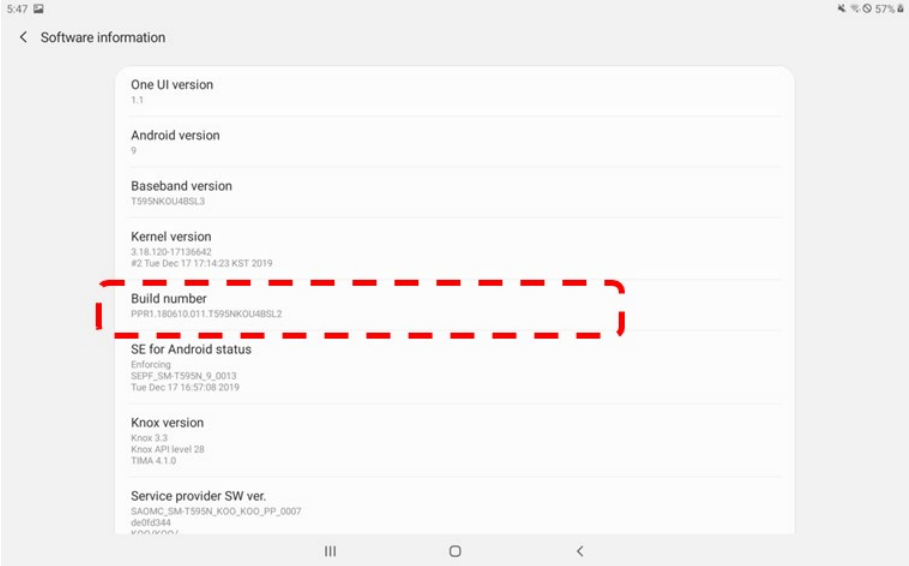
2. **Klicken Sie im Einstellungsmenü auf “Tablet Information” → “Software Information”.**



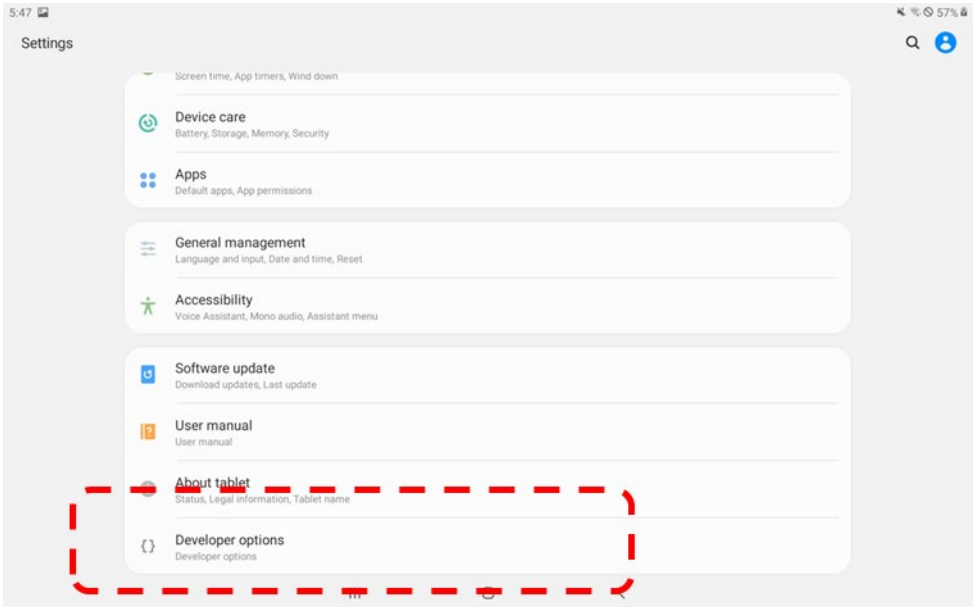


© RAINBOW ROBOTICS Inc. All rights reserved.

3. Klicken Sie mehrfach auf “Build Number” unter “About Tablet” (7 mal oder mehr).

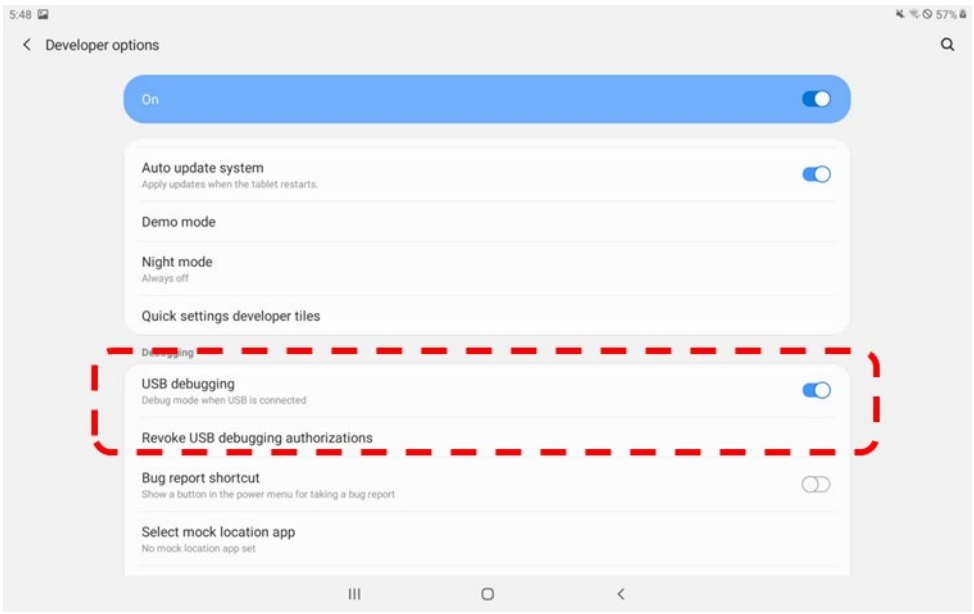


4. Ein Menü mit dem Namen “Developer Option” wird unter “About Tablet” angezeigt, wie unten gezeigt.



© RAINBOW ROBOTICS Inc. All rights reserved.

5. Aktivieren Sie “USB-Debugging” in den “Developer Option”.

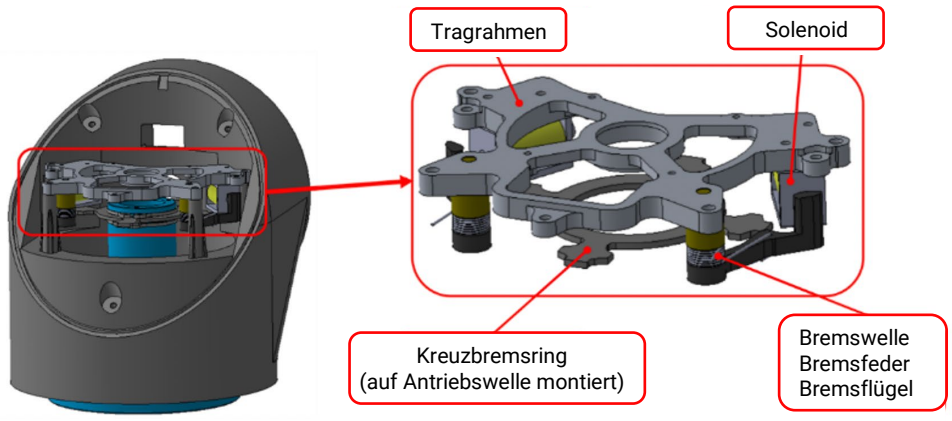


6. Installieren Sie das UI-Programm auf dem Tablet, indem Sie das von Rainbow Robotics vertriebene APK ausführen.

© RAINBOW ROBOTICS Inc. All rights reserved.

Anhang L Mechanische Bremssystem (Brake System)

Die an jeder Achse des Roboterarms installierte mechanische Stoppvorrichtung (Brake System) besteht aus einem Stützrahmen, einem Solenoid, einem Bremsring, einer Bremswelle, einer Bremsfeder und einem Bremsflügel und ist wie in der Abbildung unten gezeigt im Robotergelenk installiert.



[Robotergelenk] [Bremssystem]

Wenn das Solenoid eingeschaltet ist, wird die physische Interferenz zwischen dem Drehradius des Bremsrings und dem Bremsflügel aufgehoben. Wenn das Solenoid ausgeschaltet ist, tritt eine physische Interferenz zwischen dem distalen Ende des Bremsrings und dem Bremsflügel auf, um die Drehung der Antriebswelle zu stoppen.

Wenn sich der Bremsring dreht und den Bremsflügel drückt, kehrt der Flügel mit der Kraft der Feder zurück, und dann wird durch physikalische Interferenz eine bidirektionale Bremse erzeugt und beide Drehungen der Antriebswelle werden in einem gestoppten Zustand gehalten.

© RAINBOW ROBOTICS Inc. All rights reserved.

Rainbow Robotics Inc.

Hauptsitz und Fabrik

10-19, Expo-ro 339beon-gil, Yuseong-gu, Daejeon,
Republik Korea

Telefon	042. 719. 8070
Telefax	042. 719. 8071
Email	rainbow@rainbow-robotics.com
Vertriebsanfrage	sales@rainbow-robotics.com
Webseite	www.rainbow-robotics.com