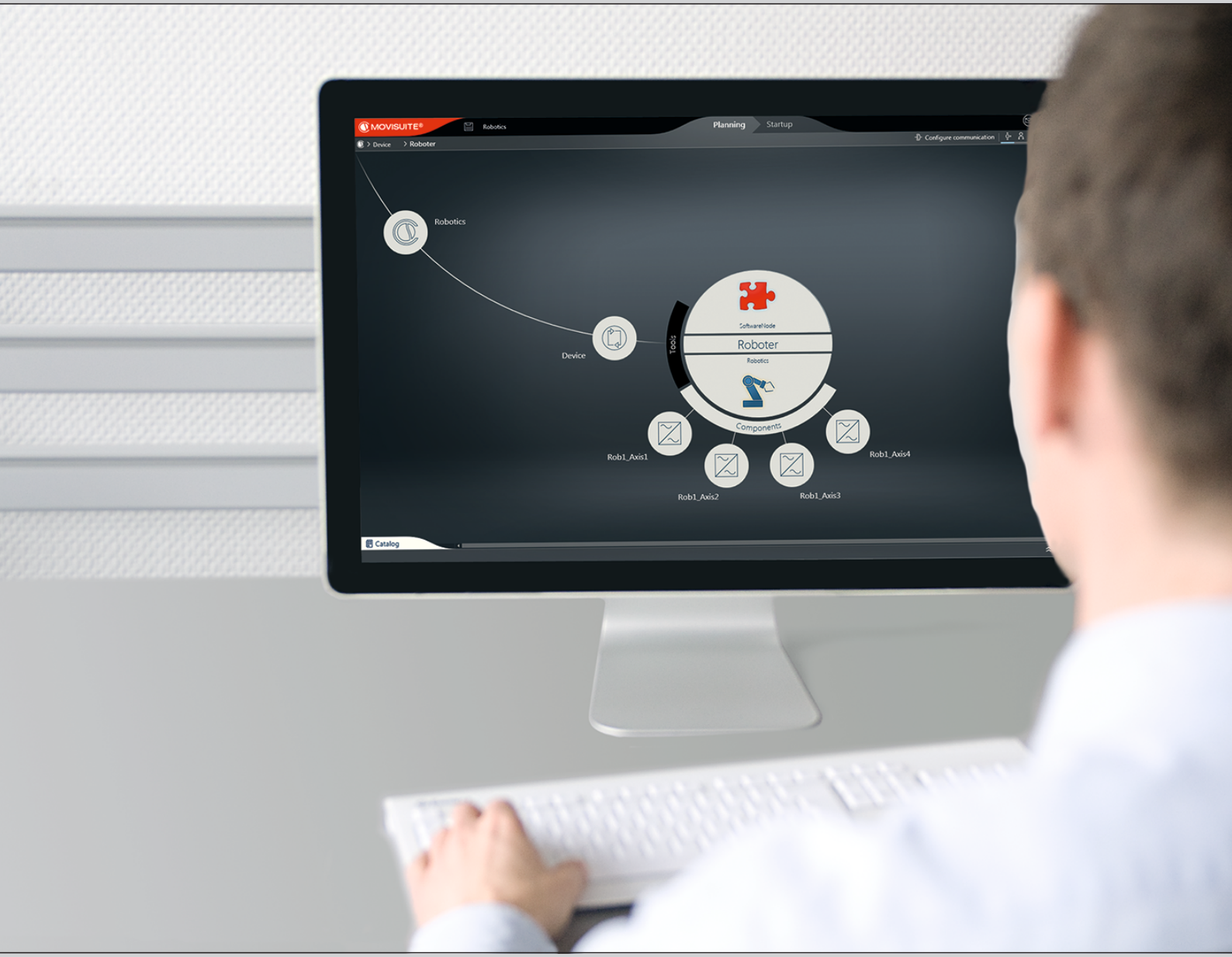
SEW EURODRIVE

Handbuch



MOVIKIT®
Robotics

Ausgabe 07/2021

26873346/DE

Inhaltsverzeichnis

26873346/DE – 07/2021

26873346/DE – 07/2021

26873346/DE – 07/2021

26873346/DE – 07/2021

26873346/DE – 07/2021

26873346/DE – 07/2021

26873346/DE – 07/2021

1 Allgemeine Hinweise

1.1 Gebrauch der Dokumentation

Diese Dokumentation ist Bestandteil des Produkts. Die Dokumentation wendet sich an alle Personen, die Arbeiten an dem Produkt ausführen.

Stellen Sie die Dokumentation in einem leserlichen Zustand zur Verfügung. Stellen Sie sicher, dass die Anlagen- und Betriebsverantwortlichen sowie Personen, die unter eigener Verantwortung mit dem Produkt arbeiten, die Dokumentation vollständig gelesen und verstanden haben. Bei Unklarheiten oder weiterem Informationsbedarf wenden Sie sich an SEW-EURODRIVE.

1.2 Inhalt der Dokumentation

Die Beschreibungen in dieser Dokumentation beziehen sich auf die aktuelle Version der Software zum Zeitpunkt der Publikation. Wenn Sie eine neuere Version der Software installieren, kann die Beschreibung abweichen.

Die aktuellste Ausgabe der Dokumentation finden Sie auch immer im Online-Support auf der Website von SEW-EURODRIVE.

1.3 Aufbau der Warnhinweise

1.3.1 Bedeutung der Signalworte

Die folgende Tabelle zeigt die Abstufung und Bedeutung der Signalworte der Warnhinweise.

Signalwort	Bedeutung	Folgen bei Missachtung
⚠ **GEFAHR**	Unmittelbar drohende Gefahr	Tod oder schwere Verletzungen
⚠ **WARNUNG**	Mögliche, gefährliche Situation	Tod oder schwere Verletzungen
⚠ **VORSICHT**	Mögliche, gefährliche Situation	Leichte Verletzungen
ACHTUNG	Mögliche Sachschäden	Beschädigung des Produkts oder seiner Umgebung
HINWEIS	Nützlicher Hinweis oder Tipp: Erleichtert die Handhabung mit dem Produkt.	

1.3.2 Aufbau der abschnittsbezogenen Warnhinweise

Die abschnittsbezogenen Warnhinweise gelten nicht nur für eine spezielle Handlung, sondern für mehrere Handlungen innerhalb eines Themas. Die verwendeten Gefahrensymbole weisen entweder auf eine allgemeine oder spezifische Gefahr hin.

Hier sehen Sie den formalen Aufbau eines abschnittsbezogenen Warnhinweises:

SIGNALWORT!

Art der Gefahr und ihre Quelle.

Mögliche Folge(n) der Missachtung.

- Maßnahme(n) zur Abwendung der Gefahr.

26873346/DE – 07/2021

Bedeutung der Gefahrensymbole

Die Gefahrensymbole, die in den Warnhinweisen stehen, haben folgende Bedeutung:

Gefahrensymbol	Bedeutung
	Allgemeine Gefahrenstelle

1.3.3 Aufbau der eingebetteten Warnhinweise

Die eingebetteten Warnhinweise sind direkt in die Handlungsanleitung vor dem gefährlichen Handlungsschritt integriert.

Hier sehen Sie den formalen Aufbau eines eingebetteten Warnhinweises:

⚠ **SIGNALWORT!** Art der Gefahr und ihre Quelle. Mögliche Folge(n) der Missachtung. Maßnahme(n) zur Abwendung der Gefahr.

1.4 Dezimaltrennzeichen bei Zahlenwerten

Diese Dokumentation verwendet den Punkt als Dezimaltrennzeichen.

Beispiel: 30.5 kg

1.5 Mängelhaftungsansprüche

Beachten Sie die Informationen in dieser Dokumentation. Dies ist die Voraussetzung für den störungsfreien Betrieb und die Erfüllung eventueller Mängelhaftungsansprüche. Lesen Sie zuerst die Dokumentation, bevor Sie mit dem Produkt arbeiten!

1.6 Produktnamen und Marken

Die in dieser Dokumentation genannten Produktnamen sind Marken oder eingetragene Marken der jeweiligen Titelhalter.

1.7 Urheberrechtsvermerk

26873346/DE – 07/2021

1.8 Mitgeltende Unterlagen

Für alle weiteren Komponenten gelten die dazugehörigen Dokumentationen.

Verwenden Sie immer die aktuelle Ausgabe der Dokumentationen und Software.

Auf der Webseite von SEW-EURODRIVE (www.sew-eurodrive.com) finden Sie eine große Auswahl an Dokumentationen in verschiedenen Sprachen zum Herunterladen. Bei Bedarf können Sie die Dokumentationen in gedruckter und gebundener Form bei SEW-EURODRIVE bestellen.

1.9 Kurzbezeichnung

In dieser Dokumentation gilt folgende Kurzbezeichnung:

Typenbezeichnung	Kurzbezeichnung
MOVIKIT® Robotics	Softwaremodul
Gesamtsystem aus Hardware- und Softwarekomponenten zur Ausführung der Bewegungsaufgaben	Roboter
SEW Robot Language	SRL
MOVISUITE® RobotMonitor	RobotMonitor

26873346/DE – 07/2021

2 Sicherheitshinweise

2.1 Vorbemerkungen

Die folgenden grundsätzlichen Sicherheitshinweise dienen dazu, Personen- und Sachschäden zu vermeiden und beziehen sich vorrangig auf den Einsatz der hier dokumentierten Produkte. Wenn Sie zusätzlich weitere Komponenten verwenden, beachten Sie auch deren Warn- und Sicherheitshinweise.

2.2 Zielgruppe

Fachkraft für Arbeiten mit Software

Alle Arbeiten mit der eingesetzten Software dürfen ausschließlich von einer Fachkraft mit geeigneter Ausbildung ausgeführt werden. Fachkraft im Sinne dieser Dokumentation sind Personen, die über folgende Qualifikationen verfügen:

- Geeignete Unterweisung
- Kenntnis dieser Dokumentation und der mitgeltenden Dokumentationen
- Für die Nutzung dieser Software empfiehlt SEW-EURODRIVE zusätzlich Schulungen zu den Produkten.

2.3 Netzwerksicherheit und Zugriffsschutz

Mit einem Bussystem ist es möglich, elektronische Antriebskomponenten in weiten Grenzen an die Anlagengegebenheiten anzupassen. Dadurch besteht die Gefahr, dass eine von außen nicht sichtbare Änderung der Parameter zu einem unerwarteten, aber nicht unkontrollierten Systemverhalten führen kann und die Betriebssicherheit, Systemverfügbarkeit oder Datensicherheit negativ beeinflusst.

Stellen Sie sicher, dass insbesondere bei Ethernet-basierenden vernetzten Systemen und Engineering-Schnittstellen kein unbefugter Zugriff erfolgen kann.

Die Verwendung von IT-spezifischen Sicherheitsstandards ergänzt den Zugriffsschutz auf die Ports. Eine Portübersicht finden Sie jeweils in den technischen Daten des verwendeten Geräts.

2.4 Bestimmungsgemäße Verwendung

MOVIKIT® Robotics ist ein Softwaremodul für MOVI-C® CONTROLLER zur Bahnsteuerung von Robotern sowie zum Tippen und Referenzieren von Roboterachsen.

Verwenden Sie die geräteübergreifende Engineering-Software MOVISUITE®, um die Achsen in Betrieb zu nehmen, zu konfigurieren und die fertige Konfiguration auf einen MOVI-C® CONTROLLER zu übertragen.

26873346/DE – 07/2021

3 Projektierungshinweise

3.1 Voraussetzung

Die richtige Projektierung und eine fehlerfreie Installation der Komponenten sind Voraussetzung für eine erfolgreiche Inbetriebnahme und für den Betrieb.

Ausführliche Projektierungshinweise finden Sie in der Dokumentation zu den betreffenden Komponenten.

3.2 Hardware

Folgende Hardware wird vorausgesetzt:

- MOVI-C® CONTROLLER (alle Leistungsklassen)
- MOVIDRIVE® modular, MOVIDRIVE® system **oder** MOVIDRIVE® technology (als interpolierendes Gerät)

HINWEIS

Für die Projektierung der Getriebemotoren und Umrichter des Roboters kann die Funktion "Physiksimulation" (→ 🗎 95) verwendet werden. In Verbindung mit der SEW Workbench lassen sich die Antriebe projektieren.

HINWEIS

Kontaktieren Sie zum Verwenden des Softwaremoduls in Verbindung mit Geräten wie MOVITRAC®, MOVIGEAR®, MOVIMOT® der Generation C den Service von SEW-EURODRIVE.

3.3 Software

Folgende Software wird vorausgesetzt:

- Engineering-Software MOVISUITE®

  (Enthält MOVIRUN® flexible, MOVIKIT® MultiMotion und den IEC-Editor)
- MOVISUITE® RobotMonitor

  Weitere Informationen zur Installation finden Sie im Kapitel "RobotMonitor installieren" (→ 🗎 107).

Beide Software-Komponenten sind im Online-Support von SEW-EURODRIVE als Download verfügbar.

Detailliertere Informationen bezüglich der Hardwarevoraussetzungen der einzelnen Softwarekomponenten können Sie der Dokumentation zur jeweiligen Software entnehmen.

26873346/DE – 07/2021

3.4 Zykluszeiten

Je nach dem welche Hard- und Software-Komponenten verwendet werden, resultieren die im Folgenden aufgelisteten einzustellenden Zykluszeiten. Diese Zykluszeiten müssen auf dem MOVI-C® CONTROLLER und/oder auf den Umrichtern eingestellt werden. Eine Anleitung zum Einstellen der Zykluszeit finden Sie im Kapitel "Zykluszeit einstellen" (→ 🗎 98).

Hinsichtlich der verwendeten Hardware gelten folgende Vorgaben:

- Beim Verwenden des MOVI-C® CONTROLLER UHX25A, die Zykluszeit auf dem MOVI-C® CONTROLLER und auf den Umrichtern **≥ 3 ms** einstellen. Es ist maximal eine Robotics-Instanz möglich.
- Beim Verwenden des MOVI-C® CONTROLLER UHX45A, die Zykluszeit auf dem MOVI-C® CONTROLLER **≥ 2 ms** einstellen. Es sind maximal 5 Robotics-Instanzen (bei höheren Zykluszeiten) möglich.
- Beim Verwenden des MOVI-C® CONTROLLER UHX65A-R01 die Zykluszeit auf dem MOVI-C® CONTROLLER **≥ 5 ms** einstellen. Es sind maximal 2 Robotics-Instanzen (bei höherer Zykluszeit) möglich.
- Beim Verwenden des MOVI-C® CONTROLLER UHX65A-R02/R04 oder UHX85A kann die Zykluszeit auf dem MOVI-C® CONTROLLER auf **1 ms** eingestellt werden. In Konstellationen ist jedoch eine Erhöhung der Zykluszeit notwendig.

  HINWEIS: Die Zykluszeit-Angaben sind nur Richtwerte bei typischen Pick-and-Place Anwendungen mit einem 3-achsigen Roboter. Bei mehreren Robotics-Instanzen oder zusätzlich eingesetzter Software sowie abhängig von der verwendeten Funktionalität muss die Zykluszeit höher eingestellt werden bzw. sind weniger Instanzen betreibbar als angegeben. Halten Sie gegebenenfalls Rücksprache mit SEW-EURODRIVE.

Hinsichtlich der verwendeten Software gelten folgende Vorgaben:

- Beim Verwenden folgender Software, die Zykluszeit auf dem MOVI-C® CONTROLLER und auf den Umrichtern auf **≤2 ms** einstellen:
  - MOVIKIT® MultiMotion addon PositionController
  - MOVIKIT® MultiMotion addon CombinedEncoderEvaluation
- Beim Verwenden des MOVIKIT® addon AntiSway, die Zykluszeit auf dem MOVI-C® CONTROLLER und auf den Umrichtern auf **1 ms** einstellen.

Zum Verwenden anderer Zykluszeiten kontaktieren Sie den Service von SEW-EURODRIVE.

26873346/DE – 07/2021

3.5 Lizenzierung

Folgende Lizenzen sind verfügbar bzw. werden vorausgesetzt:

- MOVIRUN® flexible

  Lizenz für die Softwareplattform MOVIRUN® flexible. Je MOVI-C® CONTROLLER einmal erforderlich.

- MOVIKIT® Robotics

  Lizenz für das Basismodul MOVIKIT® Robotics. Je Roboter einmal erforderlich.

- MOVIKIT® Robotics addon MediumModels

  Lizenz für das Add-on "MediumModels". Je Roboter mit 3 oder 4 Gelenkachsen einmal erforderlich.

- MOVIKIT® Robotics addon LargeModels

  Lizenz für das Add-on "LargeModels". Je Roboter mit 5 oder 6 Gelenkachsen oder mindestens 2 Orientierungsfreiheitsgraden einmal erforderlich. Beachten Sie zudem die Erläuterungen zur Lizenzierung des Add-Ons im Kapitel "LargeModels-Kinematikmodelle" (→ 🗎 47).

- MOVIKIT® Robotics addon Touchprobe

  Lizenz für das Add-on "Touchprobe". Je Roboter mit Touchprobe-Messung oder Touchprobe-Positionierung einmal erforderlich.

- MOVIKIT® Robotics addon ConveyorTracking

  Lizenz für das Add-on "ConveyorTracking". Je Roboter mit Interpolation in einem bewegten USER-Koordinatensystem einmal erforderlich.

- MOVIKIT® Robotics addon Circle (in Vorbereitung)

  Lizenz für das Add-on "Circle". Je Roboter mit Kreisinterpolation einmal erforderlich.

Weitere Informationen zur Lizenzierung erhalten Sie im Dokument "MOVI-C® Softwarekomponenten". Das Dokument ist über die Webseite von SEW-EURODRIVE (www.sew-eurodrive.com) abrufbar.

26873346/DE – 07/2021

4 Systembeschreibung

4.1 Modulbeschreibung

In der Automation ist die Handhabung eines Produkts ein wichtiger Schritt in der Prozesskette. Die Beladung und Entladung geht direkt in die Taktzeit der Maschine ein. Das Bewegungsprofil muss schnell und gleichzeitig schonend für Produkt und Mechanik sein.

Positioniert ein Antrieb ein Werkstück vorwärts oder rückwärts, dann bewegt er sich in einer Dimension. Es ist einfach möglich, die Positionierzeit und damit den Prozess zu optimieren. Bereits ab 2 Achsen bewegt sich das Werkstück zweidimensional im Raum. Es ist nur noch schwer zu überblicken, welches Bewegungsprofil das Beste ist, damit die Handhabung möglichst schnell und trotzdem noch prozess-sicher erfolgt. Die Positionen im Raum sind oft variabel und über unterschiedliche Bahnen erreichbar.

Typische Anwendungen sind Pick-and-Place, Palettieren und Sekundärverpackung. Ähnliche Anforderungen stellen Anwendungen wie Plotten, Leimauftrag oder die Veredelung von Produkten, wie z. B. das Dekorieren von Lebensmitteln. Diese Prozesse können statisch oder dynamisch ablaufen.

Für diese beiden Anwendungsfelder wurde das Modul MOVIKIT® Robotics entwickelt. Es bildet die ideale Lösungsplattform, Bahnbewegungen genauso einfach und optimiert umzusetzen wie das Verfahren von Einzelachsen.

Das Softwaremodul MOVIKIT® Robotics im Detail:

- Lauffähig mit der Softwareplattform MOVIRUN® flexible
- Große Auswahl an Standard-Kinematikmodellen (Sonderkinematiken sind auf Anfrage möglich)
- 3D-Simulation des Roboters zur Verkürzung der Inbetriebnahmezeit sowie als zusätzlicher Schutz vor fehlerhafter Konfiguration.
- Intuitive Konfiguration des Roboters durch adaptive 3D-Modelle
- Programmieren in der Programmierumgebung von SEW-EURODRIVE (auch direkt am Bedienterminal)
- Optimale Unterstützung von Handling-Aufgaben wie Pick-and-Place oder Conveyor-Tracking
- Bidirektionale Kopplung mit IEC-61131-Laufzeitsystem für Flexibilität in vielfältigen Anwendungen.

4.1.1 Vorteile

Die Vorteile des Softwaremoduls im Überblick:

- Vielfach bewährte, gekapselte Bahnsteuerung bei Robotern
- Schnelle und einfache Inbetriebnahme eines Roboters durch einen intuitiv bedienbaren Assistenten sowie übersichtliche Diagnose- und Monitorfunktionen
- Intuitiv verwendbare Robotersprache zum Konfigurieren des Roboters
- Reduzierung der Taktzeit, durch die taktsynchrone Bahnsteuerung mit individuell definierbarem Überschleifen und konturtreuer Umfahrung von Störkanten

26873346/DE – 07/2021

4.1.2 Funktionsweise

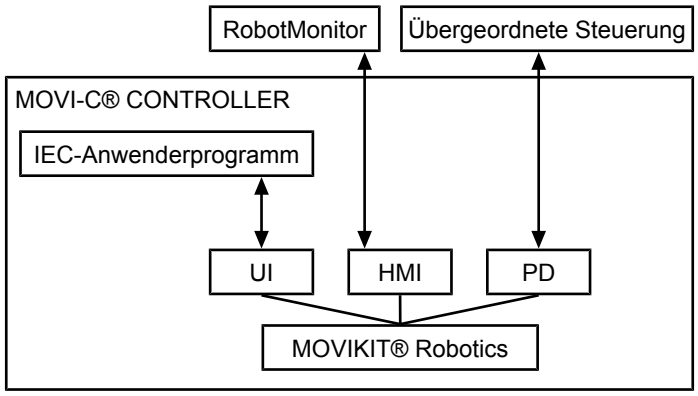
Prozess-Steuerung

Das Softwaremodul wird von einer Prozess-Steuerung angesteuert. Die Prozess-Steuerung gibt dem Softwaremodul durch das Starten und Parametrieren (optional) von SRL-Programmen vor, welche Bahnen mit welchen Bewegungsparametersätzen abgefahren werden.

Die Prozess-Steuerung kann wie folgt realisiert werden:

- Über die MOVISUITE® app RobotMonitor auf einem beliebigen Panel
- Über die Ansteuerung durch ein IEC-Anwenderprogramm
- Über die Ansteuerung durch eine übergeordnete Steuerung (SPS)

Ansteuerung



31654692235

RobotMonitor

Im RobotMonitor wird die Bewegungsbahn parametriert. Die Bahnpunkte und weitere Prozess-Signale können fest vorgegeben sein oder von der Prozess-Steuerung zur Laufzeit verändert werden.

Durch parametrierbare Schaltsignale wird gesteuert, wann der Roboter, welche Bewegung ausführt. Mit welcher Geschwindigkeit und Beschleunigung der Roboter diese Bahnsegmente abfährt, wird durch vorkonfigurierbare Bewegungsparametersätze definiert. Beispiele für Bewegungsparametersätze sind Eilgang, Schleichgang oder Greifbewegungen.

Steuerungsarten

Zusätzlich zum Programmbetrieb kann das Softwaremodul die Gelenkachsen, die kartesischen Achsen und die Einzelachsen im Tippbetrieb verfahren. Die untergeordneten Achsen bieten zudem die volle Funktionalität der jeweiligen Einzelachse. Beispielsweise können die Einzelachsen direkt über das Softwaremodul referenziert werden.

Fehlerbehandlung

Eine Fehlerbehandlung mit einer Fehlermeldungsübermittlung an die Prozess-Steuerung ist integriert. Für eine detailliertere Fehleranalyse und die Inbetriebnahme stehen mehrere Diagnose-Tools zur Verfügung, z. B. die im RobotMonitor integrierte 3D-Simulation.

26873346/DE – 07/2021

4.2 Funktionen

Die Funktionen im Überblick:

- Ansteuerung und Koordination der Antriebe einer Roboteranwendung
- Bewegungssteuerung mittels Bahninterpolation
- Unterstützung von Mechaniken mit standardmäßig bis zu 2 Gelenkachsen,
  mit den entsprechenden Add-ons bis zu 4 bzw. 6 Gelenkachsen
- Handhabung von ruhenden Objekten oder mit dem Add-on "ConveyorTracking" von bewegten Objekten.
- Betriebsarten: "Manuell" und "Automatik" (Programm)
- Tippbetrieb der Gelenkachsen, kartesischen Achsen und Einzelachsen
- Verschiedene Programmablaufarten im Programmbetrieb:
  Automatik, Einzelschritt für jeden Satz- oder Bewegungsbefehl
- Bedienung über den RobotMonitor oder den IEC-Editor
- Simulation von Abläufen zum Erkennen von Problemen ohne reale Maschine
- Reproduzierbare Bahntreue auch nach Störungen durch Rückpositionierung (BackToPath) auf die Bahn
- Konfiguration von Wartepunkten an beliebigen Stellen des SRL-Programms
- Geschwindigkeit mittels Override-Eingabewert skalieren
- Werkzeugtransformation
- Integrierte, automatisch aufgebaute 3D-Simulation des Roboters und seiner Bahnen im RobotMonitor
- SRL-Programmierung mit der SEW Robot Language (SRL):
  - Teach-In-Funktion
  - 20 Speicherplätze für Programme mit jeweils hunderten Bewegungsbefehlen
  - Linearinterpolation mit ruckbegrenztem Überschleifen
  - Verwenden von expliziten Koordinaten oder variable Posen
  - Alle Variablen (BOOL, REAL, POSE) in IEC schreib- und lesbar
  - Verwendung von Kontrollstrukturen: IF, WHILE
  - CallFunctions für synchronisierte und konsistente Ausführung von IEC-Code
  - Bahnereignisse weg- oder zeitbasiert oder kombiniert
- Ausführen des RobotMonitor auf einem mobilen Bedienpanel mit Zustimmtaster, Schlüsselschalter und Nothalt-Taster zur Erstellung und Anpassung des SRL-Programms sowie Diagnose direkt an der Maschine
- Kombinierbar mit dem MOVIKIT® MultiAxisController und dessen Add-ons (z. B. für Anwendungen, bei denen mehrere Antriebe eine Gelenkachse antreiben)
- Realitätsgetreue Physiksimulation der Kraft-Momenten-Belastungen auf die Antriebe von Robotern für ausgewählte Kinematikmodelle und Import der Belastungen in die SEW-Workbench, um dort eine Antriebsauswahl für den Roboter zu treffen (Antriebsprojektierung)

26873346/DE – 07/2021

4.3 Add-ons

HINWEIS

Das Aktivieren der Add-ons erfolgt im Konfigurationsmenü "Additional Functions" des Softwaremoduls (z. B. Touchprobe, Circle, ConveyorTracking) oder durch Verwendung der entsprechenden Funktionalität (z. B. MediumModels bzw. LargeModels durch Konfiguration eines Kinematikmodells mit 3/4 bzw. 5/6 Gelenkachsen). Nach dem Aktivieren ist in der Konfiguration ein zusätzliches Konfigurationsmenü sichtbar. Beachten Sie ggf. die zum Verwenden des Add-ons notwendige "Lizenz" (→ 🗎 16).

4.3.1 MOVIKIT® Robotics addon MediumModels

Unterstützung von Kinematikmodellen mit 3 oder 4 Gelenkachsen der Typen Portalroboter, Rollenportale, Deltaroboter, Tripoden, Quadropoden, SCARA und MIXED.

Das Add-on beinhaltet jeweils die 3D-Modelle der Kinematikmodelle im RobotMonitor und ermöglicht die komfortable Konfiguration der Modelle in MOVISUITE®.

Weitere Informationen finden Sie im Kapitel "Kinematikmodelle" (→ 🗎 44).

4.3.2 MOVIKIT® Robotics addon LargeModels

Unterstützung von Kinematikmodellen mit 5 oder 6 Gelenkachsen oder mindestens 2 Orientierungsfreiheitsgraden.

Weitere Informationen finden Sie im Kapitel "Kinematikmodelle" (→ 🗎 44).

4.3.3 MOVIKIT® Robotics addon Touchprobe

Erweiterung des Funktionsumfangs um die Möglichkeit der mehrdimensionalen Touchprobe-Funktionalität.

Beim Schalten eines Sensors oder dem Zustandswechsel einer BOOL-Variable im SRL-Programm wird die kartesische Istposition des Roboters auf der Bahn des Roboters ermittelt. Daraufhin kann eine definierte Aktion ausgeführt werden.

Eine mögliche auszuführende Aktion ist z. B. eine Restwegpositionierung. Dabei wird ausgehend vom gemessenen Bahnpunkt auf der programmierten Bahn des Roboters ein bestimmter Restweg verfahren. Der Restweg wird hierbei in eine bestimmte Richtung angegeben, beispielsweise beim Palettieren entlang der Z-Koordinate.

Weitere Informationen finden Sie in den Kapiteln "Grundlagen" (→ 🗎 29), "Funktionsbeschreibung" (→ 🗎 86) und "SRL-Programmierung" (→ 🗎 183).

4.3.4 MOVIKIT® Robotics addon ConveyorTracking

Erweiterung des Funktionsumfangs um die Möglichkeit der Interpolation in bewegten Koordinatensystemen. Anwendungsfälle sind z. B. die Entnahme von Teilen von einem Förderband und Ablage in statischer Umgebung oder das direkte Umsetzen von Waren zwischen mehreren Förderbändern.

Weitere Informationen finden Sie in den Kapiteln "Synchronisierte Bewegung mit einem Transportband" (→ 🗎 224) und "Synchronisierte Bewegung mit einem Drehtisch" (→ 🗎 227).

26873346/DE – 07/2021

4.3.5 MOVIKIT® Robotics addon Circle (in Vorbereitung)

Erweiterung des Funktionsumfangs um die Möglichkeit der Kreisinterpolation. Die Parametrierung des Kreissegments kann durch folgende Angaben erfolgen:

- Kreismittelpunkt und Winkel
- Kreismittelpunkt und Endpunkt des Kreissegments
- Zwischenpunkt auf dem Kreissegment und Endpunkt des Kreissegments
- Radius und Winkel
- Radius und Endpunkt des Kreissegments

26873346/DE – 07/2021

5 Grundlagen

5.1 Kinematikmodelle

Die Kinematikmodelle unterscheiden sich in den folgenden Punkten:

- Art und Anordnung der Gelenke: Grundtyp (z. B. SCARA)
- Art und Anzahl der Gelenkachsen: J1 – J6, linear/rotativ (z. B. LRRR)
- Detailunterschiede: Anordnung der Antriebe (z. B. M10)

Die Bezeichner der Kinematikmodelle sind entsprechend aufgebaut.

5.1.1 Grundtypen

CARTESIAN GANTRY

CARTESIAN GANTRY ist ein Kinematikmodell für einen Portalroboter, bei dem 2 oder 3 Linearachsen senkrecht zueinander stehen und damit einen kartesischen Arbeitsraum aufspannen.

ROLLER GANTRY

ROLLER GANTRY ist ein Kinematikmodell für ein Rollenportal, bei dem 2 translatorische Freiheitsgrade von 2 in der Regel stationären Antrieben über einen umlaufenden Zahnriemen gesteuert werden. Dieser Baugruppe können weitere Freiheitsgrade vor- und/oder nachgelagert sein.

SCARA

SCARA ist die englischsprachige Abkürzung für "Selective Compliance Assembly Robot Arm". SCARA ist eine kinematische Kette, bei der 2 rotatorische Achsen zueinander parallel angeordnet sind. Diese Achsen werden Schulter- und Ellbogengelenk in Analogie zum menschlichen Arm genannt.

DELTA

DELTA ist ein Kinematikmodell, bei dem 2 kinematische Teilketten zwischen Kinematikbasis und Werkzeugflansch in einer Dreiecksanordnung parallel verbunden sind.

TRIPOD

TRIPOD ist ein Kinematikmodell, das als Dreibein charakterisierbar ist und aus 3 parallel angeordneten kinematischen Teilketten zwischen Kinematikbasis und Werkzeugflansch besteht.

QUADROPOD

QUADROPOD ist ein Kinematikmodell, das aus 4 parallel angeordneten kinematischen Teilketten zwischen Kinematikbasis und Werkzeugflansch besteht.

HEXAPOD

HEXAPOD ist ein Kinematikmodell, das aus 6 parallel angeordneten kinematischen Teilketten zwischen Kinematikbasis und Werkzeugflansch besteht.

ARTICULATED

ARTICULATED ist das Kinematikmodell eines Knickarmroboters. Ein Knickarmroboter weist 5 bis 6 Freiheitsgrade auf und umfasst mehrere Drehgelenke.

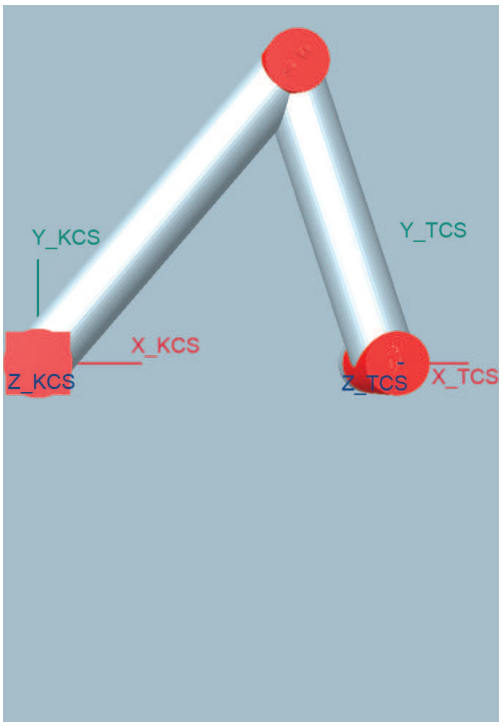
26873346/DE – 07/2021

MIXED

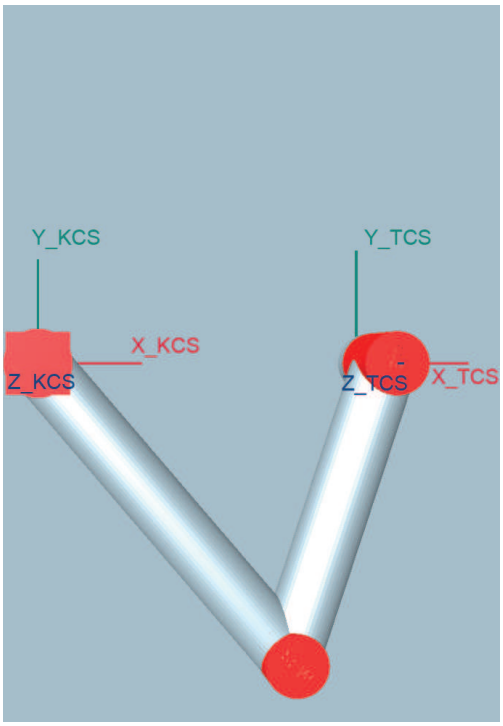
Als MIXED werden solche Kinematikmodelle bezeichnet, die nicht eindeutig die Charakteristiken anderer Kinematikmodelle aufweisen. Insbesondere entsprechen diese Kinematikmodelle nicht den Kinematikmodellen CARTESIAN CANTRY, ROLLER GANTRY, SCARA, DELTA, TRIPOD, QUADROPOD, HEXAPOD und ARTICULATED.

5.1.2 Konstellation

Bei bestimmten Kinematiken reicht die kartesische Pose (ISO 8373: "Kombination von Position und Orientierung im Raum") zur eindeutigen Beschreibung der Achsstellung nicht aus. Als einfaches Beispiel kann hierfür die SCARA-Kinematik genannt werden, bei der die gleiche kartesische Pose mit 2 unterschiedlichen Achsstellungen erreicht werden kann (siehe folgende Abbildungen). In welcher Stellung der Roboter bei der Pose steht, wird durch die Konstellation beschrieben. Die Konstellation ist eine Numerierung der möglichen Achsstellungen.



13935221643



13935224075

HINWEIS

Die Konstellation wird beim Verfahren durch die Bahninterpolation und beim Tippen der kartesischen Achsen beibehalten.

Durch das Tippen der Einzelachsen, der Gelenkachsen oder PTP-Interpolation kann die Konstellation geändert werden.

5.1.3 Transformation zwischen Achsen und Gelenkachsen

Bei jedem Kinematikmodell besteht die Möglichkeit, eine Transformation zwischen den Einzelachsen (durch Antrieb bewegte Linear- oder Drehachse) und den angetriebenen Gelenkachsen der Kinematik zu konfigurieren.

So lassen sich z. B. mechanische Vorgelege (Umwandlung Linearbewegung in Drehbewegung oder Drehbewegung einer Wickelachse in Linearbewegung) und Kopplungen zwischen den Achsen (Umleitung einer Drehbewegung durch ein Parallelogramm oder einen Riemen oder die Kopplungen in einem integrierten Hub-Dreh-Modul oder den Handachsen eines typischen Knickarmroboters) abbilden.

5.2 Bewegungssteuerung

5.2.1 Interpolation

Die Roboter-Bewegungssteuerung generiert für alle Betriebszustände der Kinematik einen Verlauf der Motor-Sollpositionen. Die Bewegungsbahn wird durch eine Bahninterpolation erzeugt. Das Ziel wird in kartesischen Koordinaten vorgegeben.

Bahninterpolation

Die Bahninterpolation erzeugt eine geometrisch definierte Bahn der translatorischen und der rotatorischen Freiheitsgrade.

Das Werkzeug des Roboters verfährt entlang einer im Raum definierten Bahn aus Geradenabschnitten oder Kreisbögen und Überschleifbögen. Bei den Übergängen zwischen 2 Geradenabschnitten und/oder Kreisbögen kann ein Abrunden der Bahn (das Überschleifen) definiert werden.

PTP-Interpolation

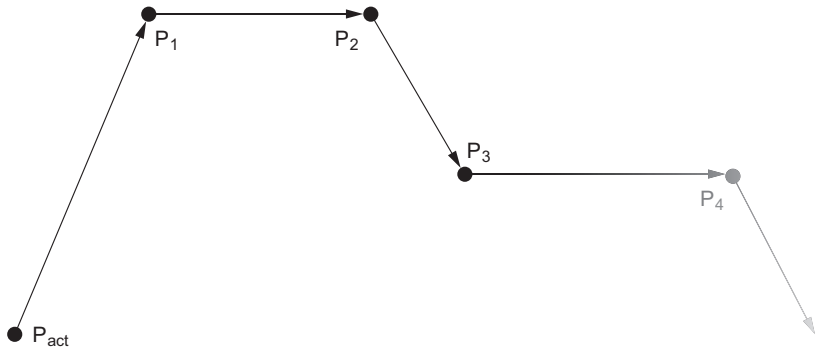
Bei der PTP-Interpolation werden die Gelenkachsen synchronisiert zu den Gelenkachswerten in der Zielposition verfahren. Anwendungsfälle für PTP-Interpolation sind:

- Wechseln der Konstellation im Programmbetrieb
- Bewegen in oder nahe Singularitäten
- Ausrichten der Gelenkachsen bei Verwendung eines Kinematikmodells mit mehr Gelenkachsen als kartesische Freiheitsgrade oder ohne inverse Kinematiktransformation, bevor kartesisch interpoliert wird (z. B. JOG_CART, LIN, CIRC)
- Anfahren einer eindeutig definierten Position zu Beginn eines SRL-Programms (Konstellation und Gelenkachsenphasen)
- Bewegen mit maximal zulässiger Geschwindigkeit der Gelenkachsen

26873346/DE – 07/2021

5.2.2 Bahn

Das SRL-Programm gibt dem Roboter die Bahn des Werkzeugs vor. Die Bahn wird durch eine Folge von Bahnpunkten aufgespannt. Eine Bahn, bei der bei jedem Bahnpunkt angehalten wird, kann mit 4 Bahnpunkten beispielsweise wie folgt aussehen.



13929520267

P_{act} Bahnpunkt aktuell
P_{1-4} Bahnpunkte 1 – 4

Der Bewegungsabschnitt von einem Bahnpunkt zum Nächsten wird als Bahnsegment bezeichnet.

HINWEIS

Ob genau auf der Geraden zwischen den einzelnen Bahnpunkten verfahren wird, hängt von der Interpolationsart, den Wartepunkten und dem Überschleifen ab.

26873346/DE – 07/2021

5.2.3 Überschleifen

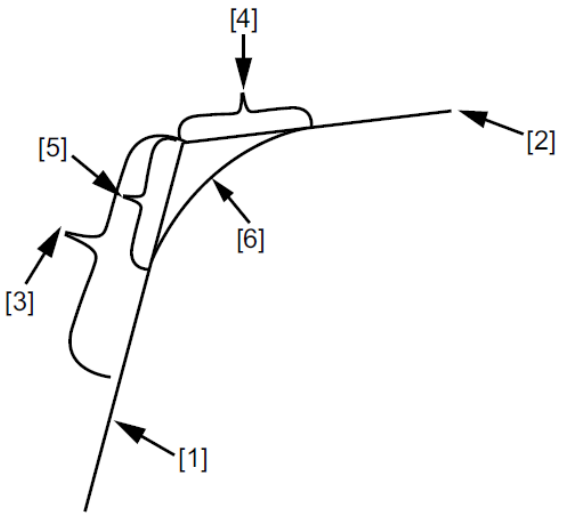
Das Überschleifen (Blending) ist das Verrunden der Ecken einer Bewegungsbahn. Das Überschleifen sorgt für einen stetigen Übergang der Bahn sowie der Bahngeschwindigkeit zum nächsten Bahnsegment. Überschleifen schont die Mechanik und reduziert die Taktzeit. Ohne Überschleifen hält die Kinematik an der *Zielpose* an und startet anschließend die Bewegung zum nächsten Zielpunkt.

Das Überschleifen wird gestartet, sobald sich das Werkzeug nahe genug an der aktuellen Zielpose befindet. Der Abstand (Überschleifdistanz), ab dem auf das neue Bahnsegment übergeschliffen wird, gibt das SRL-Programm für jedes Bahnsegment vor. Jedoch wird diese Überschleifdistanz auf einen Prozentsatz der Länge des Segments begrenzt, auf das übergeschliffen wird. Der Standardwert für diesen Begrenzungsprozentsatz sind 50 %. Der Wert kann jedoch auch auf bis zu 99 % erhöht werden.

Somit ergibt sich die tatsächliche Überschleifdistanz *(Überschleifdistanz$_{effektiv}$)* aus dem kleineren Wert (Minimum) der beiden Größen, begrenzt durch die verbleibende Reststrecke, wenn das neue Bahnsegment (z. B. wegen eines Wartepunkts) erst spät übernommen wird.

$$\textit{Überschleifdistanz}_{effektiv} =$$

$$\min(\textit{Überschleifdistanz}_{Soll}, \textit{Segmentlänge} \cdot \textit{Begrenzungsprozentsatz}, \text{Reststrecke})$$



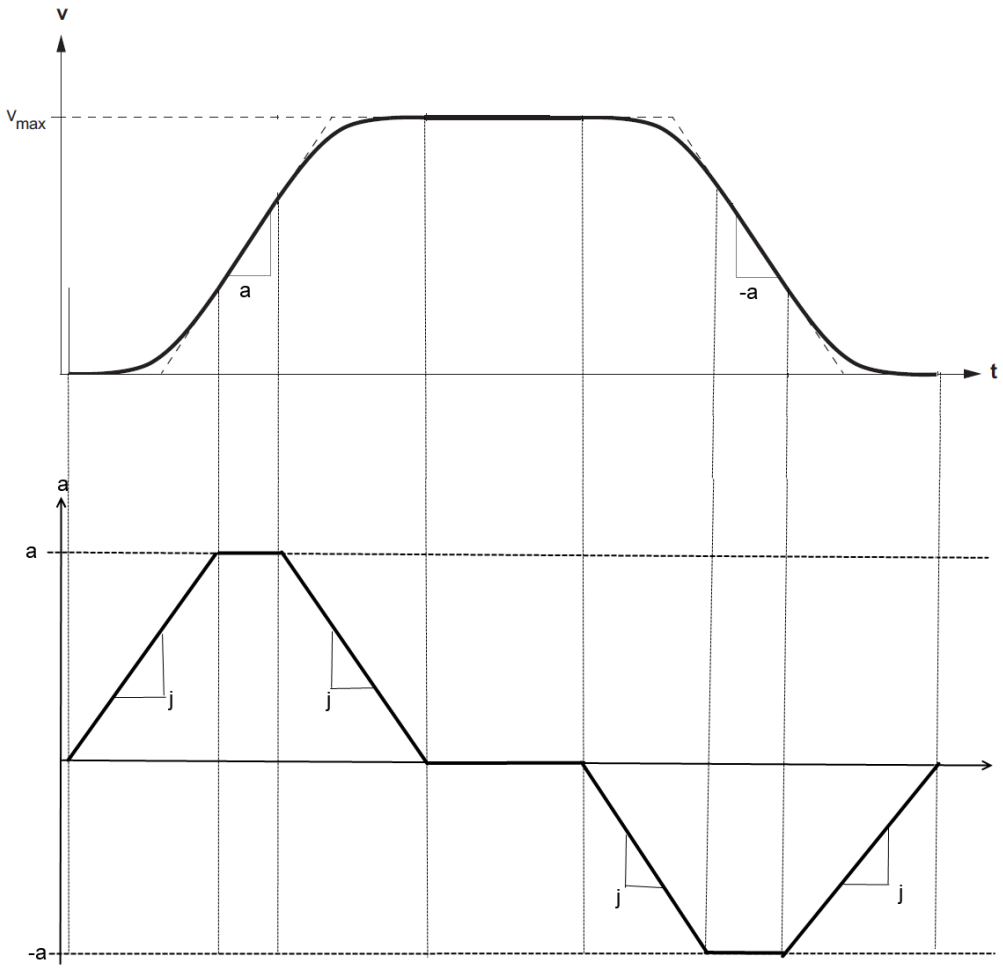
31458163211

[1] Linearsegment zur ersten Zielposition
[2] Zielposition des zweiten Linearsegments
[3] Eingestellte Überschleifdistanz$_{SOLL}$
[4] Segmentlänge * Begrenzungsprozentsatz
[5] Überschleifdistanz$_{effektiv}$
[6] Resultierender symmetrischer Überschleifbogen

26873346/DE – 07/2021

5.2.4 Bewegungsprofile

Für Bewegungen entlang einer Achse oder um eine Achse herum (Gelenkachse, kartesische Achse oder Bahnachse) muss ein Bewegungsprofil definiert werden. Die folgende Abbildung zeigt, mit welchen Größen das Bewegungsprofil beim Verwenden dieses Softwaremoduls definiert wird.



9007222991719691

v_{max} Maximal erlaubte Geschwindigkeit für den Bewegungsauftrag
a Maximal erlaubte Beschleunigung für den Bewegungsauftrag
j Ruck (Änderungsrate der Beschleunigung)

In den folgenden Fällen wird v_{max} nicht erreicht (keine Konstantfahrt):

- *a* sehr gering im Verhältnis zu v_{max}.
- j ist sehr gering im Verhältnis zu *a*.
- Der zurückzulegende Weg ist sehr kurz.

In diesen Fällen vereinfacht sich das oben dargestellte Fahrdiagramm zu einem ruckbegrenzten Dreiecksprofil. Dementsprechend besteht auch die Möglichkeit, dass die Beschleunigung oder Bremsverzögerung nicht erreicht wird (kein Abschnitt mit konstanter Beschleunigung).

26873346/DE – 07/2021

HINWEIS

- Die Ruckbegrenzung erhöht bei vorgegebenem Weg die dafür benötigte Zeit.
- Neben diesen Grundlagen gibt es weitere Faktoren und Randbedingungen, die das reale Fahrdiagramm beeinflussen können. Das oben dargestellte Fahrdiagramm dient zur prinzipiellen Erklärung der relevanten Bewegungsparameter.

5.2.5 Bewegungsparametersätze

Für das Bewegen des Roboters ist es notwendig, Bewegungsparametersätze (z. B. Vorgaben für Eilgang und für Schleichgang) zu konfigurieren. Dabei müssen für jede Achse die kartesischen Freiheitsgrade oder für die Bahnsegmente bestimmte Bewegungsparameter definiert werden (Geschwindigkeit, Beschleunigung, Bremsverzögerung, Ruck).

Beispiel für das Konfigurieren der Bewegungsparameter:

Wenn eine Geschwindigkeit von 1 m/s gefahren und diese in 0.2 s erreicht werden soll, muss eine Beschleunigung von 5 m/s^2 eingestellt werden (a=v/t; Annahme: ohne Ruckbegrenzung). Wenn wiederum eine Beschleunigung von 5 m/s^2 in 0.1 s erreicht werden soll, muss ein Ruck von 50 m/s^2 eingestellt werden. Im ruckbegrenzten Fall dieses Beispiels wird die gewünschte Geschwindigkeit dann in weniger als 0.4 s erreicht (0.2 s + 2 * 0.1 s = 0.4 s).

Zum Einstellen der Parameter siehe "Bewegungsparameter einstellen" (→ 🗎 235).

5.2.6 Skalierung mittels Override

Zusätzlich zu den Bewegungsparametern in den Bewegungsparametersätzen wirkt der Eingabewert *Override*. Mit dem *Override* kann die programmierte Geschwindigkeit prozentual zwischen 0 % und 100 % skaliert werden. Der *Override* wirkt nur auf die Geschwindigkeit, nicht auf Beschleunigung, Bremsverzögerung und Ruck. Die benötigte Zeit für das komplette SRL-Programm wird damit also nicht proportional skaliert.

5.2.7 Standardeinheiten

Folgende Einheiten werden standardmäßig verwenden:

Größe	Einheit
Streckenmaß (Translation)	mm
Translationsgeschwindigkeit	mm/s
Translationsbeschleunigung	mm/s^2
Translationsruck	mm/s^3
Winkelmaß (Rotation)	Grad
Rotationsgeschwindigkeit	Grad/s
Rotationsbeschleunigung	$Grad/s^2$
Rotationsruck	$Grad/s^3$
Überschleifdistanz	mm

26873346/DE – 07/2021

5.2.8 Not-Halt

Der Not-Halt bremst die Kinematik mit den für die aktuelle Betriebsart eingestellten Rampen in den Stillstand ab. Ein Not-Halt erfolgt in folgenden Situationen:

- Fehlerzustand eines Softwaremoduls
- Wegnahme des Freigabesignals: In diesem Fall wird nach Erreichen des Stillstands die Freigabe der Einzelachse automatisch weggenommen und somit der Halt durch die Bremsen oder durch die Umrichterregler aktiviert.

Der Not-Halt wird nicht verwendet, wenn der Umrichter einen Fehler meldet oder die Reglersperre gesetzt wird. In diesem Fall bremst jeder Umrichter seinen Antrieb mit der im Umrichter konfigurierten Not-Halt-Rampe einzeln ab.

5.2.9 Bahnereignisse

Bahnereignisse sind Ereignisse, die an einer definierten Stelle auf der Bahn des Roboters oder eine definierte Zeit vor bzw. nach Erreichen dieser Stelle auslösen. Beispielsweise kann ein Leimauftrag an bestimmten Stellen gestartet und auch wieder gestoppt werden oder kann das Vakuum eines Sauggreifers eine bestimmte Zeit vor Erreichen des Bahnendes eingeschaltet werden. Die Bahnereignisse lösen unabhängig von der Abarbeitung des SRL-Programms, also dem Fortschritt des Programmzeigers, an der parametrierten Stelle bzw. die parametrierte Zeit davor oder danach aus.

Weitere Informationen finden Sie in den Kapiteln "Funktionsbeschreibung" (→ 🗎 80) und "SRL-Programmierung" (→ 🗎 181).

5.2.10 Touchprobe

Die Funktion "Touchprobe" ermöglicht das Auslösen einer Anweisung während eines Programms durch das Schalten eines Sensors oder den Zustandswechsel einer BOOL-Variablen. Beim Auslösen des Ereignisses wird die kartesische Istposition des Roboters auf der Bahn bestimmt. Als Anweisung kann die gemessene Position in eine POSE-Variable gespeichert, der Zustand einer BOOL-Variable verändert, eine Funktion aufgerufen oder eine Restwegpositionierung durchgeführt werden. Bei der Restwegpositionierung wird ausgehend vom gemessenen Bahnpunkt die programmierte Bahn um eine bestimmte Länge in die vorgegebene Richtung fortgesetzt. Ist der Restweg größer als die programmierte Bahn, wird das letzte Segment verlängert. Der Sensor kann direkt am MOVI-C® CONTROLLER angeschlossen werden (z. B. über einen digitalen Eingang) und den Zustandswechsel einer BOOL-Variablen bewirken. In diesem Fall werden die Istpositionen der angeschlossenen Antriebe für die Messung verwendet. Bei einer sehr hohen Genauigkeitsanforderung wird der Sensor an allen zum Roboter gehörenden Umrichtern angeschlossen. In diesem Fall werden die Touchprobe-Positionen der Umrichter für die Messung des Bahnpunkts verwendet. Die gemessenen Positionen der Antriebe werden in die kartesische Position transformiert.

Weitere Informationen finden Sie in den Kapiteln "Funktionsbeschreibung" (→ 🗎 86) und "SRL-Programmierung" (→ 🗎 183).

26873346/DE – 07/2021

5.3 Koordinatensysteme

Die Pose des Roboters lässt sich über verschiedene Koordinatensysteme angeben. Dabei sind abhängig von der Lage des Koordinatensystems die Koordinaten einer Pose in jedem Koordinatensystem unterschiedlich. Beim MOVIKIT® Robotics gibt es die folgenden Koordinatensysteme:

- Base

  Koordinatensystem, das im Allgemeinen körperfest im Sockel der Kinematik positioniert ist. Es dient als Bezugskoordinatensystem für die direkte kinematische Transformation (Transformation der Gelenkachswerte in kartesische Werte des Werkzeugs).

- Joint

  Koordinatensystem, in dem jede Koordinate einer Gelenkachsen des Kinematikmodells entspricht.

- User

  Koordinatensystem, das sich in der aktuellen Version der Software auf Base bezieht. Es kann statisch oder bewegt sein.

5.4 Kommunikation und Prozessdatenaustausch

In der Hardware-Topologie des Softwaremoduls gibt es mindestens die folgenden Systemkomponenten, die miteinander kommunizieren:

- MOVI-C® CONTROLLER mit dem Softwaremodul
- Applikationsumrichter

Der MOVI-C® CONTROLLER mit dem Softwaremodul kommuniziert mit den Applikationsumrichtern über den EtherCAT®-basierenden Systembus. Um eine eindeutige Adressierung der Nachrichten und somit der Bewegungsaufträge für die einzelnen Antriebe zu gewährleisten, muss jeder Applikationsumrichtern am Systembus eine andere Adresse haben.

Darüber hinaus kann es optional eine dem MOVI-C® CONTROLLER topologisch übergeordnete Speicherprogrammierbare Steuerung (SPS) geben. Mit dieser SPS kommuniziert der MOVI-C® CONTROLLER über den Feldbus. Hierfür werden Prozessdaten ausgetauscht. Beide Systeme müssen wissen, wie die Daten zu interpretieren sind.

26873346/DE – 07/2021

6 Funktionsbeschreibung

6.1 Roboterzustände

6.1.1 Betriebsart

⚠ GEFAHR

Unsichere Bewegung des Roboters durch nicht sicherheitsbewertet Funktionen der Robotersteuerung (MOVI-C® CONTROLLER mit MOVIKIT® Robotics)

Tod, schwere Verletzungen oder Sachschaden

- Führen Sie vor dem Verwenden eines Roboters eine Sicherheitsbeurteilung für den Roboter bzw. die Maschine durch.

Wenn in der Maschine eine Robotersicherheitssteuerung existiert, wird die Betriebsart explizit über einen an die Robotersicherheitssteuerung angeschlossenen Betriebsarten-Wahlschalter vorgegeben. Siehe Kapitel "Externen Anwahlschalter für die Betriebsarten verwenden" (→ 🗎 232). Wenn keine Robotersicherheitssteuerung existiert, kann die Prozess-Steuerung die Betriebsart explizit vorgeben.

Die folgende Tabelle zeigt eine Übersicht der Betriebsarten:

Unterscheidungs-merkmal	**Betriebsart**	
	Manuell mit hoher Geschwindigkeit (Manual High Speed)	**Automatik (Programm)**
Tippbetrieb möglich?	Ja	Nein
Verhalten im Programmbetrieb	Tipp-Verhalten: Der Start-Knopf des Programmbetriebs muss gehalten werden, damit der Programmbetrieb und das Abfahren der Bahn fortgesetzt werden (Bahntippen).	Start-Stopp-Verhalten: Der Start-Knopf des Programmbetriebs muss nur einmalig gedrückt werden, um die automatische Programmabarbeitung zu starten.

26873346/DE – 07/2021

6.1.2 Freigabeart

HINWEIS

Die Freigabeart kann nicht explizit angewählt werden, sondern ergibt sich implizit aus den Rahmenbedingungen, die im Kapitel "Roboterzustände" beschrieben sind.

HINWEIS

Die Antriebe werden beim achsweisen Not-Halt individuell über den Not-Halt im Umrichter gebremst. Es erfolgt kein Abbremsen über einen zentralen Profilgenerator. Dadurch kann es beim untergeordneten MOVIKIT® MultiAxisController in der Betriebsart "Priorität Drehmoment" zum Verspannen der Mechanik und in der Betriebsart "Priorität Schrägstellung" zu unerwarteter Schrägstellung kommen.

Der Zustand "Freigabe" ist Voraussetzung dafür, dass der Roboter im Tippbetrieb und im Programmbetrieb eine Bewegung ausführt. Es gibt folgende Freigabearten:

0: Not-Halt (achsweise), 1: Not-Halt (bahntreu), 2: Applikationshalt, 3: Freigabe

Diese Reihenfolge entspricht der Hierarchie der Freigabearten. Eine höherwertige Freigabeart wird aktiv, wenn keine Ursache für eine niedrigere Freigabeart gültig ist.

Folgende Tabelle zeigt die Unterschiede der verschiedenen Freigabearten:

	Applikationshalt	Not-Halt (bahntreu)	Not-Halt (achsweise)
Berechnung der Bremsung	MOVIKIT® Robotics	MOVIKIT® Robotics	MOVIKIT® des Achsgruppenteilnehmers
Bahntreue bei der Bremsung	Ja	Ja	Nein
Verwendeter Bewegungsparametersatz für die Bremsung	aktuell angewählter Bewegungsparametersatz	parametrierter Bewegungsparametersatz des Not-Halts (außer beim Tippen im Koordinatensystem "Axis". Hier wirkt die Bremsverzögerungsrampen des MOVIKIT® Achsgruppenteilnehmer)	Bremsverzögerungsrampen des MOVIKIT® Achsgruppenteilnehmer
Verhalten der Umrichter im Stillstand	Interpolierte Positionsregelung (mit Vorgabe einer konstanten Sollposition)	Not-Halt durch die Bremsen oder die Halteregelung des Umrichters	Not-Halt durch die Bremsen oder die Halteregelung des Umrichters
Fortsetzen der Bahn möglich nach Stillstand	Ja	Ja, Nachdem eine "Rückpositionierung (BackTo-Path)" (→ 🗎 39) durchgeführt wurde.	Ja, Nachdem eine "Rückpositionierung (BackTo-Path)" (→ 🗎 39) durchgeführt wurde.
Mögliche Ursachen der Ansteuerung	Tippbetrieb oder Programmbetrieb ist nicht aktiv.	keine Freigabe	Zugriff des Achsgruppenteilnehmers ist aktiviert und damit nicht auf "Upper".
Mögliche Ursachen des Softwaremoduls	-	Fehlerzustand	-

26873346/DE – 07/2021

	Applikationshalt	Not-Halt (bahntreu)	Not-Halt (achsweise)
Mögliche Ursachen der Achsgruppen-teilnehmer	-	-	• nicht verbunden • im Sicherheitshalt • nicht bereit • nicht referenziert (beim Tippen im Koordinaten-system "Joint" und "Base" oder im Pro-grammbetrieb) • Fehlerzustand
Mögliche Ursachen der Sicherheits-steuerung	fordert Applikationshalt	fordert Not-Halt	fordert Not-Halt (achsweise)

26873346/DE – 07/2021

6.1.3 Freigabezustand

Zur Diagnose des Softwaremoduls wird der Freigabezustand des Roboters beim Freigeben als Signal zurückgegeben.

(Schnittstelle im IEC-Editor: Basic.OUT.eEnableState)

Freigabezustände mit Zugriff auf die Achsgruppenteilnehmer

- Freigabezustände, bei denen der Sollwert der Roboterbewegungssteuerung nicht aktiv ist.
  - Achsen im Not-Halt (EmergencyStoppedAxes):

    Der Not-Halt ist bei allen Achsgruppenteilnehmern angefordert. Die Achsgruppenteilnehmer melden Stillstand.
  - Not-Halt - achsweise (EmergencyStoppingAxes):

    Der Not-Halt ist bei allen Achsgruppenteilnehmer angefordert. Die Achsgruppenteilnehmer bremsen.
  - Warte auf "Sollwert aktiv" (WaitingForSetpointActive):

    Die Interpolierte Lageregelung ist bei allen Achsgruppenteilnehmern angefordert. Es wird darauf gewartet, dass der Sollwert aktiv wird.
- Freigabezustände, bei denen die interpolierte Lageregelung bei allen Achsgruppenteilnehmern und der Sollwert der Roboterbewegungssteuerung aktiv ist.
  - Not-Halt auf der Bahn (EmergencyStoppingOnPath):

    Die Roboterbewegungssteuerung bremst gerade auf der Bahn mit den vorgegebenen Not-Halt-Rampen ab.
  - Positionshalteregelung (PositionHoldControl):

    Der Sollwert der Roboterbewegungssteuerung ist konstant (Stillstand). Der Applikationshalt des Roboters ist aktiv.
  - Applikationshalt auf der Bahn (ApplicationStoppingOnPath):

    Die Roboterbewegungssteuerung bremst gerade auf der Bahn mit den Applikationsrampen ab.
  - Warte auf Bewegungsbefehl (WaitingForMotionCommand):

    Die Roboterbewegungssteuerung wartet auf den nächsten Bewegungsbefehl des SRL-Programms.
  - Bahnbewegung (PathMotion):

    Die Roboterbewegungssteuerung führt gerade einen Bewegungsbefehl aus.

26873346/DE – 07/2021

6.1.4 Steuerungsart

Das Softwaremodul bietet folgende Steuerungsarten für den Roboter:

HINWEIS

Die Steuerungsart ist nicht explizit anwählbar, sondern wird implizit durch die Ansteuerung der Tipp- und der Programmsteuerungssignale gewählt.

Unterscheidungsmerkmal	Inaktiv	Tippbetrieb	Programmbetrieb	Referenzierbetrieb
Beschreibung	Tippbetrieb und Programmbetrieb sind nicht aktiv. Eine Bremsung wird durchgeführt.	Tippen des Roboters	Ausführen von SRL-Programmen	Referenzieren der Roboterachsen
Zielvorgabe durch	(nur Bremsung: siehe Freigabeart)	Tippsignale	SRL-Programm	Startbefehl zum Referenzieren einer Achse, parametrierte Referenzfahrt
Ursachen	• Tippbetrieb inaktiv • Programmbetrieb inaktiv	Mindestens ein Tippsignal ist aktiviert.	Programmstart ist aktiviert.	Mindestens ein Referenzier-Start einer Achse ist aktiviert
Zweck	Übergangszustand	• Erreichbarkeitsprüfung • Bewegung im Fehlerfall	Automatisiertes Bewegen des Roboters (Sollzustand)	Referenzieren der Roboterachsen
Maximal resultierende Freigabeart	Applikationshalt	Freigabe	Freigabe	Freigabe

26873346/DE – 07/2021

6.2 Tippbetrieb

⚠ VORSICHT

Unvorhergesehene Bewegung des Roboters beim Quittieren eines Fehlers im Tippbetrieb bei gesetztem Tippen-Eingangssignal.

Tod, schwere Verletzungen oder Sachschaden

- Stellen Sie vor dem Quittieren eines Fehlers im Tippbetrieb sicher, dass kein Tippsignal gesetzt ist.

⚠ VORSICHT

Wenn unreferenziert getippt wird (nur möglich im Koordinatensystem "Axis") können die angezeigten Positionen von den realen Positionen abweichen. Die eingestellten Software-Endschalter werden beim unreferenzierten Tippen nicht überprüft.

Tod, schwere Verletzungen oder Sachschaden

- Achten Sie beim unreferenzierten Tippen sehr genau auf das Verhalten der Achsen und stellen Sie sicher, dass diese nicht kollidieren. Verlassen Sie sich nicht auf die angezeigten Achspositionen.

Der Tippbetrieb ermöglicht einen einfachen Handbetrieb, um beispielsweise zu prüfen, ob eine bestimmte Pose erreicht werden kann.

Im Tippbetrieb kann der Roboter folgendermaßen verfahren werden:

- Tippbetrieb entlang jeder Gelenkachse (Jog Joints)
  - Tippen der Gelenkachsen in den Gelenkachskoordinaten
- Tippbetrieb entlang den kartesischen Raumachsen (Jog Cartesian)
  - Tippen des Roboterwerkzeugs in den kartesischen Translations- und Orientierungskoordinaten
- Tippbetrieb entlang jeder Einzelachse (Jog Axis)
  - Tippen der Einzelachsen in den Achskoordinaten. Das Tippen der Einzelachsen ist auch dann möglich, wenn diese nicht referenziert sind.

Die Bewegungsprofile für den Tippbetrieb werden durch den angewählten Bewegungsparametersatz festgelegt (Standardmäßig Bewegungsparametersatz 8). Für "Jog Axis" werden die Bewegungsprofile verwendet, die im selektierten Bewegungsparametersatz für die Gelenkachsen (Joints) festgelegt wurden. Da sich "Jog Axis" auf die Achskoordinaten bezieht, ist bei Kurbelanwendungen damit zu rechnen, dass sich die Gelenkachse durch "Jog Axis" bei identischem Bewegungsparametersatz deutlich schneller bewegt als beim Tippen der Gelenkachse ("Jog Joints"). In diesem Fall kann sich die Wahl eines anderen Bewegungsparametersatzes anbieten.

26873346/DE – 07/2021

6.3 Programmbetrieb

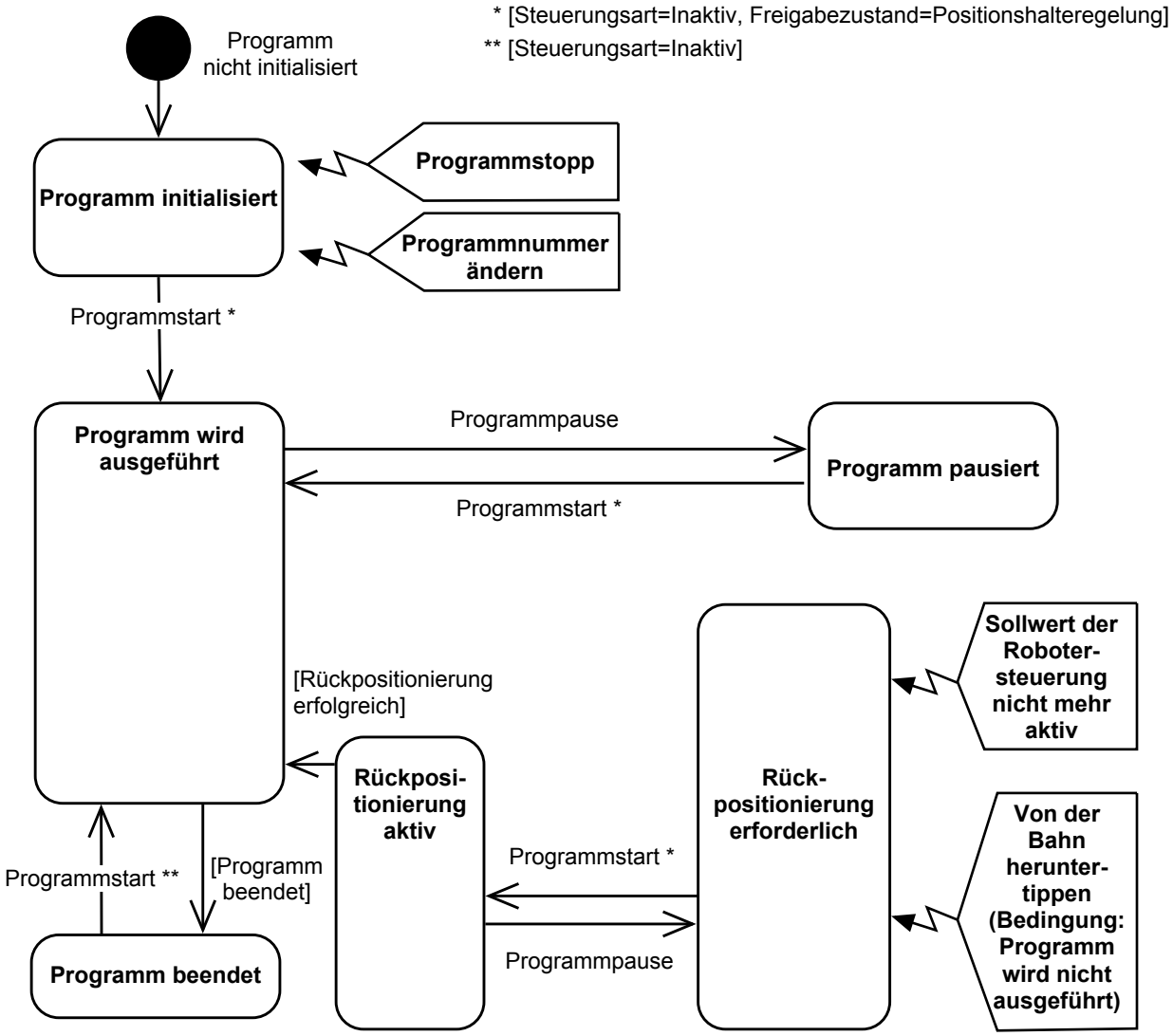
Der Programmbetrieb wird zum Abfahren von Bahnpunkten durch SRL-Programme verwendet. Zur Programmsteuerung gibt es die folgenden flankengesteuerten Signale:

Start, Pause, Stopp

Die Programmausführung kann sich in folgenden Zuständen befinden:

Programm nicht initialisiert, Programm initialisiert, Programm wird ausgeführt, Programm pausiert, Programm beendet, Rückpositionierung erforderlich, Rückpositionierung aktiv

Das folgende Diagramm visualisiert die Übergänge zwischen den Zuständen:



Weitere Informationen zum SRL-Programm finden Sie im Kapitel "SRL-Programmierung" (→ 🗎 137).

26873346/DE – 07/2021

HINWEIS

- Auch wenn das Programm beendet ist, kann eine Rückpositionierung erfolgreich sein z. B. wenn von der Bahn heruntergetippt wird oder der Sollwert nicht mehr aktiv ist. Wenn die Rückpositionierung nicht durchgeführt werden soll, kann dies mit Programmstopp unterdrückt werden.

- Ein Programmwechsel ist nur möglich, wenn das aktuelle Programm initialisiert oder beendet ist. Wird die Programmnummer geändert, während das aktuelle Programm schon ausgeführt aber noch nicht beendet ist, wird eine Fehlermeldung ausgegeben.

-Wenn ein Programm abgebrochen und ein anderes gestartet werden soll, muss das aktuell ausgeführte Programm durch einen Programmstopp gestoppt und initialisiert werden.

6.3.1 Programmablaufarten

Es gibt folgende Ablaufarten für die Programmabarbeitung:

- AUTO - Das Programm wird bis zum Ende ausgeführt.
- STEP_BLOCK - Eine Programmzeile wird ausgeführt.

6.3.2 Herstellen des Grundzustands

Beim Aufruf eines Programms wird durch die Software automatisch ein definierter Grundzustand hergestellt, d. h. bestimmte Parameter der Bewegungssteuerung werden gesetzt. Beim Aufruf von Unterprogrammen wird dieser Grundzustand nicht hergestellt.

Folgende Parameter werden bei jedem Programmstart gesetzt:

- Die Zielposition wird auf die aktuelle Position gesetzt. Damit ist es möglich unabhängig von vorangegangenen Programmen nur einzelne Koordinaten anzusteuern, ohne dass sich die anderen Koordinaten verändern.
- Das Überschleifen wird freigeschaltet.
- Die Überschleifdistanz wird auf 0 mm eingestellt.
- Das Basiskoordinatensystem wird angewählt.
- Die Zielvorgabe wird auf absolute Positionsvorgabe gesetzt (nicht auf relative Positionsvorgabe).
- Die Bewegungsart wird auf Linearinterpolation gesetzt.
- Die X-Y-Ebene wird als Ebene für die Kreisinterpolation gesetzt.
- Die Vorgabe für Zusatzpositionen bei der Kreisinterpolation wird auf relative Positionsvorgabe gesetzt.
- Die Parameter *Constellation* und *JointPhase[1..6]* von PTP-Befehlen werden auf 0 gesetzt.
- Die Parameter für Bahnereignisse werden auf Standardwerte gesetzt:

  Bezugspunkt = Segmentanfang, Abstand = 0 mm, Zeitversatz = 0 s
- Die Parameter für Touchprobe-Ereignisse und Restwegpositionierung werden auf Standardwerte gesetzt:

  Triggerquelle = Umrichter, Index der Trigger-Boolvariable = 1, Pegel = steigende Flanke, Modus = Single, Messrichtung = Z, Restweglänge = 0 mm, Touchprobe-Zähler wird auf "0" gesetzt

26873346/DE – 07/2021

6.4 Rückpositionierung (BackToPath)

Wenn die vorgegebene Bahn verlassen wurde, ist es erforderlich ein Rückpositionieren auf die Bahn durchzuführen, um die Bewegung anschließend fortsetzen zu können. Die Bahn wird verlassen, wenn der Sollwert der Robotersteuerung im Zustand "Programm wird ausgeführt" oder "Programm pausiert" nicht mehr aktiv ist. Dies passiert in den Freigabearten "Not-Halt (bahntreu)", Not-Halt (achsweise) und im Stillstand. Der Roboter meldet dann den Programmzustand "Rückpositionierung erforderlich". Wenn der Sollwert der Bewegungssteuerung anschließend wieder aktiv ist (Freigabezustand "Positionshalteregelung"), wird bei einer steigenden Flanke an Programmstart auf die Bahn zurückpositioniert.

Die Rückpositionierung kann mit Programmpause oder in der Betriebsart "Manuell mit hoher Geschwindigkeit" mit deaktiviertem Programmstart angehalten werden. Mit Programmstart kann sie wieder fortgeführt werden (Programmpause und Programmstopp dürfen nicht gesetzt sein).

Wenn sich die Programmausführung im Zustand "Programm wird ausgeführt" befindet, ist die Rückpositionierung abgeschlossen. Das Programm wird nach der Rückpositionierung automatisch fortgesetzt. Eine steigende Flanke an Programmstart ist nicht erforderlich.

Bei der Rückpositionierung werden alle Gelenkachsen synchronisiert auf die Bahn zurückpositioniert (Point-to-Point). Für die Rückpositionierung werden die in der Schnittstelle für die Rückpositionierung vorgegeben Werte im Register "Joint" beim "Einstellen der Bewegungsparametersätze" (→ 🗎 108) verwendet. Standardmäßig ist der Bewegungsparametersatz 8 eingestellt.

26873346/DE – 07/2021

6.5 Referenzierbetrieb

⚠ GEFAHR

Unvorhergesehene Bewegung des Roboters beim Quittieren eines Fehlers im Referenzierbetrieb bei gesetztem Referenzier-Eingangssignal (Start)

Tod, schwere Verletzungen oder Sachschaden

- Stellen Sie vor dem Quittieren eines Fehlers im Tippbetrieb sicher, dass kein Referenziersignal gesetzt ist.

Der Referenzierbetrieb wird zum Festlegen des Achsennullpunktes verwendet.

Das Referenzieren der Antriebe wird durch die Signale *Start Referenzieren* für die Achsen 1 - 6 angestoßen. Wenn der Umrichter referenziert ist, wird das zugehörige Signal *Achse 1 - 6 referenziert* auf "TRUE" gesetzt.

Wenn während einer Bewegung in den Referenzierbetrieb geschaltet wird (z. B. Bewegung, die durch ein Programm ausgelöst wurde), wird die Kinematik mit den in der Konfiguration definierten Notstopprampen abgebremst.

Wenn eines der Signale *Start Referenzieren* der Achsen 1 - 6 auf "TRUE" steht, während in den Referenzierbetrieb gewechselt wird, wird das Referenzieren für die entsprechende Achse angestoßen. In diesem Fall ist keine steigende Flanke des Signals Start Referenzieren erforderlich.

Das Referenzieren wird in den Antriebsfunktionen der jeweiligen Antriebsachse unter "FCB 12 Referenzfahrt" parametriert. Weitere Informationen dazu finden Sie in der Betriebsanleitung des entsprechenden Frequenzumrichters.

Weitere Informationen zum Referenzierbetrieb finden Sie in diesem Handbuch im Kapitel "Steuerung des Referenzierbetriebs" (→ 🗎 144).

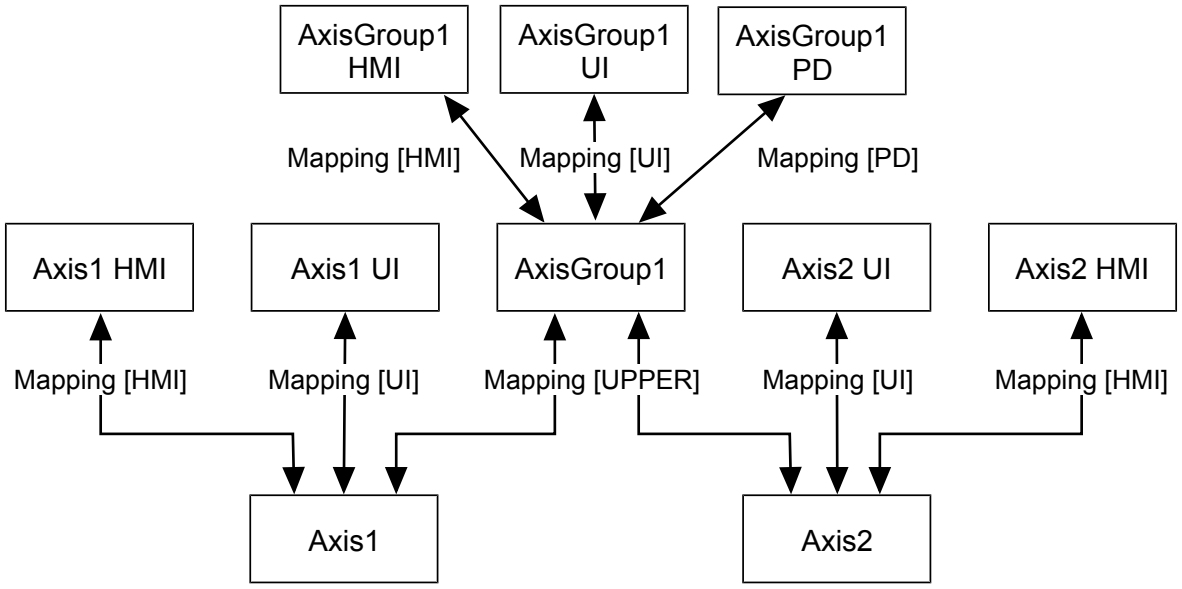
26873346/DE – 07/2021

6.6 Zugriffsverwaltung

Die Zugriffsverwaltung steuert, welche Steuerquelle auf die Anwenderschnittstelle des Softwaremoduls zugreifen darf.

Jede Steuerquelle kann den Zugriff anfordern und erhält eine entsprechende Rückmeldung, ob der Zugriff möglich ist.

Für das Softwaremodul gibt es die Steuerquellen HMI, UI, PD und UPPER. Diese Reihenfolge entspricht der Priorität der Steuerquellen für den Zugriff.



9007223774487435

Für die Einzelachsen gibt es eine zusätzliche Zugriffsverwaltung. Wenn für eine Einzelachse der Zugriff über UI oder HMI gewährt ist, hat das Softwaremodul darauf keinen Zugriff mehr.

26873346/DE – 07/2021

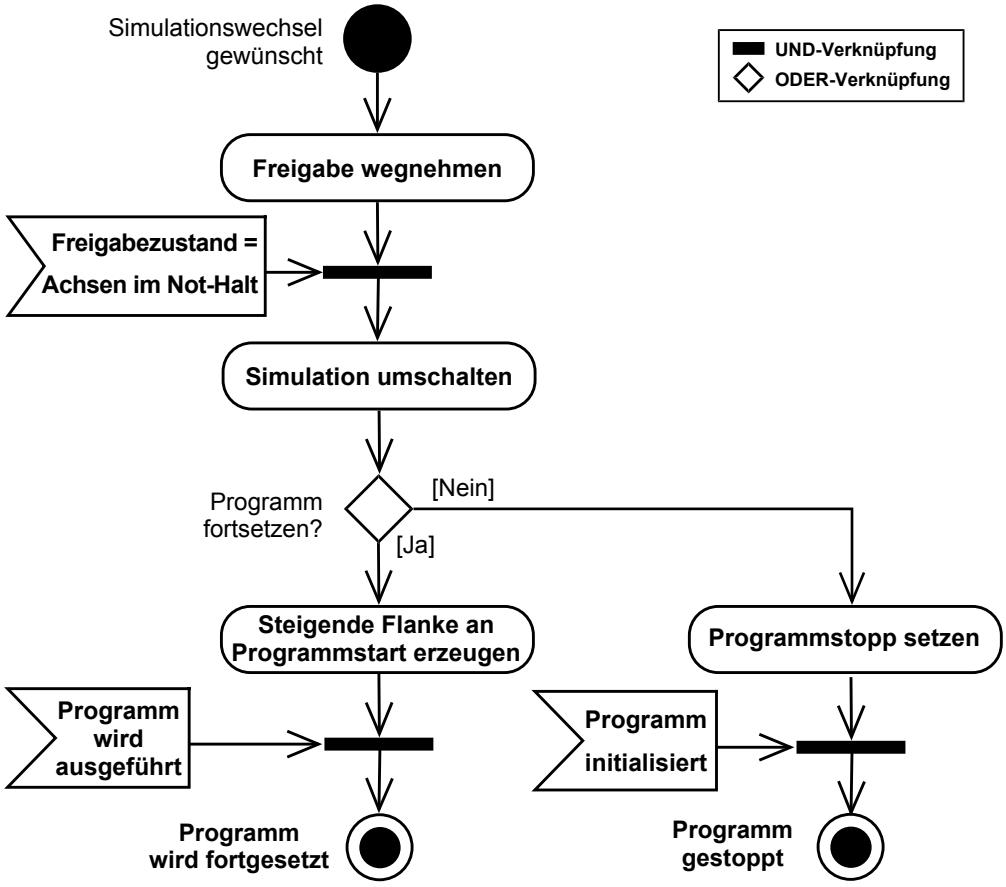
6.7 Simulation

Der Roboter kann simuliert werden, in dem das Eingangssignal *Simulation* gesetzt wird. Dadurch werden alle untergeordneten Achsgruppen bzw. Achsgruppenteilnehmer auf Simulation gestellt. Dadurch ist der Zustand der Umrichter egal und es wird stattdessen ein simulierter Antrieb verfahren.

Dies kann in den folgenden Fällen sinnvoll sein:

- Es sind keine Umrichter am Roboter angeschlossen, dennoch soll die Ansteuerung und Bewegung des Roboters getestet werden.
- Umrichter sind zwar angeschlossen, es wurde jedoch eine neue Bewegung programmiert, die vor der realen Anwendung simuliert getestet werden soll.

Für den zweiten Anwendungsfall ist die Umschaltung von realen auf simulierte Achsen und umgekehrt notwendig. Hierfür müssen sich die Achsen (egal ob simuliert oder real) in einem sicheren Zustand befinden. Dies ist der Fall, wenn die Freigabe am Roboter weggenommen ist und sich alle Umrichter im Not-Halt befinden und Stillstand melden. Folgendes Diagramm veranschaulicht die Umschaltung:



31362370315

Es besteht auch die Möglichkeit nur eine Teil der Achsen des Roboters zu simulieren. In diesem Fall muss die Simulation in MOVISUITE® bei den jeweiligen Achsen aktiviert werden. Für diese Achsen wird dann der Vorgabewert vom Roboter ignoriert. Die Achsen bleibenso lange simuliert, wie die Einstellung im MOVISUITE® aktiviert ist.

26873346/DE – 07/2021

6.8 Software-Endschalter

Einleitung

Im Softwaremodul MOVIKIT® Robotics werden mehrere Arten von Software-Endschaltern zur Überwachung der Werte folgender Größen unterschieden:

- Angetriebene Gelenkachsen
- Kartesische Position
- Schlepp-Gelenkachsen
- Schlepp-Abstände

Wird der durch die Software-Endschalter eingegrenzte Bereich überschritten, tritt ein Fehlerzustand ein. Dann ist nur noch das Verfahren im Tippbetrieb möglich, um den Roboter wieder in den zulässigen Bereich zu verfahren. Zudem wirken die Software-Endschalter der angetriebenen Gelenkachsen in der Betriebsart "JOG_JOINT" und die Software-Endschalter der kartesischen Positionen in der Betriebsart "JOG_CART" begrenzend, d. h. der zulässige Bereich kann in der jeweiligen Betriebsart nicht noch weiter verlassen werden.

Deaktivieren der Software-Endschalter

Die Software-Endschalter können deaktiviert werden (Disable SWLS), In diesem Fall wirken die Software-Endschalter im Tippbetrieb nicht mehr begrenzend und es kann auch im Programmbetrieb außerhalb des zulässigen Bereichs verfahren werden. Das Verlassen des zulässigen Bereichs wird jedoch weiterhin angezeigt. Wenn die Software-Endschalter deaktiviert sind, erscheint die Warnung "0x17E61" und ein Hinweis, die Software-Endschalter wieder zu aktivieren.

HINWEIS

In der aktuellen Version des Softwaremoduls wirkt das Deaktivieren der Software-Endschalter nur auf die im Softwaremodul selbst konfigurierten Software-Endschalter. Die Software-Endschalter unterlagerter MultiMotion-Achsen werden beim Deaktivieren nicht berücksichtigt.

Software-Endschalter der angetriebenen Gelenkachsen

Die Software-Endschalter der angetriebenen Gelenkachsen wirken, wenn die Positionen der angetriebenen Gelenkachsen gültig und die Software-Endschalter nicht deaktiviert sind. Sie wirken somit nicht im Referenzierbetrieb und bei unreferenziertem Tippen.

Kartesische Software-Endschalter

Die kartesischen Software-Endschalter wirken, wenn die kartesische Position gültig ist und die Software-Endschalter nicht deaktiviert sind. Sie wirken somit nicht im Referenzierbetrieb, bei unreferenziertem Tippen oder wenn die Kinematiktransformation nicht möglich ist. Die kartesischen Software-Endschalter begrenzen die Werkzeugkoordinaten im Basiskoordinatensystem für die translatorischen Freiheitsgrade X, Y und Z sowie die Rotation um diese Achsen. Hier werden nicht die Gelenkachsen der Kinematik begrenzt, sondern die 3D-Raumkoordinaten des Werkzeugs.

Software-Endschalter von Schlepp-Gelenkachsen

Die Software-Endschalter von Schlepp-Gelenkachsen wirken, wenn die kartesische Position gültig ist und die Software-Endschalter nicht deaktiviert sind. Sie wirken somit nicht im Referenzierbetrieb, bei unreferenziertem Tippen oder wenn die Kinematiktransformation nicht möglich ist. Die Software-Endschalter von Schlepp-Gelenkachsen begrenzen die Bewegung von Gelenkachsen, denen kein Antrieb zugeordnet ist. Ein Beispiel hierzu sind die Ellbogen des Kinematikmodells TRIPOD_RRR_M10.

26873346/DE – 07/2021

Software-Endschalter von Schlepp-Abständen

Die Software-Endschalter von Schlepp-Abständen wirken, wenn die kartesische Position gültig ist und die Software-Endschalter nicht deaktiviert sind. Sie wirken somit nicht im Referenzierbetrieb, bei unreferenziertem Tippen oder wenn die Kinematiktransformation nicht möglich ist. Die Software-Endschalter von Schlepp-Abständen begrenzen ausgewählte Abstände bei bestimmten Kinematikmodellen. Ein Beispiel hierzu sind die Abstände zu den seitlichen Begrenzungen des Kinematikmodells QUADROPOD_LLLL_M20.

6.9 Kinematikmodelle

Das Softwaremodul stellt zahlreiche Kinematikmodelle zum Konfigurieren von Applikationen zur Verfügung. Das Auswählen und Konfigurieren dieser Kinematikmodelle erfolgt über das Konfigurationsmenü "Kinematic Model" (→ 🗎 113). In diesem Kapitel finden Sie detailliertere Informationen zu den Kinematikmodellen und den darin enthaltenen Parametern.

HINWEIS

Das ausgewählte Kinematikmodell (angezeigt in der "3D-Simulation" (→ 🗎 142) im RobotMonitor) muss mit dem realen Roboter übereinstimmen. Andernfalls ist ein korrekter Betrieb der Kinematiksteuerung nicht möglich. Zudem müssen die realen Achsen des Roboters gemäß der Anzeige in der 3D-Simulation referenziert und in der Drehrichtung angepasst werden. Eine Anleitung hierzu folgt weiter unten im Kapitel "Achsen referenzieren und Funktionstest durchführen" (→ 🗎 110).

6.9.1 Not-Halt-Rampen und Software-Endschalter der Gelenke

Das Konfigurieren der Not-Halt-Rampen und Software-Endschalter erfolgt bei allen verfügbaren Kinematikmodellen über folgende Einstellungsfelder:

Parameterbezeichnung	Beschreibung
Emergency Stop Deceleration	Verzögerung, mit der die Gelenkachse bei einem Not-Halt abbremst.
Emergency Stop Jerk	Ruck, mit dem die Verzögerung der Gelenkachse bei einem Not-Halt aufgebaut wird. Diese Einstellung dient zur Ruckbegrenzung, also zur Vermeidung von Stößen und Schäden an der Mechanik.
Software Limit Switch	Zulässiger Bereich der Gelenkachse. Wird dieser Bereich überschritten, tritt ein Fehlerzustand ein. Die Software-Endschalter sollten so gewählt werden, dass sie um den Bremsweg entfernt im Arbeitsraum vor den Software-Endschaltern der Umrichter liegen.

Weiterführende Informationen finden Sie in den Kapiteln "Bewegungsparametersätze" (→ 🗎 28), "Bewegungsparameter einstellen" (→ 🗎 235) und "Software-Endschalter" (→ 🗎 43).

26873346/DE – 07/2021

6.9.2 Konstellation und Gelenkachsenphasen

Eine kartesische Position (X, Y, Z, RotZ, RotY, RotX) kann bei zahlreichen Kinematikmodellen mit mehreren Konstellationen (z. B. Ellbogen links/rechts) und bei Drehachsen mit Vielfachen von 360° angefahren werden. Die aktuelle Konstellation wird in der Variablen *Basic.Out.SetpointPose.usiConstellation* der Anwenderschnittstelle ausgegeben.

Die gewünschte Konstellation (Constellation) und Gelenkachsphase (JointPhase) können im Programmbetrieb mittels eines PTP-Befehls explizit vorgegeben und angefahren werden. Bei den Kinematikmodellen ist angegeben, welche Konstellationen möglich sind und welche Achsen über den Parameter *JointPhase* parametriert werden können.

Wenn die Überlagerung mehrerer Drehachsen zu einem kartesischen Orientierungswert führt (RotZ, RotY, RotX), so wird die Position der letzten Drehachse aus dem kartesischen Orientierungswert und den Positionen der vorgelagerten Drehachsen ermittelt. Beispielsweise ist bei einem SCARA-Roboter (Kinematikmodell SCARA_LRRR_M10) die letzte Drehachse die Handachse und nur die Schulter- und Ellbogenachse werden mittels *JointPhase* parametriert.

Falls möglich, sollte der Verfahrbereich der vorgelagerten Drehachsen (im Beispiel Schulter, Ellbogen) über die Software-Endschalter auf 360° begrenzt sein, da sich dann die Parametrierung mittels *JointPhase* erübrigt.

Parametrierung: Constellation

- Eingestellter Wert "0" (Default-Einstellung)

  Zuletzt durch den Roboter erreichte Konstellation beibehalten
- Andere Werte

  Explizite Vorgabe der Konstellation

Parametrierung: JointPhase

- Eingestellter Wert "0" (Default-Einstellung)

  Automatische Auswahl der nächstgelegenen Phase ohne Verletzung der Software-Endschalter
- Andere Werte

  Explizite Auswahl der nächsten Phase. Beispielsweise bedeutet "-5", dass die entsprechende Achse in die Phase]-4 * 360°, -5 * 360°] verfahren soll.

26873346/DE – 07/2021

6.9.3 Cartesian Assignment

Ein Kinematikmodell hat eine bestimmte Anordnung von Nullpunkt und Drehrichtung der Gelenkachsen in Bezug auf das Kinematik-Koordinatensystem. In der Standard-Einstellung (Wert 0) ist das Kinematik-Koordinatensystem identisch mit dem BASE-Koordinatensystem.

Das Kinematikmodell kann mit dem Kinematik-Koordinatensystem relativ zum BASE-Koordinatensystem um 90°-Vielfache gekippt werden. Je nach Kinematikmodell gibt es hierzu ausgehend von der Standard-Einstellung bis zu 23 Möglichkeiten.

In der folgenden Tabelle wird zu jedem Cartesian Assignment angegeben, wie das BASE-Koordinatensystem relativ zum Kinematik-Koordinatensystem ausgerichtet ist bzw. aus der inversen Sichtweise, wie das Kinematik-Koordinatensystem relativ zum BASE-Koordinatensystem ausgerichtet ist.

Die Einstellung erfolgt im Konfigurationsmenü des Kinematikmodells und ist ab MOVISUITE®-Berechtigungsstufe "Advanced" möglich.

Cartesian Assignment	Ausrichtung des BASE-Koordinatensystems relativ zum Kinematik-Koordinatensystem			Ausrichtung des Kinematik-Koordinatensystems relativ zum BASE-Koordinatensystem		
	X_BASE	Y_BASE	Z_BASE	X_KIN	Y_KIN	Z_KIN
0	X_KIN	Y_KIN	Z_KIN	X_BASE	Y_BASE	Z_BASE
1	- Y_KIN	X_KIN	Z_KIN	Y_BASE	- X_BASE	Z_BASE
2	- X_KIN	- Y_KIN	Z_KIN	- X_BASE	- Y_BASE	Z_BASE
3	Y_KIN	- X_KIN	Z_KIN	- Y_BASE	X_BASE	Z_BASE
4	X_KIN	- Y_KIN	- Z_KIN	X_BASE	- Y_BASE	- Z_BASE
5	- Y_KIN	- X_KIN	- Z_KIN	- Y_BASE	- X_BASE	- Z_BASE
6	- X_KIN	Y_KIN	- Z_KIN	- X_BASE	Y_BASE	- Z_BASE
7	Y_KIN	X_KIN	- Z_KIN	Y_BASE	X_BASE	- Z_BASE
8	Z_KIN	X_KIN	Y_KIN	Y_BASE	Z_BASE	X_BASE
9	Z_KIN	- Y_KIN	X_KIN	Z_BASE	- Y_BASE	X_BASE
10	Z_KIN	- X_KIN	- Y_KIN	- Y_BASE	- Z_BASE	X_BASE
11	Z_KIN	Y_KIN	X_KIN	- Z_BASE	Y_BASE	X_BASE
12	Y_KIN	Z_KIN	X_KIN	Z_BASE	X_BASE	Y_BASE
13	X_KIN	Z_KIN	- Y_KIN	X_BASE	- Z_BASE	Y_BASE
14	- Y_KIN	Z_KIN	- X_KIN	- Z_BASE	- X_BASE	Y_BASE
15	- X_KIN	Z_KIN	Y_KIN	- X_BASE	Z_BASE	Y_BASE
16	- Z_KIN	Y_KIN	X_KIN	Z_BASE	Y_BASE	- X_BASE
17	- Z_KIN	X_KIN	- Y_KIN	Y_BASE	- Z_BASE	- X_BASE
18	- Z_KIN	- Y_KIN	- X_KIN	- Z_BASE	- Y_BASE	- X_BASE
19	- Z_KIN	- X_KIN	Y_KIN	- Y_BASE	Z_BASE	- X_BASE
20	X_KIN	- Z_KIN	Y_KIN	X_BASE	Z_BASE	- Y_BASE
21	- Y_KIN	- Z_KIN	X_KIN	Z_BASE	- X_BASE	- Y_BASE
22	- X_KIN	- Z_KIN	- Y_KIN	- X_BASE	- Z_BASE	- Y_BASE
23	Y_KIN	- Z_KIN	- X_KIN	- Z_BASE	- X_BASE	- Y_BASE

26873346/DE – 07/2021

6.9.4 LargeModels-Kinematikmodelle

Lizenzierung

Für den Betrieb eines Kinematikmodells des MOVIKIT® Robotics addon LargeModels sind die entsprechende Steuerungs-Software und je Roboter-Instanz eine Lizenz MOVIKIT® Robotics addon LargeModels erforderlich.

Die Lizenz des MOVIKIT® Robotics addon LargeModels sowie die dazugehörige LargeModels-Steuerungs-Software sind aufgrund der Möglichkeit, in Verbindung miteinander mehr als 4 interpolierende Achsen oder mindestens 2 rotatorische Freiheitsgrade am Werkzeugflansch simultan zur "Bahnsteuerung" zu koordinieren, von den derzeit gültigen Güterlisten (Anhang I DUVO 428/2009; Listennummer: 2D002) erfasst. Die Handelspapiere enthalten einen Hinweis auf die Dual-Use-Relevanz durch Nennung der Ausfuhrlistennummer AL-Nr. 2D002. Die Kunden sind in Folge selbst für die rechtmäßige Behandlung verantwortlich.

Die LargeModels-Steuerungs-Software wird nicht automatisch mit MOVISUITE® installiert und muss explizit erworben werden. Für jedes betroffene Kinematikmodell gibt es einen eigenen USB-Stick mit der dazugehörigen Steuerungs-Software. Der USB-Stick hat keine Dongle-Funktionalität, d. h. es besteht keine Notwendigkeit, dass der USB-Stick während des Steuerbetriebs im Engineering-PC oder im MOVI-C® CONTROLLER gesteckt ist. Pro Kinematikmodell und Anwendung muss der USB-Stick nur einmal erworben werden. Es können damit beliebig viele Roboter-Instanzen betrieben werden.

Die Steuerungs-Software wird in das MOVISUITE®-Projekt eingebunden. Solange die Steuerungs-Software für das ausgewählte LargeModels-Kinematikmodell auf dem Engineering-PC nicht installiert ist, bricht die Aktualisierung des IEC-Projekts mit dem im Meldungsfenster des IEC-Editors angezeigten Fehler ‚Das Objekt "SEW MOVIKIT Robotics LM …" vom Typ "PLCLibrary" konnte nicht erzeugt werden.‘ ab.

Die Softwarelizenz MOVIKIT® Robotics addon LargeModels kann nicht als Testlizenz aktiviert werden. Es können jedoch Softwarelizenzen nachträglich erworben und auf einer Speicherkarte aktiviert werden. Auch andere Softwarelizenzen sind beliebig kombinierbar und nachträglich aktivierbar.

Die Dual-Use-relevante Steuerungs-Software ist nach Installation im Bibliotheksrepository enthalten und wird in ein MOVISUITE®-Projekt bzw. -Projektarchiv (Projektdatei `*.msproj` oder Projektexport `*.mspro`) genau dann integriert, wenn die Steuerungs-Software zur Steuerung eines LargeModels-Kinematikmodells im Bibliotheksverwalter eingebunden ist. Das ist der Fall, wenn in MOVISUITE® in dem betroffenen Projekt ein LargeModels-Kinematikmodell konfiguriert ist oder zuvor konfiguriert war und das IEC-Projekt aktualisiert wurde oder wenn die Steuerungs-Software manuell in den Bibliotheksverwalter eingebunden wurde. Der Empfänger solch eines MOVISUITE®-Projekts bzw. -Projektarchivs hat Zugriff auf die enthaltene Steuerungs-Software. Für die Rechtmäßigkeit der Weitergabe der Dual-Use-relevanten Steuerungs-Software mit solch einem Projekt bzw. Projektarchiv sind die Kunden selbst verantwortlich. **HINWEIS:** Im Bibliotheksverwalter nicht eingebundene Bibliotheken werden hingegen nicht in das Projektarchiv integriert und somit auch nicht mit einem MOVISUITE®-Projekt bzw. -Projektarchiv (z. B. zur Steuerung einer anderen Maschine ohne Dual-Use-relevante Software) weitergegeben.

Eine Archivierung des MOVISUITE®-Projekts bzw. Speichern des Projektarchivs auf der Speicherkarte (Data Restore) ist zu unterlassen bzw. zu deaktivieren, wenn Unbefugte Zugang zu der Speicherkarte haben.

Die Version der LargeModels-Steuerungs-Software ist anhand des Ausgabedatums des USB-Sticks sowie der Versionsnummer des Software-Packages ersichtlich. Wenn ein neuerer Ausgabestand erforderlich ist, muss ein neuer USB-Stick erworben werden.

26873346/DE – 07/2021

Installation Steuerungs-Software

1. Öffnen Sie MOVISUITE® und den IEC-Editor.
2. Öffnen Sie im IEC-Editor über das Menü "Tools" den "Package Manager".
3. Klicken Sie auf die Schaltfläche [Installieren].
4. Wählen Sie das zu installierende Package "SEW_MOVIKIT_Robotics_addon_LM_... .package" vom USB-Stick aus.
5. Führen Sie die Installation des Packages durch. Wählen Sie dabei "Typische Installation" aus.
6. Schließen Sie den IEC-Editor und MOVISUITE®.
7. Öffnen Sie MOVISUITE® erneut.
8. Aktualisieren Sie das IEC-Projekt.

Interpolation bei 3 Orientierungsfreiheitsgraden

Wenn das gewählte Kinematikmodell 3 Orientierungsfreiheitsgrade aufweist, kann eine bestimmte Orientierung im Raum durch mehrere Tupel von Kardan-Winkeln RotZ/Y/X dargestellt werden. Die Interpolation von RotZ/Y/X ist bei 3 Orientierungsfreiheitsgraden nicht eindeutig und führt wegen des Gimbal Locks zum Verlust eines Orientierungsfreiheitsgrads. Deshalb wird die Orientierung bei Robotern mit 3 Orientierungsfreiheitsgraden im Raum auf dem kürzesten Weg zur Zielorientierung hingeführt. Die Interpolation zur Zielorientierung ist für die eindeutige Ausführung der Bewegung in einem Bewegungsbefehl auf < 180° begrenzt.

Die während der Interpolation jeweils erreichte Orientierung wird mit einem möglichen Tupel der Kardan-Winkel RotZ/Y/X ausgegeben. Die Winkel springen während der Interpolation, da es keine stetigen Verläufe für alle Orientierungsänderungen gibt. Da die Kardan-Winkel RotZ/Y/X nicht eindeutig sind, werden die Software-Endschalter RotZ/RotY/RotX nicht geprüft und sind deshalb in der Konfiguration auch nicht einstellbar.

Die partielle Zielvorgabe von RotZ/Y/X ist zu Beginn eines Programms oder nach einem Wechsel des Koordinatensystems problematisch, da im Allgemeinen nicht bekannt ist, mit welchen Kardan-Winkeln die Orientierung zu Beginn oder nach dem Wechsel dargestellt wird und die nicht angegebenen Koordinaten somit nicht eindeutig wären.

Wenn mittels der Zuweisung *SET_ABS_REL_POS* als *TargetPos* "Relative" eingestellt ist, beziehen sich die Zielkoordinaten relativ auf die letzen Zielkoordinaten. Auch relative RotZ/Y/X beziehen sich auf die Werte des letzten Bewegungsbefehls und stellen keine homogene Koordinatentransformation dar. Das ist unabhängig von der Anzahl von Orientierungsfreiheitsgraden der Fall. Bei 3 Orientierungsfreiheitsgraden ist die relative Angabe von RotZ/Y/X zu Beginn eines Programms oder nach einem Wechsel des Koordinatensystems jedoch problematisch, da im Allgemeinen nicht bekannt ist, mit welchen Kardan-Winkeln die Orientierung zu Beginn oder nach dem Wechsel dargestellt wird und somit die Bezugswerte für die relativen Koordinaten nicht eindeutig sind.

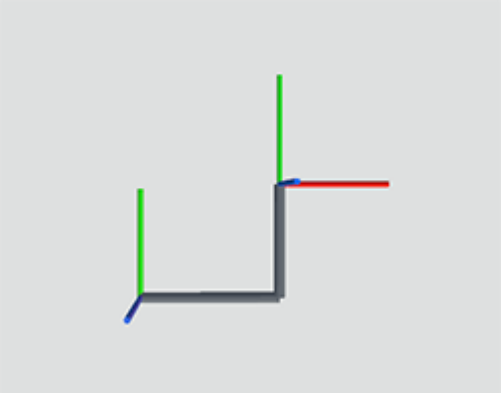
Es wird deshalb empfohlen, beim ersten Befehl in einem Programm und nach Wechsel des Koordinatensystems alle 3 Werte absolut anzugeben, also alle 3 Koordinaten explizit oder eine Posevariable mit Einstellung *TargetPos* "Absolute".

Die Angaben in diesem Abschnitt beziehen sich nicht auf die XYZ-Koordinaten. Weiter beziehen sie sich nicht auf RotZ/Y/X, wenn im Kinematikmodell weniger als 3 Orientierungsfreiheitsgrade vorhanden sind.

26873346/DE – 07/2021

6.9.5 CARTESIAN_GANTRY_LL_M10

Enthalten in der Lizenz MOVIKIT® Robotics.

CARTESIAN_GANTRY_LL_M10	
	CARTESIAN GANTRY (Portal) mit 2 Linearachsen: • Gelenkachse 1: X-Richtung • Gelenkachse 2: Y-Richtung

Gelenkachsen (Joints)

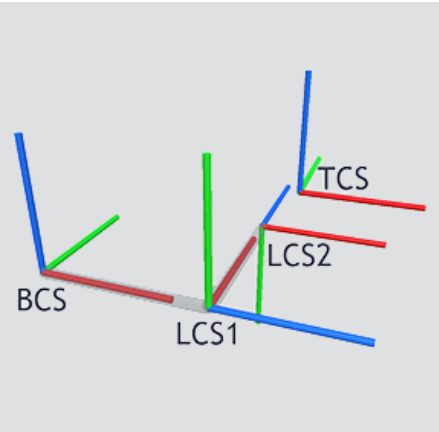
Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

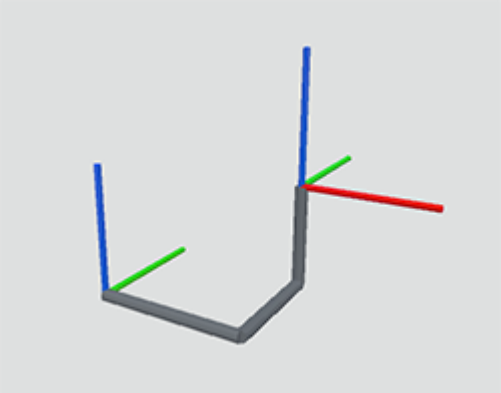


33968986635

26873346/DE – 07/2021

6.9.6 CARTESIAN_GANTRY_LLL_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

CARTESIAN_GANTRY_LLL_M10	
	CARTESIAN GANTRY (Portal) mit 3 Linearachsen: • Gelenkachse 1: X-Richtung • Gelenkachse 2: Y-Richtung • Gelenkachse 3: Z-Richtung

Gelenkachsen (Joints)

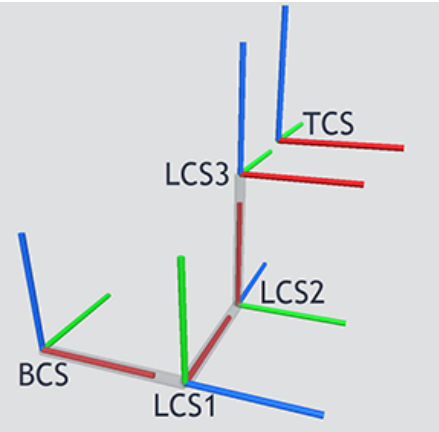
Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

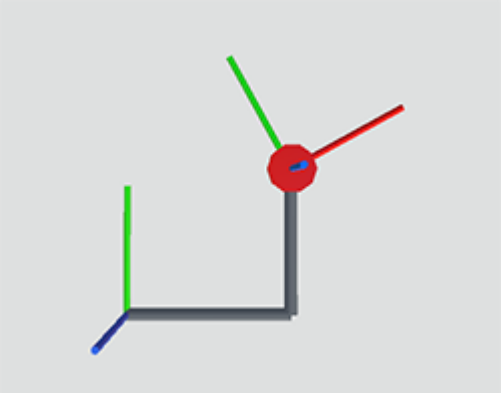


33968989067

26873346/DE – 07/2021

6.9.7 CARTESIAN_GANTRY_LLR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

CARTESIAN_GANTRY_LLR_M10	
	CARTESIAN GANTRY (Portal) mit 2 Linearachsen und 1 Drehachse (Drehung um Z): • Gelenkachse 1: X-Richtung • Gelenkachse 2: Y-Richtung • Gelenkachse 3: Drehung um Z

Gelenkachsen (Joints)

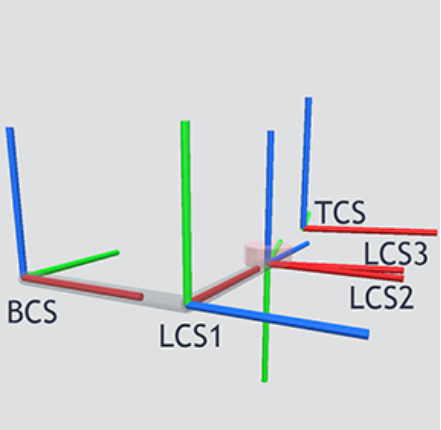
Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

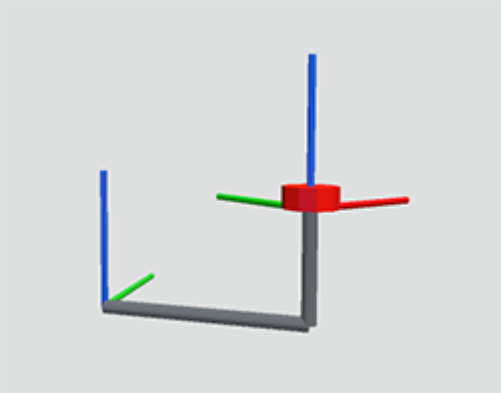


33968993931

26873346/DE – 07/2021

6.9.8 CARTESIAN_GANTRY_LLR_M20

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

CARTESIAN_GANTRY_LLR_M20	
	CARTESIAN GANTRY (Portal) mit 2 Linearachsen und 1 Drehachse (Drehung um Z): • Gelenkachse 1: X-Richtung • Gelenkachse 2: Z-Richtung • Gelenkachse 3: Drehung um Z

Gelenkachsen (Joints)

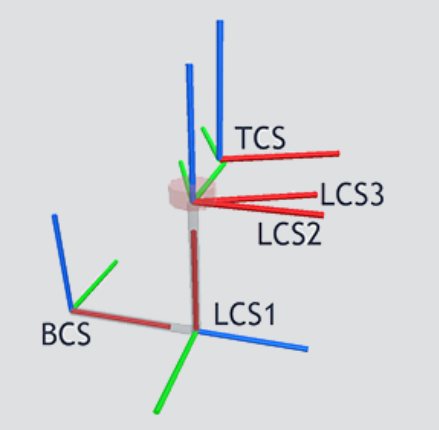
Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

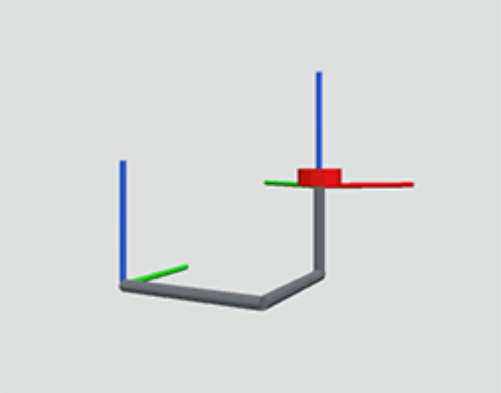


33968996363

26873346/DE – 07/2021

6.9.9 CARTESIAN_GANTRY_LLLR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

CARTESIAN_GANTRY_LLLR_M10	
	CARTESIAN GANTRY (Portal) mit 3 Linearachsen und 1 Drehachse (Drehung um Z): • Gelenkachse 1: X-Richtung • Gelenkachse 2: Y-Richtung • Gelenkachse 3: Z-Richtung • Gelenkachse 4: Drehung um Z

Gelenkachsen (Joints)

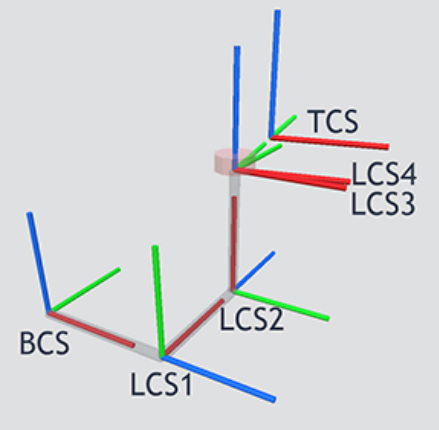
Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

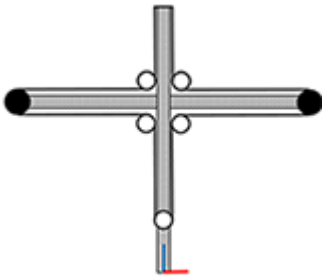


33968991499

26873346/DE – 07/2021

6.9.10 ROLLER_GANTRY_RR_M10

Enthalten in der Lizenz MOVIKIT® Robotics.

ROLLER_GANTRY_RR_M10	
	ROLLER GANTRY mit 2 Drehachsen: • Gelenkachsen 1, 2: Antrieb des umlaufenden Riemens in der ZX-Ebene • Gelenkachse 2: In Richtung X-Achse gegenüber von Gelenkachse 1 angebracht

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Roll 1 (left)	
Pitch	Teilung Riemenscheibe 1 (Wert > 0)
Number Of Teeth	Anzahl Zähne Riemenscheibe 1 (Wert > 0)
Roll 2 (right)	
Pitch	Teilung Riemenscheibe 2 (Wert > 0)
Number Of Teeth	Anzahl Zähne Riemenscheibe 2 (Wert > 0)

Gelenkachsen (Joints)

Im Antriebsstrang die Gelenkachsen 1 und 2 als Drehachsen mit Anwendereinheit Grad einstellen. Siehe Kapitel "Untergeordnete Knoten konfigurieren" (→ 🗎 100). Sie entsprechen den Riemenscheiben 1 und 2. Die Teilung und Zähnezahl werden nicht im Antriebsstrang, sondern in der Kinematik-Konfiguration eingestellt.

Die Software-Endschalter der Gelenkachsen 1 und 2 sollten ungenutzt, also auf sehr großen Werten eingestellt bleiben.

Das Referenzieren der Gelenkachsen 1 und 2 erfolgt zunächst durch eine Referenzierung der Einzelachsen an der Stelle, an der der Roboter steht. Anschliessend lässt er sich kartesisch bis zu Referenzschaltern verfahren, z. B. zunächst in Z- und dann in X-Richtung. Sobald die Referenzschalter durch Berührung z. B. des Werkzeugs betätigt werden, wird das SRL-Programm angehalten bzw. der kartesische Tippbetrieb durch Wegnahme der Jog-Signale unterbrochen. Anschliessend wird der Roboter an der erreichten Stelle erneut durch eine Referenzierung der Einzelachsen referenziert, sodass X = Z = 0 gilt.

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

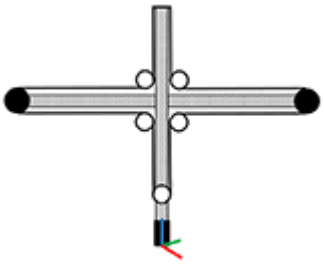
Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

26873346/DE – 07/2021

6.9.11 ROLLER_GANTRY_RRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModel.

ROLLER_GANTRY_RRR_M10	
	ROLLER GANTRY mit 3 Drehachsen: • Gelenkachsen 1, 2: Antrieb des umlaufenden Riemens in der ZX-Ebene • Gelenkachse 2: In Richtung X-Achse gegenüber von Gelenkachse 1 angebracht • Gelenkachse 3: Drehung um Achse parallel zur Z-Achse

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Roll 1 (left)	
Pitch	Teilung Riemenscheibe 1 (Wert > 0)
Number Of Teeth	Anzahl Zähne Riemenscheibe 1 (Wert > 0)
Roll 2 (right)	
Pitch	Teilung Riemenscheibe 2 (Wert > 0)
Number Of Teeth	Anzahl Zähne Riemenscheibe 2 (Wert > 0)

Gelenkachsen (Joints)

Im Antriebsstrang die Gelenkachsen 1 und 2 als Drehachsen mit Anwendereinheit Grad einstellen. Siehe Kapitel "Untergeordnete Knoten konfigurieren" (→ 🗎 100). Sie entsprechen den Riemenscheiben 1 und 2. Die Teilung und Zähnezahl werden nicht im Antriebsstrang, sondern in der Kinematik-Konfiguration eingestellt.

Die Software-Endschalter der Gelenkachsen 1 und 2 sollten ungenutzt, also auf sehr großen Werten eingestellt bleiben.

Das Referenzieren der Gelenkachsen 1 und 2 erfolgt zunächst durch eine Referenzierung der Einzelachsen an der Stelle, an der der Roboter steht. Anschliessend lässt er sich kartesisch bis zu Referenzschaltern verfahren, z. B. zunächst in Z- und dann in X-Richtung. Sobald die Referenzschalter durch Berührung z. B. des Werkzeugs betätigt werden, wird das SRL-Programm angehalten bzw. der kartesische Tippbetrieb durch Wegnahme der Jog-Signale unterbrochen. Anschliessend wird der Roboter an der erreichten Stelle erneut durch eine Referenzierung der Einzelachsen referenziert, sodass X = Z = 0 gilt.

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

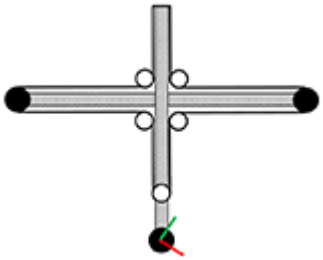
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.12 ROLLER_GANTRY_RRLR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModel.

ROLLER_GANTRY_RRLR_M10	
	ROLLER GANTRY mit 3 Dreh- und einer Linearachse: • Gelenkachsen 1, 2: Antrieb des umlaufenden Riemens in der ZX-Ebene • Gelenkachse 2: In Richtung X-Achse gegenüber von Gelenkachse 1 angebracht • Gelenkachse 3: Z-Richtung (senkrecht aus Zeichenebene heraus zeigend) • Gelenkachse 4: Drehung um Achse parallel zur Z-Achse

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Roll 1 (left)	
Pitch	Teilung Riemenscheibe 1 (Wert > 0)
Number Of Teeth	Anzahl Zähne Riemenscheibe 1 (Wert > 0)
Roll 2 (right)	
Pitch	Teilung Riemenscheibe 2 (Wert > 0)
Number Of Teeth	Anzahl Zähne Riemenscheibe 2 (Wert > 0)

Gelenkachsen (Joints)

Im Antriebsstrang die Gelenkachsen 1 und 2 als Drehachsen mit Anwendereinheit Grad einstellen. Siehe Kapitel "Untergeordnete Knoten konfigurieren" (→ 🗎 100). Sie entsprechen den Riemenscheiben 1 und 2. Die Teilung und Zähnezahl werden nicht im Antriebsstrang, sondern in der Kinematik-Konfiguration eingestellt.

Die Software-Endschalter der Gelenkachsen 1 und 2 sollten ungenutzt, also auf sehr großen Werten eingestellt bleiben.

Das Referenzieren der Gelenkachsen 1 und 2 erfolgt zunächst durch eine Referenzierung der Einzelachsen an der Stelle, an der der Roboter steht. Anschliessend lässt er sich kartesisch bis zu Referenzschaltern verfahren, z. B. zunächst in Z- und dann in X-Richtung. Sobald die Referenzschalter durch Berührung betätigt werden, wird das SRL-Programm angehalten bzw. der kartesische Tippbetrieb durch Wegnahme der Jog-Signale unterbrochen. Anschliessend wird der Roboter an der erreichten Stelle erneut durch eine Referenzierung der Einzelachsen referenziert, sodass gilt X = Z = 0.

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

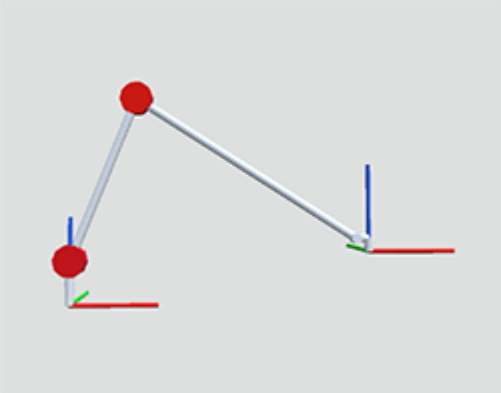
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.13 SCARA_RR_M10

Enthalten in der Lizenz MOVIKIT® Robotics.

SCARA_RR_M10	
	SCARA mit 2 Drehachsen: • Gelenkachse 1: Dreht den Oberarm um die Schulterachse (parallel zur Y-Achse). • Gelenkachse 2: Dreht den Unterarm um die Ellbogenachse (parallel zur Y-Achse). Die Flanschorientierung wird mit einem Parallelogramm oder Riemen konstant gehalten oder das Werkzeug besteht nur z. B. aus einem Stift, dessen Orientierung keine Rolle spielt.

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Length Upper Arm	Länge Oberarm (Wert > 0)
Length Forearm	Länge Unterarm (Wert > 0)
Flange Offset (X)	Versatz am Flansch in X-Richtung
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**1, 2**

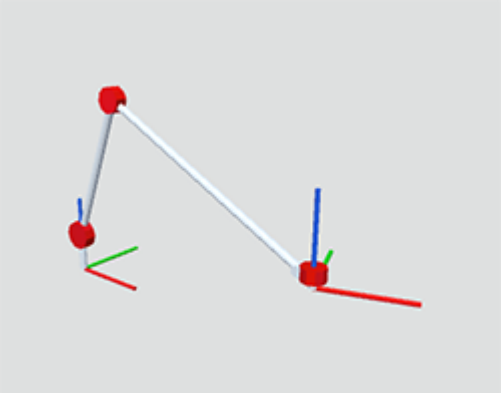
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.14 SCARA_RRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

SCARA_RRR_M10	
	SCARA mit 3 Drehachsen: • Gelenkachse 1: Dreht den Oberarm um die Schulterachse (parallel zur Y-Achse). • Gelenkachse 2: Dreht den Unterarm um die Ellbogenachse (parallel zur Y-Achse). Der Antrieb ist stationär. • Gelenkachse 3: Dreht das Werkzeug um die Handgelenksachse (parallel zur Z-Achse).

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Length Upper Arm	Länge Oberarm (Wert > 0)
Length Forearm	Länge Unterarm (Wert > 0)
Flange Offset (X)	Versatz am Flansch in X-Richtung
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**1, 2**

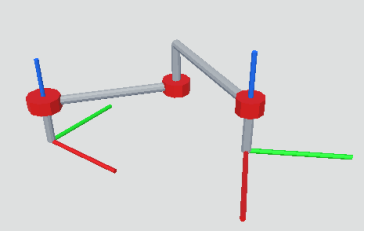
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.15 SCARA_RRR_M20

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

SCARA_RRR_M10	
	SCARA mit 3 Drehachsen: • Gelenkachse 1: Dreht den Oberarm um die Schulterachse (parallel zur Z-Achse). • Gelenkachse 2: Dreht den Unterarm um die Ellbogenachse (parallel zur Z-Achse). • Gelenkachse 3: Dreht das Werkzeug um die Handgelenksachse (parallel zur Z-Achse).

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Length Upper Arm	Länge Oberarm (Wert > 0)
Offset Upper Arm -> Forearm	Versatz vom Ober- zum Unterarm in Z-Richtung
Length Forearm	Länge Unterarm (Wert > 0)
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

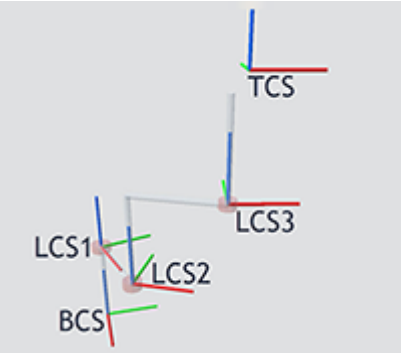
PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**1, 2**

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

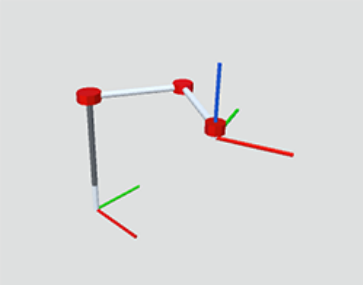


33969018251

26873346/DE – 07/2021

6.9.16 SCARA_LRRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

SCARA_LRRR_M10	
	SCARA mit 1 Linearachse und 3 Drehachsen: • Gelenkachse 1: Z-Richtung • Gelenkachse 2: Dreht den Oberarm um die Schulterachse (parallel zur Z-Achse). • Gelenkachse 3: Dreht den Unterarm um die Ellbogenachse (parallel zur Z-Achse). • Gelenkachse 4: Dreht das Werkzeug um die Handgelenksachse (parallel zur Z-Achse).

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Length Upper Arm	Länge Oberarm (Wert > 0)
Offset Upper Arm -> Forearm	Versatz vom Ober- zum Unterarm in Z-Richtung
Length Forearm	Länge Unterarm (Wert > 0)
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

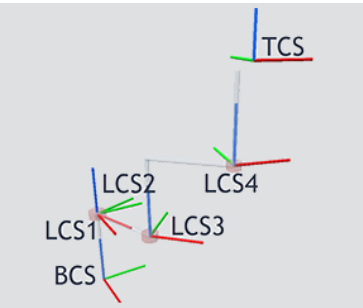
PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**2, 3**

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

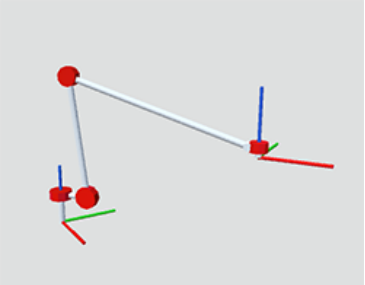


33969010955

26873346/DE – 07/2021

6.9.17 SCARA_RRRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

SCARA_RRRR_M10	
	SCARA mit 4 Drehachsen: • Gelenkachse 1: Dreht den Drehturm um die Drehturmachse (parallel zur Z-Achse). • Gelenkachse 2: Dreht den Oberarm um die Schulterachse. • Gelenkachse 3: Dreht den Unterarm um die Ellbogenachse. • Gelenkachse 4: Dreht Werkzeug um Z-Achse. Die Neigung des Werkzeugs wird mit einem Parallelogramm oder Riemen konstant gehalten.

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Base Offset (X)	Versatz am Sockel in X-Richtung
Base Offset (Y)	Versatz am Sockel in Y-Richtung
Length Upper Arm	Länge Oberarm (Wert > 0)
Length Forearm	Länge Unterarm (Wert > 0)
Flange Offset (X)	Versatz am Flansch in X-Richtung
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1 bis 4**
Über *JointPhase* parametrierbare Gelenkachsen	**1, 2, 3**

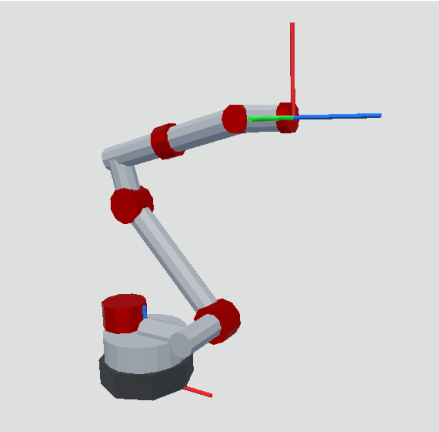
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.18 ARTICULATED_RRRRRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon LargeModels. Weitere Informationen dazu finden Sie im Kapitel "LargeModels-Kinematikmodelle" (→ 🗎 47).

ARTICULATED_RRRRRR_M10	
	ARTICULATED mit 6 Drehachsen. • Gelenkachse 1: Dreht den Drehturm um die Drehturmachse (parallel zur Z-Achse). • Gelenkachse 2: Dreht den Oberarm um die Schulterachse. • Gelenkachse 3: Dreht den Unterarm um die Ellbogenachse. • Gelenkachsen 4 bis 6: Schneiden sich im Handwurzelpunkt.

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Base Offset (X)	Versatz am Sockel in X-Richtung
Base Offset (Y)	Versatz am Sockel in Y-Richtung
Length Upper Arm	Länge Oberarm (Wert > 0)
Offset Upper Arm to Forearm	Versatz Ober- zum Unterarm
Length Forearm	Länge Unterarm (Wert > 0)
Offset Wrist to Flange	Versatz am Flansch vom Handwurzelpunkt zum Ursprung des Flanschkoordinatensystems (Wert > 0)

26873346/DE – 07/2021

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

Zusätzlich zu den 6 Gelenkachsen ist 1 Kinematiklimitierung zu konfigurieren. Diese wirkt als Software-Endschalter.

Parameterbezeichnung	Beschreibung
Angle between the lines joint 2 to joint 3 and joint 3 to wrist	Winkel zwischen folgenden Verbindungslinien: • Gelenkachse 2 zu Gelenkachse 3 • Gelenkachse 3 zum Handwurzelpunkt (Schnittpunkt der Achsen 4 bis 6)

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1 bis 8**
Über *JointPhase* parametrierbare Gelenkachsen	**1 bis 6**

Physics

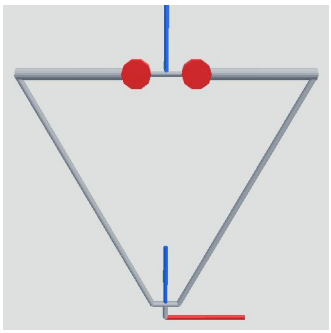
Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.19 DELTA_RR/RRR_M10, DELTA_RRRL_M20

Enthalten in der Lizenz MOVIKIT® Robotics.

DELTA_RR_M10



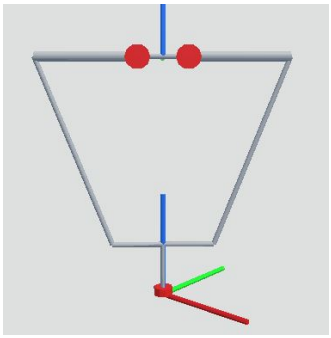
DELTA mit 2 Drehachsen:

- Gelenkachsen 1 und 2: Ermöglichen zusammen die translatorische Bewegung des Werkzeugs in der ZX-Ebene. Die Bewegung einer Gelenkachse ist nicht eindeutig der Bewegung einer kartesischen Achse zuzuordnen.

Die Flanschorientierung wird mit einem Parallelogramm oder Riemen konstant gehalten oder das Werkzeug besteht nur z. B. aus einem Stift, dessen Orientierung keine Rolle spielt.

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

DELTA_RRR_M10



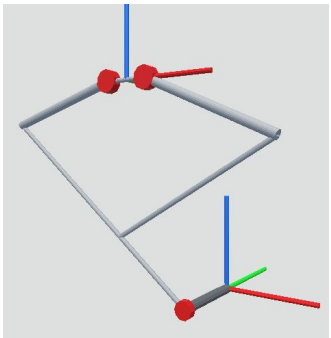
DELTA mit 3 Drehachsen:

- Gelenkachsen 1 und 2: Ermöglichen zusammen die translatorische Bewegung des Werkzeugs in der ZX-Ebene. Die Bewegung einer Gelenkachse ist nicht eindeutig der Bewegung einer kartesischen Achse zuzuordnen.
- Gelenkachse 3: Dreht das Werkzeug um die Handgelenksachse (parallel zur Z-Achse).

Die Orientierung der Verankerung von Gelenkachse 3 wird mit einem Parallelogramm oder Riemen konstant gehalten.

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

DELTA_RRRL_M20



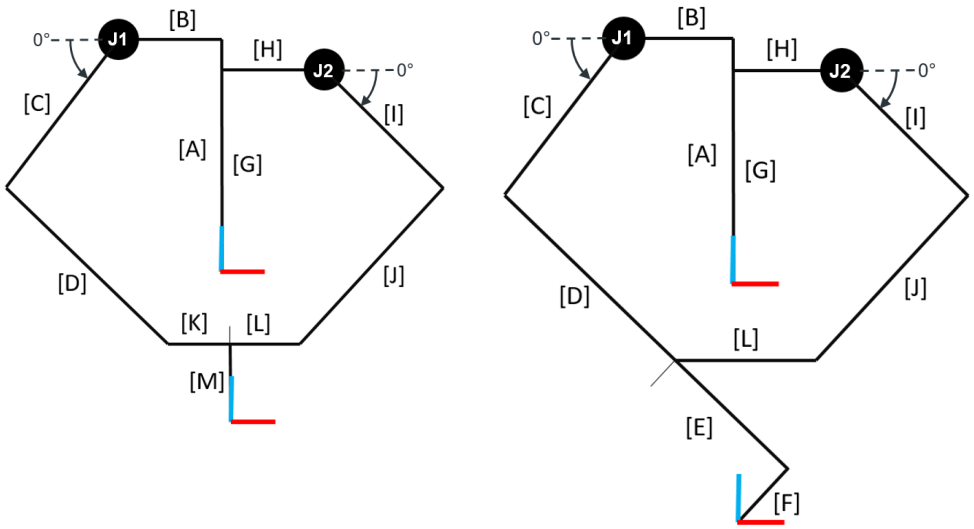
DELTA mit 3 Dreh- und 1 Linearachse:

- Gelenkachsen 1 und 2: Ermöglichen zusammen die translatorische Bewegung des Werkzeugs in der ZX-Ebene. Die Bewegung einer Gelenkachse ist nicht eindeutig der Bewegung einer kartesischen Achse zuzuordnen.
- Gelenkachse 3: Dreht das Werkzeug um die Handgelenksachse (parallel zur Y-Achse).
- Gelenkachse 4: Verschiebt das Werkzeug parallel zur Y-Achse.

Wenn die Parameter (E) und (F) beide gleich 0 sind, wird die Orientierung der Verankerung von Gelenkachse 3 mit einem Parallelogramm oder Riemen konstant gehalten. Ansonsten ist Gelenkachse 3 an der Verlängerung (E) bzw. dem Versatz (F) des Oberarms (D) angebracht und die X-Achse des Flanschkoordinatensystems zeigt in Nullstellung von Gelenkachse 3 in Richtung der Verlängerung von Oberarm (D).

26873346/DE – 07/2021

Kinematik-Parameter (KinPar)



Parameterbezeichnung		Beschreibung
Left Side		
(A)	Offset to Joint (Z)	Versatz zum Gelenk 1 in Z-Richtung
(B)	Offset to Joint (X)	Versatz zum Gelenk 1 in X-Richtung
(C)	Length Upper Arm	Länge Oberam (Wert > 0)
(D)	Length Forearm	Länge Unterarm (Wert > 0)
(E)	Extension	Verlängerung Unterarm ab Verbindung mit anderem Arm (Wert ≥ 0)
(F)	Offset	Orthogonaler Versatz (In Abbildung Wert > 0)
Right Side		
(G)	Offset to Joint (Z)	Versatz zum Gelenk 2 in Z-Richtung
(H)	Offset to Joint (X)	Versatz zum Gelenk 2 in X-Richtung (Wert ≥ Wert B)
(I)	Length Upper Arm	Länge Oberam (Wert > 0)
(J)	Length Forearm	Länge Unterarm (Wert > 0)
Tool Plate		
(K)	Distance Left Side	Werkzeugplatte Abschnitt links (Wert ≥ 0)
(L)	Distance Right Side	Werkzeugplatte Abschnitt rechts (Wert ≥ 0)
(M)	Offset (Z)	Versatz am Flansch in Z-Richtung

Anmerkung:

Die Parameter "Distance Left Side" (K) und "Offset (Z)" (M) können nur dann ungleich "0" sein, wenn für die Parameter "Extension" (E) und "Offset" (F) der Wert "0" gesetzt ist.

26873346/DE – 07/2021

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

Zusätzlich zu den Gelenkachsen sind 3 nicht-angetriebene Gelenke zu konfigurieren. Diese wirken als Software-Endschalter:

Parameterbezeichnung	Beschreibung
Linker Ellbogen	Winkel zwischen Ober- und Unterarm von Arm 1
Rechter Ellbogen	Winkel zwischen Ober- und Unterarm von Arm 2
Schnittpunkt der Arme	Winkel zwischen den beiden Unterarmen

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1 bis 4**
Über *JointPhase* parametrierbare Gelenkachsen	**1, 2**

Physics

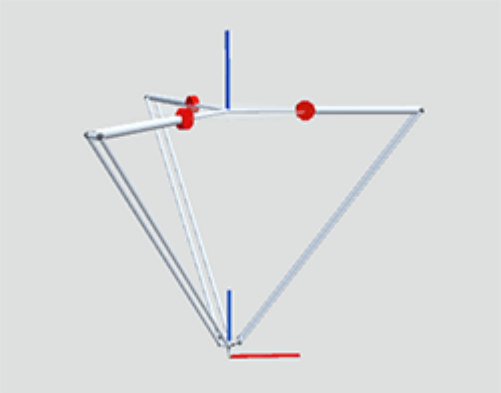
Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.20 TRIPOD_RRR/RRRR/RRRRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

TRIPOD_RRR_M10

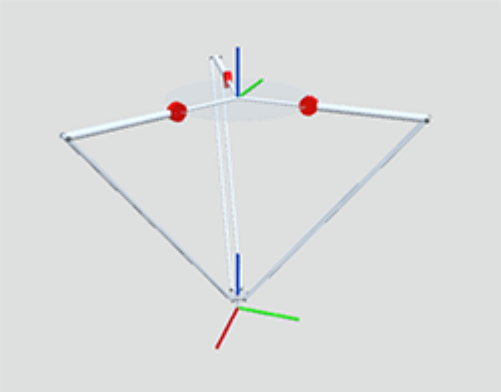


TRIPOD mit 3 Drehachsen:

- Gelenkachsen 1 – 3: Ermöglichen zusammen die translatorische Bewegung des Werkzeugs (XYZ). Die Bewegung eines Antriebs kann nicht eindeutig der Bewegung einer kartesischen Achse zugeordnet werden.

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

TRIPOD_RRRR_M10

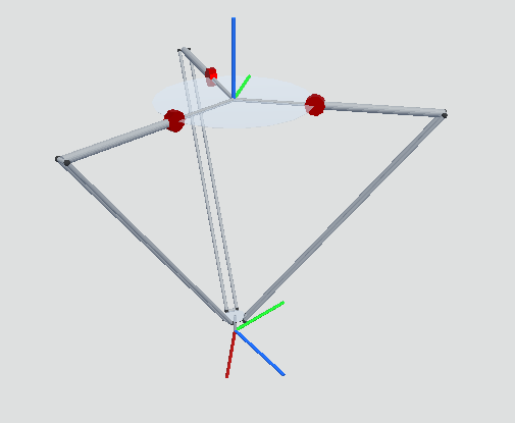


TRIPOD mit 4 Drehachsen:

- Gelenkachsen 1 – 3: Ermöglichen zusammen die translatorische Bewegung des Werkzeugs (XYZ). Die Bewegung eines Antriebs kann nicht eindeutig der Bewegung einer kartesischen Achse zugeordnet werden.
- Gelenkachse 4: Dreht das Werkzeug um die Handgelenksachse (parallel zur Z-Achse).

Enthalten in der Lizenz des MOVIKIT® Robotics addon LargeModels. Siehe auch "LargeModels-Kinematikmodelle" (→ 🗎 47).

TRIPOD_RRRRR_M10

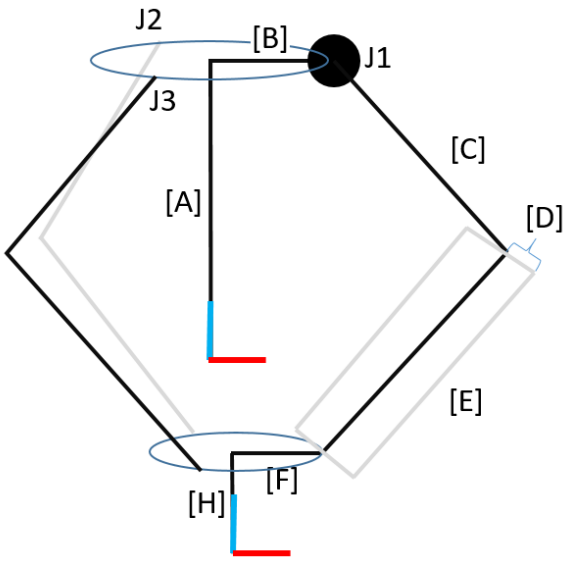


TRIPOD mit 5 Drehachsen:

- Gelenkachsen 1 – 3: Ermöglichen zusammen die translatorische Bewegung des Werkzeugs (XYZ). Die Bewegung eines Antriebs kann nicht eindeutig der Bewegung einer kartesischen Achse zugeordnet werden.
- Gelenkachse 4: Dreht das Werkzeug im Handgelenk um die Z-Achse.
- Gelenkachse 5: Dreht das Werkzeug im Handgelenk um die Y-Achse.

26873346/DE – 07/2021

Kinematik-Parameter (KinPar)



Parameterbezeichnung		Beschreibung
Tripod Kinematic Parameters		
(A)	Height Offset Shoulderplate	Versatz zur Höhe der Schultergelenke in Z-Richtung
(B)	Radius Shoulderplate	Radius des Kreises durch die Schultergelenke (Wert > 0)
(C)	Length Upper Arms	Länge der Oberarme (Wert > 0)
(D)	Half Width Parallelogram	Halbe Breite der Parallelogramme (Wert > 0)
(E)	Length Forearms	Länge der Unterarme (Wert > 0)
(F)	Radius Toolplate	Radius des Kreises durch die Schnittpunkte der Werkzeugplatte mit den Mittellinien der Parallelogramme (Wert > 0)
(G)	Horizontal Offset Toolplate	Versatz am Flansch in X-Richtung
(H)	Vertical Offset Toolplate	Versatz am Flansch in Z-Richtung

26873346/DE – 07/2021

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

Zusätzlich zu den Gelenkachsen sind 6 nicht-angetriebene Gelenke zu konfigurieren. Diese wirken als Software-Endschalter.

Parameterbezeichnung	Beschreibung
Ellbogen 1 (Passive Joint 1)	Winkel zwischen Ober- und Unterarm von Arm 1
Scherwinkel 1 (Passive Joint 2)	Winkel zwischen der Bewegungsebene von Oberarm 1 und dem Unterarm 1
Ellbogen 2 (Passive Joint 3)	Winkel zwischen Ober- und Unterarm von Arm 2
Scherwinkel 2 (Passive Joint 4)	Winkel zwischen der Bewegungsebene von Oberarm 2 und dem Unterarm 2
Ellbogen 3 (Passive Joint 5)	Winkel zwischen Ober- und Unterarm von Arm 3
Scherwinkel 3 (Passive Joint 6)	Winkel zwischen der Bewegungsebene von Oberarm 3 und dem Unterarm 3

PTP-Parameter

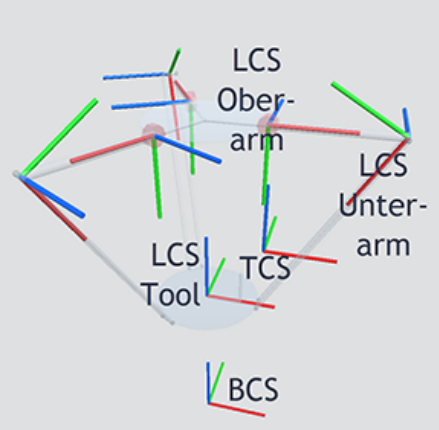
Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1 bis 8**
Über *JointPhase* parametrierbare Gelenkachsen	**1, 2, 3**

Physics

Die Physik-Simulation ist aktivierbar (die Teleskopstange wird nicht unterstützt). Es werden jedoch nur Antriebsmomente berechnet, keine Kippmomente oder Querkräfte.

Die angetriebenen Rotationsgelenke an der Werkzeugplatte der Modelle TRIPOD_RRRR/RRRRR_M10 werden nicht unterstützt.

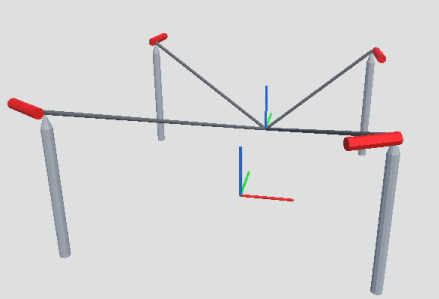


33969023115

26873346/DE – 07/2021

6.9.21 QUADROPOD_LLLL_M20

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

QUADROPOD_LLLL_M20	
	QUADROPOD mit 4 Linearachsen, gewöhnlich als Seile realisiert. Die Seilwinde befindet sich in der Nähe der Umlenkrolle, die den Nullpunkt der Linearachse darstellt. • Gelenkachse 1: Seil von Umlenkrolle 1 zum Schnittpunkt • Gelenkachse 2: Seil von Umlenkrolle 2 zum Schnittpunkt • Gelenkachse 3: Seil von Umlenkrolle 3 zum Schnittpunkt • Gelenkachse 4: Seil von Umlenkrolle 4 zum Schnittpunkt Der Roboter hat die 3 translatorischen Freiheitsgrade XYZ und ist somit überbestimmt.

Inbetriebnahme

Zunächst müssen die Gelenkachsen z. B. mittels JOG_AXIS oder JOG_JOINT ausgefahren, also die Seile abgewickelt werden, so dass sie sich schneiden können, also die Seile am Werkzeug eingehängt werden können.

Eine gültige kartesische Position kann erst dann bestimmt werden, wenn sich die Gelenkachsen 1..3 schneiden. Solange die kartesische Position nicht gültig ist, wird die Warnung 16#17E26 "Forward kinematics failed…" angezeigt.

Bevor kartesisch interpoliert werden darf (JOG_CART oder Linear-/Kreisinterpolation), müssen die Achsen initial zwingend mittels PTP-Interpolation ausgerichtet werden, da der Roboter durch die 4 Gelenkachsen bei 3 kartesischen Freiheitsgraden überbestimmt ist. Darauf ist applikativ zu achten. Die PTP-Bewegung darf pausiert, jedoch nicht mittels Programm-Stopp unterbrochen werden. Vor dem PTP-Befehl dürfen im SRL-Programm nur wenige (< 10) zuweisende Anweisungen stehen (z. B. SET_MOTIONSET, SET_BOOLVAR) und keine wartenden Befehle (z. B. WAIT_TIME) oder andere Bewegungsbefehle. Ansonsten kommt es zu einem Sprung der Sollwerte für die Achsen und in Folge gewöhnlich zu einem Schleppfehler.

Auch nach jedem Tippbetrieb mittels JOG_AXIS oder JOG_JOINT muss wieder initial eine PTP-Bewegung komplett ausgeführt werden.

Nach einer Unterbrechung der Programmausführung z. B. wegen eines Nothalts kann jedoch mittels "Rückpositionierung (BackToPath)" (→ 🗎 39) direkt im Programm fortgefahren werden.

26873346/DE – 07/2021

Kinematik-Parameter (KinPar)

HINWEIS

Die Umlenkpunkte müssen in der Draufsicht mathematisch positiv (also linksrehend) angeordnet sein, z. B. XY-Koordinaten des n-ten Umlenkpunkts im n-ten Quadranten.

Parameterbezeichnung	Beschreibung
Guide Pulley 1-4	
X	X-Koordinate des Umlenkpunkts 1-4
Y	Y-Koordinate des Umlenkpunkts 1-4
Z	Z-Koordinate des Umlenkpunkts 1-4
Flange Offset (Z)	
Versatz am Flansch in Z-Richtung	

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

Zusätzlich zu den 4 Gelenkachsen sind 2 Kinematiklimitierungen zu konfigurieren. Diese wirken als Software-Endschalter.

Parameterbezeichnung	Beschreibung
Distance intersection of ropes to side	Mindestabstand des Schnittpunkts der 4 Seile in der XY-Ebene von den Verbindungslinien der Umlenkpunkte
Distance intersection of ropes to top	Mindestabstand des Schnittpunkts der 4 Seile in Z-Richtung von der niedrigsten Höhe (Z) der 4 Umlenkpunkte

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

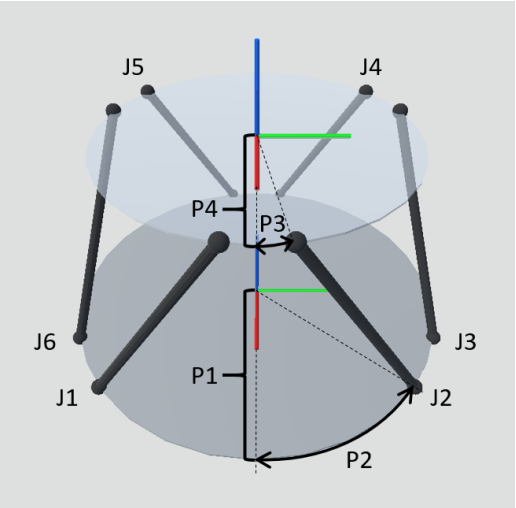
Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.22 HEXAPOD_LLLLLL_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon LargeModels. Weitere Informationen dazu finden Sie im Kapitel "LargeModels-Kinematikmodelle" (→ 🗎 47).

HEXAPOD_LLLLLL_M10



HEXAPOD mit 6 Linearachsen:

- Gelenkachse 1 bis 6: Linearachse von der Grundplatte zur Werkzeugplatte, Nullpunkt bei Länge 0, Gelenkachswerte > 0, wenn ausgefahren

Inbetriebnahme

Bevor kartesisch interpoliert werden darf (JOG_CART oder Linear-/Kreisinterpolation), müssen die Achsen initial zwingend mittels PTP-Interpolation ausgerichtet werden. Darauf ist applikativ zu achten. Die PTP-Bewegung darf pausiert, jedoch nicht mittels Programm-Stopp unterbrochen werden. Vor dem PTP-Befehl dürfen im SRL-Programm nur wenige (< 10) zuweisende Anweisungen stehen (z. B. SET_MOTIONSET, SET_BOOLVAR) und keine wartenden Befehle (z. B. WAIT_TIME) oder andere Bewegungsbefehle. Ansonsten kommt es zu einem Sprung der Sollwerte für die Achsen und in Folge gewöhnlich zu einem Schleppfehler.

Auch nach jedem Tippbetrieb mittels JOG_AXIS oder JOG_JOINT muss wieder initial eine PTP-Bewegung komplett ausgeführt werden.

Nach einer Unterbrechung der Programmausführung z. B. wegen eines Nothalts kann jedoch mittels "Rückpositionierung (BackToPath)" (→ 🗎 39) direkt im Programm fortgefahren werden.

26873346/DE – 07/2021

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
P1	Radius der Gelenke auf der Grundplatte
P2	Halber Winkel zwischen den Gelenken je eines Paares (1/2, 3/4, 5/6) auf der Grundplatte
P3	Halber Winkel zwischen den Gelenken je eines Paares (1/2, 3/4, 5/6) auf der Werkzeugplatte
P4	Radius der Gelenke auf der Werkzeugplatte

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

Zusätzlich zu den 6 Gelenkachsen sind 6 Kinematiklimitierungen zu konfigurieren. Diese wirken als Software-Endschalter.

Parameterbezeichnung	Beschreibung
Kinematiklimitierung 1-6	Winkel zwischen linearer Gelenkachse 1-6 und dem Normalenvektor auf der Grundplatte

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1**
Über *JointPhase* parametrierbare Gelenkachsen	**-**

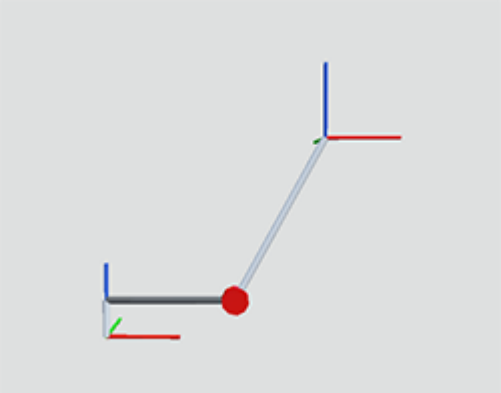
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.23 MIXED_LR_M10

Enthalten in der Lizenz MOVIKIT® Robotics.

MIXED_LR_M10	
	Ein MIXED-Kinematikmodell mit 1 Linearachse und 1 Drehachse • Gelenkachse 1: X-Richtung • Gelenkachse 2: Drehung um Achse parallel zur Y-Achse. Die Flanschorientierung wird mit einem Parallelogramm oder Riemen konstant gehalten. Oder das Werkzeug besteht nur z. B. aus einem Stift, dessen Orientierung keine Rolle spielt.

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Arm Length	Länge des Arms (Wert > 0)

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Über *JointPhase* parametrierbare Gelenkachsen	**2**

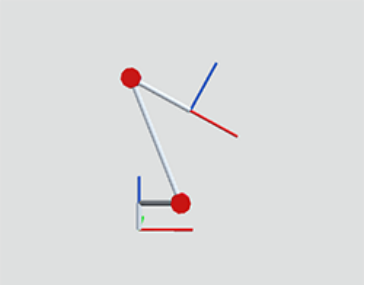
Physics

Die Physik-Simulation ist bei diesem/n Kinematikmodell/en nicht aktivierbar.

26873346/DE – 07/2021

6.9.24 MIXED_LRR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

MIXED_LRR_M10	
	Ein MIXED-Kinematikmodell mit 1 Linearachse und 2 Drehachsen • Gelenkachse 1: X-Richtung • Gelenkachse 2: Drehung um Achse parallel zur Y-Achse. • Gelenkachse 3: Drehung um Achse parallel zur Y-Achse.

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Upper Arm Length	Länge Oberarm (Wert > 0)
Forearms Length	Länge Unterarm (Wert > 0)

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

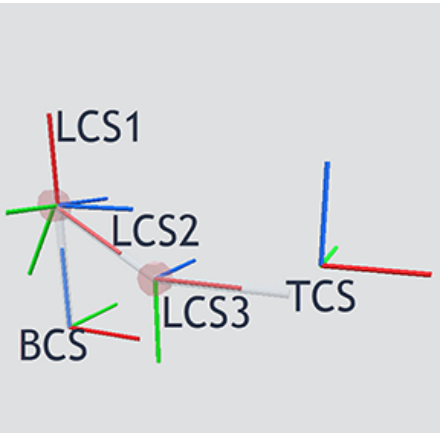
PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Über *JointPhase* parametrierbare Gelenkachsen	**2**

Physics

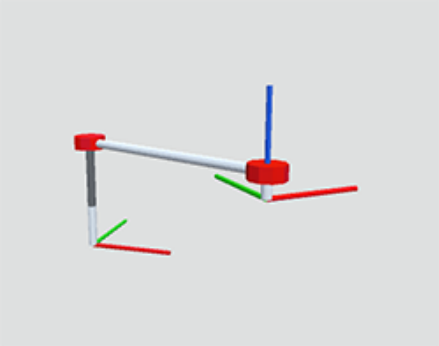
Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.



33969003659

6.9.25 MIXED_LRR_M20

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

MIXED_LRR_M20	
	Ein MIXED-Kinematikmodell mit 1 Linearachse und 2 Drehachsen, das Bewegungen des Flansches auf der Oberfläche eines Zylinders um die Linearachse sowie Drehungen des Flansches (um Achse parallel zu Z) ermöglicht. Die X-Achse verläuft horizontal auf dem Kreisbogen des Zylinders um die Linearachse: • Gelenkachse 1: Z-Richtung • Gelenkachse 2: Dreht den Arm um die Schulterachse (parallel zur Z-Achse). • Gelenkachse 3: Dreht das Werkzeug um die Handgelenksachse (parallel zur Z-Achse).

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Armlength	Länge des Arms (Wert > 0)
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

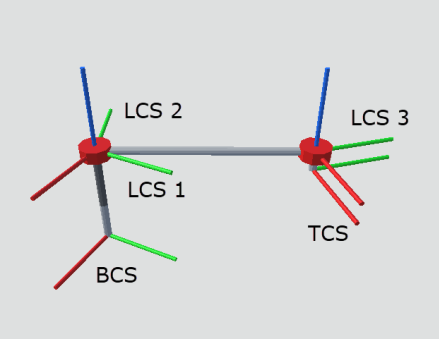
Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

PTP-Parameter

Dieses Kinematikmodell unterstützt ausschliesslich Konstellation 1 und hat keine über *JointPhase* parametrierbaren Gelenkachsen.

Physics

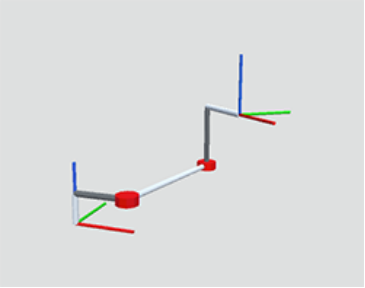
Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.



26873346/DE – 07/2021
3401

6.9.26 MIXED_LRRL_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

MIXED_LRRL_M10	
	Ein MIXED_Kinematikmodell mit 2 Linearachsen und 2 Drehachsen: • Gelenkachse 1: X-Richtung • Gelenkachse 2: Drehung um Achse parallel zur Z-Achse • Gelenkachse 3: Drehung um Achse parallel zur Z-Achse • Gelenkachse 4: Z-Richtung

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Upper Arm Length	Länge Oberarm (Wert > 0)
Forearms Length	Länge Unterarm (Wert > 0)
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

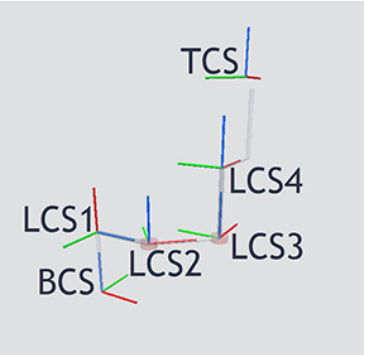
PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**2**

Physics

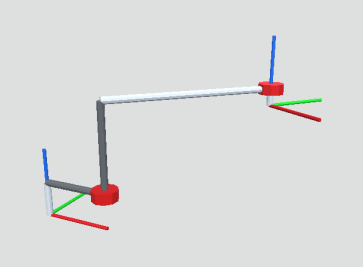
Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.



33969006091

6.9.27 MIXED_LRLR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

MIXED_LRLR_M10	
	Ein MIXED Kinematikmodell mit 2 Linearachsen und 2 Drehachsen: • Gelenkachse 1: X-Richtung • Gelenkachse 2: Drehung um Achse parallel zur Z-Achse • Gelenkachse 3: Z-Richtung • Gelenkachse 4: Drehung um Achse parallel zur Z-Achse

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Arm Length	Länge des Arms zwischen Gelenkachse 3 und 4 (Wert > 0)
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

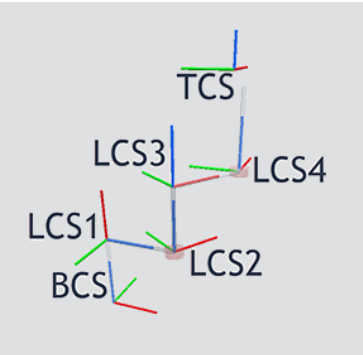
PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**2**

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.

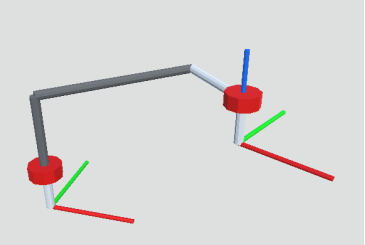


33969001227

26873346/DE – 07/2021

6.9.28 MIXED_RLLR_M10

Enthalten in der Lizenz MOVIKIT® Robotics addon MediumModels.

MIXED_RLLR_M10	
	Ein MIXED Kinematikmodell mit 2 Linearachsen und 2 Drehachsen: • Gelenkachse 1: Drehung um Achse parallel zur Z-Achse • Gelenkachse 2: Z-Richtung • Gelenkachse 3: Linearachse in XY-Ebene • Gelenkachse 4: Drehung um Achse parallel zur Z-Achse

Kinematik-Parameter (KinPar)

Parameterbezeichnung	Beschreibung
Kinematic Parameters	
Base Offset (Z)	Versatz am Sockel in Z-Richtung
Side Offset	Versatz senkrecht zu Gelenkachse 3 In Y-Richtung, wenn Gelenkachse 1 = 0°
Flange Offset (Z)	Versatz am Flansch in Z-Richtung

Gelenkachsen (Joints)

Konfiguration der Not-Halt-Rampen siehe Kapitel "Not-Halt-Rampen und Software-Endschalter der Gelenke" (→ 🗎 44).

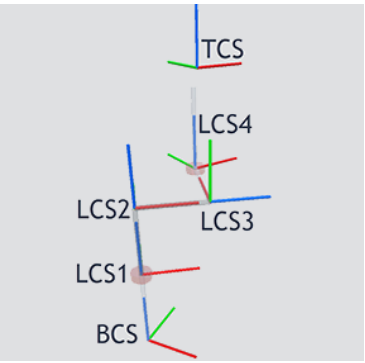
PTP-Parameter

Konfiguration der PTP-Parameter siehe Kapitel "PTP-Segmente" (→ 🗎 167).

Mögliche Konstellationen	**1, 2**
Über *JointPhase* parametrierbare Gelenkachsen	**1**

Physics

Die Physik-Simulation ist bei diesem Kinematikmodell aktivierbar.



33969008523

6.10 Bahnereignisse

Bahnereignisse ermöglichen das Ausführen einer Anweisung an einem parametrierbaren Punkt im Bahnfortschritt. Das Parametrieren der Bahnereignisse erfolgt über das Festlegen eines Referenzpunkts, das Verschieben des Referenzpunkts über den Parameter *Distance* und einen zeitlichen Versatz über den Parameter *Time*. Weitere Informationen dazu finden Sie im Kapitel "Bahnereignis" (→ 🗎 181).

Die Parametrierung des Bahnereignisses bezieht sich auf das nächste Bahnsegment im SRL-Programm und erfolgt in 3 Schritten:

1. Festlegen des Referenzpunkts: Anfangspunkt oder Endpunkt des nächsten Bahnsegments im SRL-Programm. Das SRL-Programm wird im RobotMonitor nur dann als fehlerfrei angezeigt, wenn auf die Registrierung REG PATH_EVENT ein Bewegungsbefehl folgt.
   - ⇨ wenn Reference = FALSE: Anfangspunkt des Bahnsegments
   - ⇨ wenn Reference = TRUE: Endpunkt des Bahnsegments
2. Verschieben des Referenzpunkts über den Parameter *Distance* in [mm]: Der Wert 0 führt zu keiner Verschiebung des Referenzpunkts. Ein negativer Wert führt zur Verschiebung des Referenzpunkts im Bahnfortschritt nach vorne, also zum Anfang der Bahn. Ein positiver Wert führt zur Verschiebung des Referenzpunkts im Bahnfortschritt nach hinten, also in Richtung Bahnende.
   - ⇨ wenn Distance = 0: genau im Bezugspunkt
   - ⇨ wenn Distance negativ: im Bahnfortschritt vor dem Bezugspunkt
   - ⇨ wenn Distance positiv: im Bahnfortschritt hinter dem Bezugspunkt
3. Zeitlicher Versatz über den Parameter *Time* in [ms]: Der Wert 0 führt zu keiner zeitlichen Verschiebung. Ein negativer Wert führt zur zeitlichen Verschiebung vor Erreichen des verschobenen Referenzpunkts, also zu einer Vorlaufzeit. Ein positiver Wert führt zur zeitlichen Verschiebung nach Erreichen des verschobenen Referenzpunkts, also zu einer Nachlaufzeit.
   - ⇨ wenn Time = 0: im verschobenen Bezugspunkt
   - ⇨ wenn Time negativ: vor Erreichen des verschobenen Bezugspunkts
   - ⇨ wenn Time positiv: nach Erreichen des verschobenen Bezugspunkts

HINWEIS

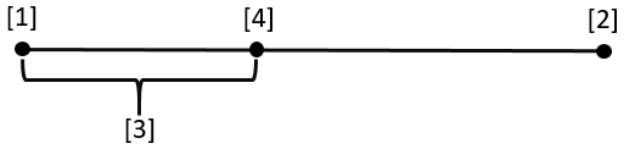
Sollen mehrere Anweisungen am gleichen Punkt im Bahnfortschritt ausgeführt werden, können Sie mehrere Bahnereignisse mit gleicher Parametrierung in das SRL-Programm einfügen.

Die in den folgenden Kapiteln erläuterten Beispiele veranschaulichen die Wirkweise der beschriebenen Parameter.

26873346/DE – 07/2021

6.10.1 Beispiel 1

Bahnereignis mit Referenzpunkt im Anfangspunkt des Segments, im Bahnfortschritt um einen positiven Wert des Parameters *Distance* nach hinten verschoben, Parameter *Time* = 0.

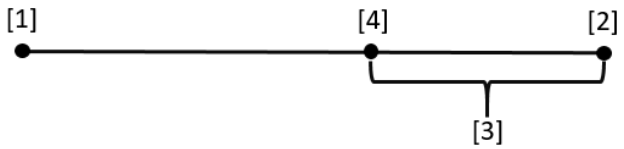


31386523275

[1] Anfangspunkt des Bahnsegments
[2] Endpunkt des Bahnsegments
[3] Strecke *Distance* in [mm], positiver Wert
[4] Punkt, an dem das Bahnereignis auslöst

6.10.2 Beispiel 2

Bahnereignis mit Referenzpunkt im Endpunkt des Segments, im Bahnfortschritt um einen negativen Wert des Parameters *Distance* nach vorne verschoben, Parameter *Time* = 0.



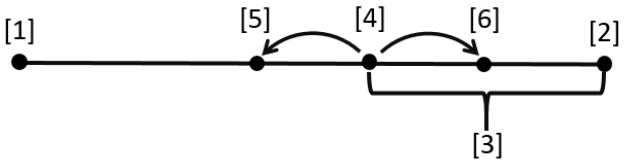
31386525707

[1] Anfangspunkt des Bahnsegments
[2] Endpunkt des Bahnsegments
[3] Strecke *Distance* in [mm], positiver Wert
[4] Punkt, an dem das Bahnereignis auslöst

26873346/DE – 07/2021

6.10.3 Beispiel 3

Bahnereignis mit Referenzpunkt im Endpunkt des Segments, im Bahnfortschritt um einen negativen Wert des Parameters *Distance* nach vorne verschoben und mit verschiedenen Werten des Parameters *Time* .



31386528139

[1] Anfangspunkt des Bahnsegments
[2] Endpunkt des Bahnsegments
[3] Strecke *Distance* in [mm], negativer Wert
[4] Punkt, an dem das Bahnereignis auslöst, wenn *Time* = 0
[5] Punkt, an dem das Bahnereignis auslöst, wenn *Time* < 0
[6] Punkt, an dem das Bahnereignis auslöst, wenn *Time* > 0

Die Verschiebung des Referenzpunkts sowie die zeitliche Verschiebung sind über mehrere Bahnsegmente möglich.

Die Verschiebungen im Bahnfortschritt nach vorne erfolgen maximal bis zu einem Rastpunkt. Rastpunkte sind der Anfangspunkt der Bahn und die Anfangspunkte von Bahnsegmenten, auf die nicht übergeschliffen wird.

Wenn der mittels des Parameters *Distance* verschobene Referenzpunkt vor dem letzten Rastpunkt liegt, wird die Fehlermeldung 16#7E82 "PathEventPositionBeforePathBegin" ausgegeben.

Wenn die parametrierte Vorlaufzeit (negativer Wert im Parameter *Time*) größer ist als die Bewegungszeit bis zum (mittels des Parameters *Distance*) verschobenen Referenzpunkt, kommt es zu folgendem Verhalten:

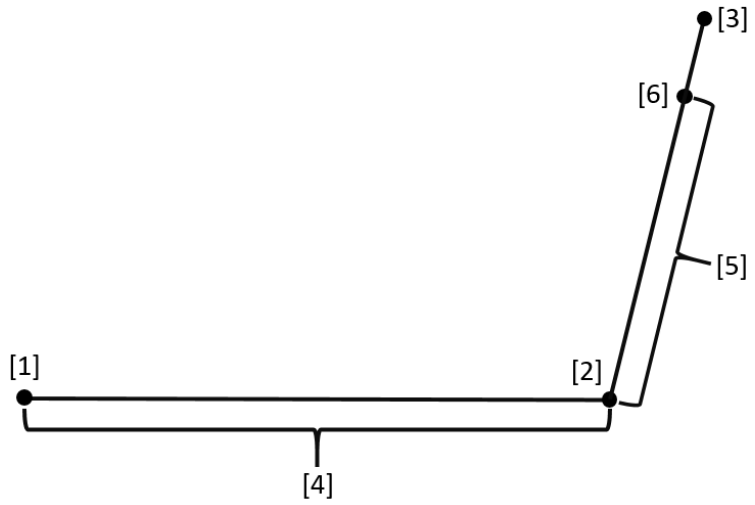
- Wenn sich der Roboter (im aktuellen Koordinatensystem) im Stillstand befindet, wird das Bahnereignis ausgelöst und die Vorlaufzeit beginnt abzulaufen. Der Roboter startet die Bewegung erst, wenn er sich nach Ablauf der parametrierten Vorlaufzeit am verschobenen Referenzpunkt befindet. Während der Wartezeit wird im RobotMonitor eine entsprechende Nachricht angezeigt.
- Wenn sich der Roboter (im aktuellen Koordinatensystem) in Bewegung befindet, wird die Fehlermeldung 16#7E83 "PathEventTimeShiftExceedsRemainingTimeInMotion" ausgegeben.

Die Verschiebungen im Bahnfortschritt nach hinten erfolgen auch über Rastpunkte hinweg, allerdings maximal bis zu einem Befehl, der das Bewegungsende erwartet: "WAIT MotionDone" oder "END_PROG WaitMotion".

26873346/DE – 07/2021

6.10.4 Beispiel 4

Bahnereignisse mit großem, positivem Wert des Parameters *Distance*, der zur Verschiebung des Referenzpunkts vom Anfangspunkt über die Segmentgrenze hinweg in das nächste Bahnsegment führt.



31386543371

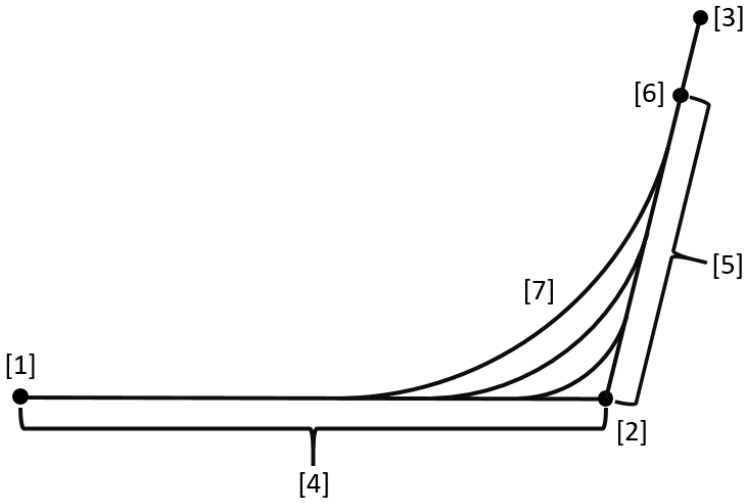
[1] Anfangspunkt des Bahnsegments, das im SRL-Programm als nächstes nach dem Befehl REG PATH_EVENT programmiert ist
[2] Endpunkt des Bahnsegments
[3] Endpunkt des darauf folgenden Bahnsegments
[4]+[5] Strecke *Distance* in [mm], positiver Wert
[6] Punkt, an dem das Bahnereignis auslöst

Der mittels des Parameters *Distance* verschobene Referenzpunkt liegt außerhalb des Überschleifbereichs immer an der gleichen Stelle. Entsprechend bezieht sich der Parameter *Distance* immer auf die Bahn ohne Verschliff, egal ob die Bahn verschliffen wird oder nicht.

26873346/DE – 07/2021

6.10.5 Beispiel 5

Bahnereignis wie in Beispiel 4, jedoch mit verschieden großen Überschleifbögen: Punkt [6], an dem das Bahnereignis auslöst, ist unabhängig vom Verschliff.

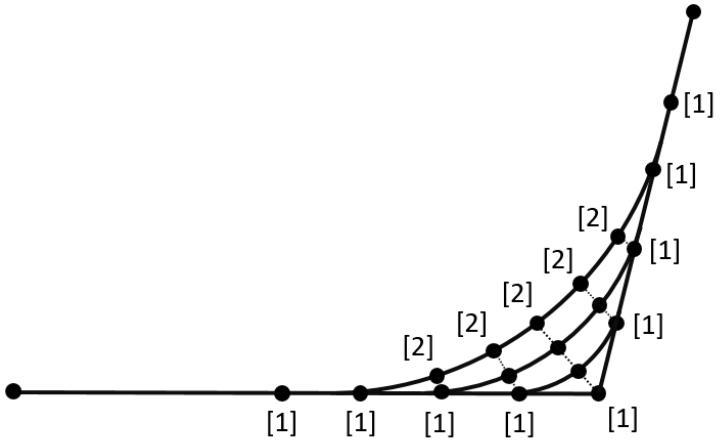


31386545803

[1] Anfangspunkt des Bahnsegments, das im SRL-Programm als nächstes nach dem Befehl REG PATH_EVENT programmiert ist
[2] Endpunkt des Bahnsegments
[3] Endpunkt des darauf folgenden Bahnsegments
[4]+[5] Strecke *Distance* in [mm], positiver Wert
[6] Punkt, an dem das Bahnereignis auslöst
[7] Mehrere, verschieden große Überschleifbögen

6.10.6 Beispiel 6

Referenzpunkte im Überschleifbereich



31386548235

[1] Mittels des Parameters *Distance* verschobene Referenzpunkte auf der Bahn ohne Verschliff, an denen das jeweilige Bahnereignis auslöst
[2] Auf den Überschleifbogen verschobene Referenzpunkte

26873346/DE – 07/2021

Die Anzahl der Bahnereignisse in einem SRL-Programm ist nicht begrenzt. Die maximale Anzahl der Bahnereignisse im Programmvorlauf beträgt 64. Wenn 64 Bahnereignisse im SRL-Programm registriert sind, schaltet der Programmzeiger erst dann weiter, wenn zuvor registrierte Bahnereignisse ausgelöst wurden und nicht mehr aktiv sind. Diese Situation wird mit einer entsprechenden Nachricht im RobotMonitor angezeigt.

Das SRL-Programm endet nicht, bevor nicht alle registrierten Bahnereignisse ausgelöst wurden. Wenn also der verschobene Referenzpunkt hinter dem Endpunkt des zuletzt interpretierten Bahnsegments liegt oder eine Nachlaufzeit noch nicht abgelaufen ist, endet das SRL-Programm nicht. In diesen Fällen wird im RobotMonitor eine entsprechende Nachricht angezeigt.

Wenn das SRL-Programm pausiert und/oder der Roboter von der Bahn im Tippbetrieb wegbewegt wird, lösen Bahnereignisse am Ende einer bereits gestarteten Nachlaufzeit noch aus. Diese Situation wird mit der Warnung 16#17E82 "PathEventTime-ShiftElapsedNotInProgramMode" im RobotMonitor angezeigt.

Beim Starten und beim Stoppen eines SRL-Programms werden alle bereits registrierten Bahnereignisse gelöscht.

Die Bahnereignisse beziehen sich auf den Setpoint Bahnfortschritt. Entsprechend kann ein Schleppfehler zu Ungenauigkeiten beim Auslösen von Bahnereignissen führen. Auch erhöhte Zykluszeiten der Task *HighPrio* führen zu verringerter Genauigkeit, da das Auslösen der Bahnereignisse in der Task *HighPrio* erfolgt.

26873346/DE – 07/2021

6.11 Touchprobe

Ein Touchprobe-Event (Ereignis) wird entweder durch das Triggern eines Sensors oder den Zustandswechsel einer BOOL-Variable ausgelöst.

Beim Auslösen des Touchprobe-Event werden die aktuellen Achspositionen ausgelesen und für eine Positionsmessung oder eine Restwegpositionierung verwendet. Die Touchprobe-Funktion des MOVIKIT® Robotics unterscheidet sich von der Touchprobe-Funktion der Einzelachsen durch die Transformation der Positionswerte der Einzelachsen in eine kartesische Touchprobe-Position und deren Abbildung auf die Bahn des Roboters.

Anwendungsfälle für die Touchprobe-Funktion sind z. B. das Palettieren oder Depalettieren bei variabler oder unbekannter Höhe der Teile oder die sensorbasierte Ausführung von Aktionen.

Die Aktionen, welche bei einem Touchprobe-Event ausführbar sind, können unterschieden werden in solche, die eine Bewegung auslösen (Positioning) und solche, die keine Bewegung auslösen (Measure, Set_Boolvar und CallFunction).

In einem SRL-Programm können beliebig viele Touchprobe-Events genutzt werden, jedoch nur ein Event zur selben Zeit. Um ein neues Event zu aktivieren, muss das vorherige Event deaktiviert sein, ansonsten wird ein Fehler ausgelöst.

Zum Verwenden der Touchprobe-Funktion muss diese im Konfigurationsmenü "Additional Functions" (→ 🗎 129) des Softwaremoduls aktiviert werden.

Weitere Informationen zum Parametrierung der SRL-Befehle im RobotMonitor finden Sie im Kapitel "Touchprobe" (→ 🗎 183).

6.11.1 Aktivierung/Deaktivierung

Die Touchprobe-Funktion ist nicht dauerhaft aktiviert. Sie kann z. B. bei Programmstart oder auch erst an einer bestimmten Stelle auf der Bahn des Roboters aktiviert werden. So lässt sich vermeiden, dass das Triggern des Sensors in einem falschen Bahnabschnitt erfolgt.

Ein Touchprobe-Event wird durch eine Registrierung mittels der Anweisung "REG_TOUCHPROBE_EVENT" im SRL-Programm aktiviert. Die Anweisung hat die im Folgenden beschriebenen Einstellmöglichkeiten:

- Über den Parameter *Source* wird die Triggerquelle eingestellt.
  - Bei der Einstellung "InverterTouchprobe" werden die Touchprobe-Positionen der Umrichter verwendet. Dafür muss ein Touchprobe-Sensor an allen Umrichtern des Roboters angeschlossen sein (außer einem MOVIKIT® MultiAxisController untergeordnete Achsen). Zudem müssen alle MultiMotion-Achsen des Roboters wie im Kapitel "Konfiguration der "MultiMotion-Touchprobe"" (→ 🗎 87) beschrieben konfiguriert sein. Über den Umrichter-Touchprobe kann die Triggerposition sehr genau (< 0.2 ms Registrierzeit) ermittelt werden.
  - Bei der Einstellung "BoolVariable" als *Source* muss der Digitale Eingang, an welchem der Sensor angeschlossen ist, im IEC-Programm auf eine SRL-Boolvariable gemappt werden. Zusätzliche Einstellungen in MOVISUITE® sind nicht nötig. Bei dieser Einstellung wird die Istposition vom Umrichter - nicht die Umrichter-Touchprobe-Position - verwendet. Sie wird im Zyklus der Task *HighPrio* (Power/Progressive: Zykluszeit ≥ 1 ms, Standard/Advanced: Zykluszeit ≥ 5 ms) gelesen.

    - Bei Verwendung einer simulierten Achse unterhalb des Roboters müssen Sie die Einstellung "BoolVariable" vornehmen.

26873346/DE – 07/2021

- Bei Verwendung des MOVIKIT® MultiAxisController oder bei einer virtuellen Achse unterhalb des Roboters wird bei der Einstellung "InverterTouchprobe" die Istposition vom Umrichter anstelle von Touchprobe-Positionen verwendet. Diese Achsenarten besitzen keine Touchprobe-Positionen.

- Bei der Einstellung "BoolVariable" als *Source* parametrieren Sie zusätzlich im Parameter *SourceBoolVar* den Index der Bool-Variable und im Parameter *Level*, auf welche Flanke getriggert werden soll.

- Über den Parameter *Mode* wird die Anzahl der zu erfassenden Trigger-Ereignisse bis zur Deregistrierung eingestellt. Mit "Single" wird ein Trigger erfasst. Mit der Einstellung "Multiple" werden bis zur Deregistrierung alle Trigger erfasst.
- Über den Parameter *MeasuringDirection* wird eingestellt, in welcher Richtung (X,Y oder Z) der Sensor einen Trigger auslöst. Wenn bei einer Palettieranwendung beispielsweise vertikal entlang Z die Höhe eines Stapels vermessen wird, stellen Sie Z ein. Zudem gibt der Parameter *MeasuringDirection* die Richtung an, auf welche sich die Distanz der Restwegpositionierung bezieht.

Die Touchprobe-Funktion kann über mehrere Bahnsegmente aktiviert sein. Sie ist nach der Aktivierung solange aktiviert, bis ...

- der (Touchprobe-)Sensor im *Mode* "Single" getriggert wird.
- die Funktion über die Anweisung "DEREG_TOUCHPROBE_EVENT deaktiviert wird.
- der Bewegungszeiger ("M" für "Motion") die Anweisung "CONTINUE AFTER POSITIONING EVENT" passiert.
- das SRL-Hauptprogramm zu Ende ist oder gestoppt wird. Wenn Unterprogramme beendet werden, bleibt die Touchprobe-Funktion aktiv und wird nicht automatisch deaktiviert.

Konfiguration der "MultiMotion-Touchprobe"

Zum Verwenden der Touchprobe-Positionen der Umrichter als Quelle (*Source* "InverterTouchprobe") für die Touchprobe-Funktion des MOVIKIT® Robotics muss die Touchprobe-Funktion aller Einzelachsen des Roboters entsprechend konfiguriert sein (außer einem MOVIKIT® MultiAxisController untergeordnete Achsen).

Das Aktivieren der Touchprobe-Funktion der Umrichter erfolgt über das Konfigurationsmenü "Grundeinstellungen" des MOVIKIT® MultiMotion unter "Verwendete Funktionen". Die Konfiguration des MOVIKIT® MultiMotion wird dann um das nachfolgend beschriebene Konfigurationsmenü erweitert. In der folgenden Tabelle finden Sie Hinweise zu Einstellungfeldern dieses Konfigurationsmenüs in Bezug auf das Verwenden der Funktion zusammen mit dem MOVIKIT® Robotics. In der Regel sind die Standardeinstellungen passend. Eine detailliertere Beschreibung des Konfigurationsmenüs finden Sie im Handbuch zum MOVIKIT® MultiMotion.

Parameter	Wert
Allgemein	
Touchprobe-Quelle	Hier als Touchprobe-Quelle "Umrichter" einstellen.
Modus	Die Auswahl des Touchprobe-Modus ist in diesem Fall irrelevant, da sie durch das SRL-Programm überschrieben wird.
Trigger	
Quelle	Quelle für das Auslösen des Triggers

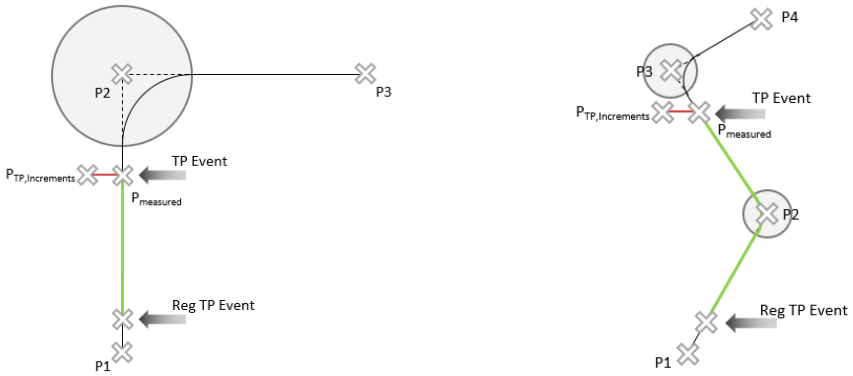
26873346/DE – 07/2021

Parameter	Wert
Ereignis	Auswahl, bei welcher Flankenform getriggert wird. Dieser Parameter ist nicht über das SRL-Programm einstellbar. • Steigende Flanke • Fallende Flanke • Steigende und fallende Flanke
Sensortotzeit steigende Flanke	Optionale Einstellung der Totzeit des verwendeten Sensors für steigende Flanke am Triggereingang. Diese Zeit wird bei der Berechnung des Touchprobe-Ereigniswerts eingerechnet.

6.11.2 Triggerung und Messung

Nach dem Auslösen eines Touchprobe-Events werden die Positionen ausgelesen, transformiert und auf die Bahn projiziert. Die ermittelte kartesische Istposition wird senkrecht zur *MeasuringDirection* auf die Bahn der Roboters projiziert. Bei Verwendung der Anweisung "MEASURE" wird der Projektionspunkt in der dort eingestellten Pose-Variable ausgegeben.

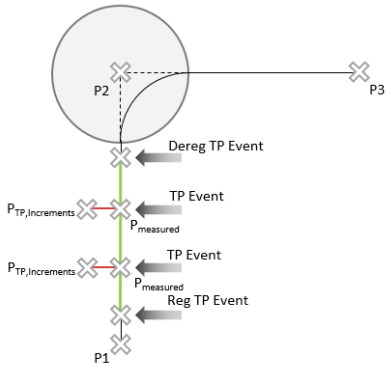
Hat das Bahnsegment, in welches projiziert werden soll, keine Bewegungskomponente in die *MeasuringDirection*, wird ein Fehler ausgegeben. Das ist beispielsweise der Fall, wenn als *MeasuringDirection* "Z" eingestellt ist, aber der Roboter entlang einer Geraden in X-Y-Richtung verfährt. Ist die Projektion auf das aktuelle Bewegungssegment räumlich nicht möglich, so wird versucht, den Punkt auf eines der benachbarten Segmente zu projizieren.



Bahnpunkt	Beschreibung
P1 .. P4	Wegpunkte auf dem Bahnverlauf
Reg TP Event	Anweisung zum Aktivieren der Touchprobe
TP Event	Triggern des Sensors oder Schalten der BOOL-Variable
$P_{TP,Increments}$	Istposition oder Touchprobe-Position der Umrichter
$P_{measured}$	Auf Bahn projizierter Messpunkt

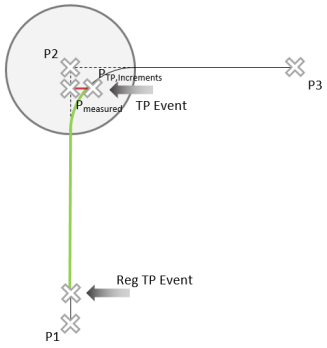
In den Abbildungen ist die Touchprobe im *Mode* "Single" mit *MeasuringDirection* vertikal in Z-Richtung dargestellt. Der grüne Bereich stellt die Segmente dar, in welchen die Touchprobe-Funktion aktiviert ist. Nach Auslösen der Triggers wird der gemessene Punkt ($P_{TP,Increments}$) auf die Bahn projiziert ($P_{measured}$). Die Bewegung des Roboters selbst bleibt dabei unbeeinflusst.

26873346/DE – 07/2021



Bahnpunkt	Beschreibung
P1 .. P3	Wegpunkte auf dem Bahnverlauf
Reg TP Event	Anweisung zum Aktivieren des Touchprobes
Dereg TP Event	Anweisung zum Deaktivieren des Touchprobes
TP Event	Triggern des Sensors oder Schalten der BOOL-Variable
$P_{TP,Increments}$	Istposition oder Touchprobe-Position der Umrichter
$P_{measured}$	Auf Bahn projizierter Messpunkt

Im *Mode* "Multiple" bleibt die Touchprobe-Funktion nach dem Auslösen eines Triggers aktiv (*MeasuringDirection* ist Z-Richtung). Erfolgt ein weiterer Trigger, wird der neu gemessene Punkt ebenfalls auf die Bahn projiziert, $P_{measured}$ wird überschrieben. Das Event ist solange aktiv (grüner Bereich), bis eine Anweisung "DEREG Touchprobe_EVENT" interpretiert wird, das Programm zu Ende ist oder gestoppt wird.



Bahnpunkt	Beschreibung
P1 .. P3	Wegpunkte auf dem Bahnverlauf
Reg TP Event	Anweisung zum Aktivieren des Touchprobes
TP Event	Triggern des Sensors oder Schalten der BOOL-Variable
$P_{TP,Increments}$	Istposition oder Touchprobe-Position der Umrichter
$P_{measured}$	Auf Bahn projezierter Messpunkt

Erfolgt das Auslösen des Touchprobe-Sensors innerhalb eines Überschleifbogens, wird der Punkt auf die Bahn projiziert, die ohne Überschleifen interpoliert wurde (Abbildung: *MeasuringDirection* Z-Richtung, *Mode* "Single").

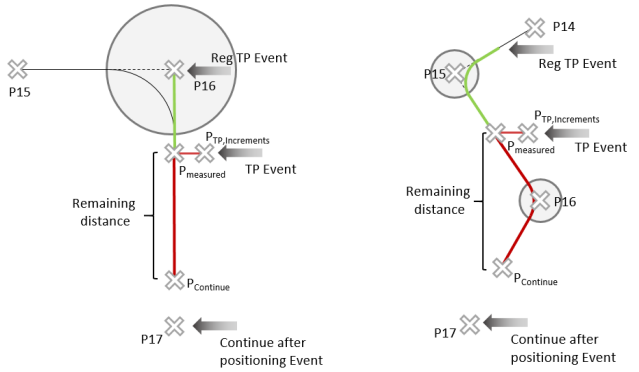
6.11.3 Restwegpositionierung

Um nach dem Auslösen eines Touchprobe-Events eine Restwegpositionierung einzuleiten, steht die Anweisung "POSITIONING" zur Verfügung. Im Parameter *RemainingDistance* stellen Sie die zurückzulegende Strecke in *MeasuringDirection* ein.

Um diese Anweisung zu nutzen, müssen Sie Folgendes beachten:

- Die "POSITIONING"-Anweisung ist für einen angegebenen Bereich aktiviert (Restwegfahrbereich). Um diesen Bereich zu kennzeichnen, muss er mit der Anweisung "CONTINUE AFTER POSITIONING EVENT" beschränkt werden.
- Der Restwegfahrbereich muss zum Berechnen der Restwegposition vollständig bekannt sein. Ist das Ende des Bereichs beim Auslösen der Restwegfahrt noch nicht bekannt, ist das Programm also noch nicht bis zu der Anweisung interpretiert und der Programmzeiger demnach noch nicht bei der Anweisung angekommen (z. B. wegen einer "WAIT"-Anweisung), wird ein Fehler ausgegeben.
- Die Anweisung "CONTINUE AFTER POSITIONING EVENT" muss nach dem Registrieren des Touchprobe-Events ausgeführt werden. Andernfalls wird ein Fehler ausgegeben.
- "POSITIONING" wird immer beim ersten Trigger des Events ausgeführt (*Mode* "Single"). Ist der *Mode* "Multiple" in Verbindung mit "POSITIONING" eingestellt, wird die Touchprobe-Funktion nach Starten der Restwegpositionierung trotzdem deaktiviert. Der Benutzer wird über eine Warnung darauf hingewiesen.
- Startpunkt derRestwegpositionierung ist der auf die Bahn projizierte Messpunkt.
- Der eingestellte Restweg (*RemainingDistance*) muss positiv sein.

Beim Ausführen des Events wird der Restweg berechnet. Die *RemainingDistance* gibt den Versatz an, welcher in Richtung *MeasuringDirection* zurückgelegt werden soll. Die Bahn selbst, also auch deren Richtung, wird durch die *MeasuringDirection* nicht verändert, d. h. der Roboter verfährt weiter auf der programmierten Bahn.



Bahnpunkt	Beschreibung
P14-P17	Wegpunkte auf dem Bahnverlauf
Reg TP Event	Anweisung zum Aktivieren der Touchprobe Funktion
CONTINUE AFTER POSITIONING EVENT	Marke, an welcher Stelle nach der "POSITIONING"-Anweisung die Bahn weitergeführt werden soll
TP Event	Triggern des Sensors oder Schalten der BOOL-Variable
$P_{TP,Increments}$	Istposition oder Touchprobe-Position der Umrichter
$P_{measured}$	Auf Bahn projizierter Messpunkt
$P_{Continue}$	Endpunkt der Restwegpositionierung

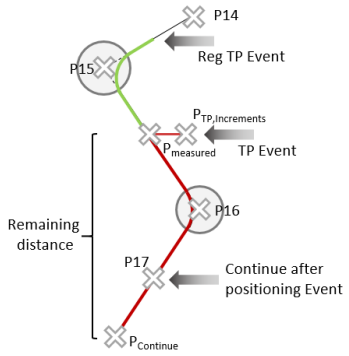
26873346/DE – 07/2021

Bahnpunkt	Beschreibung
Remaining Distance	Länge des Restwegs [mm] in Richtung der *MeasuringDirection*

Der Restweg ergibt sich entlang der Bahnsegmente des Restwegfahrbereichs (inkl. ihrer Orientierung), bis die *RemainingDistance* erreicht ist. Die rotatorischen Freiheitsgrade werden bis zum Erreichen des Restwegendes bahntreu interpoliert. Die Restwegpositionierung ist auch über mehrere Segmente hinweg möglich. Die Richtung der Segmente muss weniger als 90° von der *MeasuringDirection* abweichen, sodass eine Bewegungskomponente in *MeasuringDirection* vorhanden ist. Bei Ausführung der Restwegpositionierung werden alle Bewegungssegmente zwischen dem Restwegende und der "CONTINUE AFTER POSITIONING EVENT"-Anweisung verworfen.

Für die Restwegpositionierung werden die im Programm beim aktuellen Bahnsegment eingestellten Bewegungsparameter verwendet. Ist es mit diesen Parametern nicht möglich, den gewünschten Endpunkt einzuhalten (z. B. wegen zu hoher Geschwindigkeit beim Starten des Restwegs oder einem zu kurz gewählten Restweg), werden die Bewegungsparameter bis maximal zu den Not-Halt-Rampen erhöht. Der Benutzer wird über eine Warnung darüber informiert. Ist es trotz Not-Halt-Rampen nicht möglich den eingestellten Restweg einzuhalten, wird ein Fehler ausgegeben.

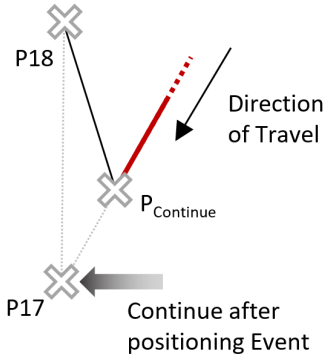
In der aktuellen Softwareversion wird die Bewegung während der Restwegpositionierung nicht beschleunigt. Somit kann es bei spätem Triggern kurz vor Erreichen des Stillstands zu einer sehr langsamen Restwegpositionierung kommen. In zukünftigen Versionen kann auch während der Restwegpositionierung beschleunigt werden.



Bahnpunkt	Beschreibung
P14-P17	Wegpunkte auf dem Bahnverlauf
Reg TP Event	Anweisung zum Aktivieren der Touchprobe Funktion
CONTINUE AFTER POSITIONING EVENT	Marke, an welcher Stelle nach der "POSITIONING"-Anweisung das Programm weitergeführt werden soll
TP Event	Triggern des Sensors oder Schalten der BOOL-Variable
$P_{TP,Increments}$	Istposition oder Touchprobe-Position der Umrichter
$P_{measured}$	Auf Bahn projizierter Messpunkt
$P_{Continue}$	Endpunkt der Restwegpositionierung

Bahnpunkt	Beschreibung
RemainingDistance	Länge des Restwegs [mm] in Richtung der *MeasuringDirection*

Ist der eingestellte Restweg länger als die eingespeisten Segmente bis zur "CONTINUE"-Marke, wird das letzte Bahnsegment verlängert. Hierbei gilt die Einschränkung, dass dieses Segment keine Rotation aufweist, also die rotatorischen Freiheitsgrade in dem Segment nicht verändert werden, ansonsten wird das Segment nicht verlängert und ein Fehler ausgegeben.



Bahnpunkt	Beschreibung
P17-P18	Wegpunkte auf dem Bahnverlauf
CONTINUE AFTER POSITIONING EVENT	Marke, an welcher Stelle nach der POSITIONING-Anweisung die Bahn weitergeführt werden soll
$P_{Continue}$	Endpunkt der Restwegpositionierung

Nach dem Durchführen einer "POSITIONING"-Anweisung können weiter Bewegungssegmente folgen. Diese Bewegungssegmente werden jedoch erst ausgeführt, nachdem die Positionierung beendet ist. Es wird als nächstes die Position nach der Anweisung "CONTINUE AFTER POSITIONING EVENT" angefahren.

6.11.4 Touchprobe als Bahnereignis

Die Anweisungen "REG_TOUCHPROBE_EVENT" und "DEREG_TOUCHPROBE_EVENT" können unterhalb einer Anweisung "REG PATH_EVENT", also bei einem Bahnereignis, platziert werden. So lässt sich die Touchprobe-Funktion auf einem bestimmten Bahnabschnitt aktivieren.

6.11.5 Gleichzeitige Verwendung von Bahnereignissen und Touchprobe

Bahnereignisse, deren Nachlaufzeit bereits abläuft, werden unabhängig von einer Touchprobe-Positionierung weiterhin ausgelöst. Sobald die Touchprobe-Positionierung jedoch gestartet ist, werden Bahnereignisse, deren verschobener Referenzpunkt (Parameter *Distance*) hinter dem Endpunkt der Positionierung liegt, nicht mehr ausgelöst. Somit kann es bei der Touchprobe-Positionierung vorkommen, dass Bahnereignisse nicht ausgelöst werden.

Wenn beispielsweise das Vakuum eines Sauggreifers vor Erreichen der Greifposition eingeschaltet werden soll, muss das Bahnereignis in einem Bereich parametriert werden, der vor dem Endpunkt der Touchprobe-Positionierung liegt. Es erfolgt keine automatische Verschiebung von Bahnereignissen mit dem Endpunkt der Positionierung.

26873346/DE – 07/2021

6.12 Statische Koordinatensysteme

Der Roboter fährt die im Programm vorgegebenen Bahnpunkte im Grundzustand im BASE-Koordinatensystem an. Darüber hinaus besteht die Möglichkeit, beliebig viele verschiedene, selbst definierte USER-Koordinatensysteme zu verwenden. Beispielsweise kann so die gleiche Bewegungsbahn an verschiedenen Plätzen in Bezug auf das jeweils eingestellte USER-Koordinatensystem ausgeführt werden. Des Weiteren ist es so möglich die Position und Verdrehung einer Palette zu vermessen und als Transformation des USER-Koordinatensystems zu verwenden, sodass eine Palettierung unabhängig von der Verschiebung in Bezug auf die Palette immer gleich ausgeführt wird.

Zum Verwenden der Funktion wird im SRL-Programm das Koordinatensystem "USER" eingestellt. Die nachfolgenden Bahnpunkte werden im USER-Koordinatensystem angefahren. Die Verwaltung der verschiedenen USER-Koordinatensysteme erfolgt in der aktuellen Version der Software im IEC-Programm. Die Umschaltung kann z. B. in einer *CallFunction* erfolgen. Weitere Informationen zur Verwendung finden Sie im Kapitel "Statische USER-Koordinatensysteme" (→ 🗎 223).

6.13 Conveyor Tracking und Rotary Table Tracking

Die Funktion "Conveyor Tracking" ermöglicht das Anfahren von Bahnpunkten mit dem Roboter in einem bewegten USER-Koordinatensystem. Das USER-Koordinatensystem kann sich dabei z. B. entlang eines Transportbands bewegen (Conveyor Tracking) oder mit einem Drehtisch um eine Achse drehen (Rotary Table Tracking). Für beide Ausprägungen wird die gleiche Funktion "Conveyor Tracking" verwendet und lediglich verschieden konfiguriert. Die Funktion wird gewöhnlich dazu verwendet, Teile von einem Transportband oder einem Drehtisch zu entnehmen. Hierzu wird im SRL-Programm das Koordinatensystem "USER" eingestellt.

Darüber hinaus besteht die Möglichkeit, einen Drehtisch zeitweise automatisch so auszurichten, dass er die Freiheitsgrade des Roboters erweitert. Beispielsweise kann sich ein Roboter so mit 3 Freiheitsgraden (Z, X, RotZ) im 4-dimensionalen Raum (X, Y, Z, RotZ) bewegen und somit eine 4-dimensionale Palettierung auf einer sich drehenden Palette ausführen. Der Roboter steuert hierzu die Bewegung der Transporteinrichtung und fährt die Bahnpunkte in dem von ihm gesteuerten USER-Koordinatensystem an. Hierzu wird im SRL-Programm das Koordinatensystem "USER_ControlledByRobot" eingestellt. Wenn "USER" oder "USER_ControlledByRobot" als aktuelles Koordinatensystem ausgewählt ist, fährt der Roboter die nachfolgenden Bahnpunkte in diesem Koordinatensystem an, bis entweder ein Wechsel in ein anderes USER-Koordinatensystem oder zurück zum BASE-Koordinatensystem erfolgt. Die Transformation, also die Lage des USER-Koordinatensystems/USER_ControlledByRobot-Koordinatensystems wird wie in den folgenden Kapiteln beschrieben im IEC-Programm eingestellt. In der aktuellen Version der Software liegen folgende Einschränkungen vor:

- Eine direkte Umschaltung zwischen zwei Transportbändern ist nur dann möglich, wenn diese dieselbe Orientierung haben (Transformations-Parameter *aIrOrigin[4..6]* sind identisch). Andernfalls muss dazwischen in das BASE-Koordinatensystem gewechselt werden.
- Ein Wechsel von einem Drehtisch in ein anderes USER-Koordinatensystem muss immer über das BASE-Koordinatensystem erfolgen.

Zum Verwenden der Funktion "Conveyor Tracking" muss diese im Konfigurationsmenü "Additional Functions" (→ 🗎 129) des Softwaremoduls aktiviert werden. Hierzu ist für jede Roboterinstanz je ein MOVIKIT® Robotics addon Conveyor Tracking erforderlich.

26873346/DE – 07/2021

Weitere Informationen zur Verwendung finden Sie in den Kapiteln "Synchronisierte Bewegung mit einem Transportband" (→ 🗎 224), "Synchronisierte Bewegung mit einem Drehtisch" (→ 🗎 227) und "Besonderheiten bei der Steuerung eines Drehtischs durch den Roboter" (→ 🗎 229).

6.13.1 Initialisierung

Durch die Initalisierung wird festgelegt, wo sich der Ursprung des bewegten USER-Koordinatensystems in Bezug auf das BASE-Koordinatensystem des Roboters befindet. Außerdem muss angegeben werden, um welchen Typ eines bewegten Koordinatensystems es sich handelt. Die Initialisierung kann z. B. im *INIT*-Teil des *USER_PRG* ausgeführt werden. Bei einer Förderbandanwendung muss der Typ "Linear", bei einer Drehtischanwendung der Typ "Rotary" angegeben werden.

Mit den Transformations-Parametern *alrOrigin[1..3]* wird die Verschiebung in X-Richtung, Y-Richtung und Z-Richtung in [mm] angegeben. Über die Transformations-Parameter *alrOrigin[4..6]* wird die sukzessive Verdrehung um die Z-Achse, die neue Y-Achse und neue X-Achse in [°] definiert. Die positive Bewegung eines USER-Koordinatensystems des Typs "Linear" erfolgt in positiver X-Richtung. Die positive Drehung eines USER-Koordinatensystems des Typs "Rotary" erfolgt in mathematisch positiver Richtung um die Z-Achse.

Die MultiMotion-Achse des Drehtischs ist bei Steuerung durch den Roboter mittels Einstellung des Koordinatensystems "USER_ControlledByRobot" als Modulo-Achse mit Verfahrbereich]-180°, 180°] einzustellen.

Zudem ist eine Anbindung des USER-Koordinatensystems an den Roboter und an die 3D-Simulation erforderlich. Weitere Informationen zur Verwendung finden Sie in den Kapiteln "Synchronisierte Bewegung mit einem Transportband" (→ 🗎 224), "Synchronisierte Bewegung mit einem Drehtisch" (→ 🗎 227) und "Besonderheiten bei der Steuerung eines Drehtischs durch den Roboter" (→ 🗎 229).

6.13.2 Positionsverlauf

Der Positionsverlauf des bewegten USER-Koordinatensystems muss stetig zugewiesen werden, bis eine Bewegung in einem anderen USER-Koordinatensystem oder im BASE-Koordinatensystem erfolgt. Die Zuweisung muss in der Task *HighPrio* (z. B. in der *HighPrio*-Aktion des *USER_PRG*) erfolgen.

Bevor in ein bewegtes Koordinatensystem gewechselt werden kann, muss die stetige Zuweisung seit mindestens 5 *HighPrio*-Zyklen erfolgen, damit ruckfrei auf die aktuelle Geschwindigkeit und Beschleunigung der Transporteinrichtung aufgesetzt werden kann.

6.13.3 Steuerung einer Transporteinrichtung durch den Roboter

Der Roboter richtet einen Drehtisch bei Einstellung "USER_ControlledByRobot" automatisch so aus, dass die Bahnpunkte im BASE-Koordinatensystem bei Y = "0" auf der dem Roboter zugewandten Seite angefahren werden. Applikativ muss darauf geachtet werden, dass der Mittelpunkt des Drehtischs durch die Bahn des Roboters im USER-Koordinatensystem nicht durchfahren wird, da sich der Drehtisch in dem Fall im Allgemeinen unendlich schnell drehen müsste. Je näher sich die Bahn am Mittelpunkt des Drehtischs befindet, umso schneller muss er sich im Allgemeinen drehen. Solche Probleme können beispielsweise durch eine Vorpositionierung des Drehtischs und Bahnpunkten in ausreichendem Abstand vom Mittelpunkt des Drehtischs vermieden werden.

26873346/DE – 07/2021

Ein Wechsel in das oder aus dem Koordinatensystem "USER_ControlledByRobot" ist bei Verwendung eines Roboters mit eingeschränkten Freiheitsgraden im Allgemeinen nur bei Stillstand des Drehtischs möglich, da ansonsten Ausgleichbewegungen erforderlich sind, denen der Roboter möglicherweise nicht folgen kann. Im Allgemeinen erfordert dies auch Stillstand des Roboters. Bei applikativ geschickter Wahl der Roboterbahnen und geeigneter Vorpositionierung des Drehtischs ist der Wechsel des Koordinatensystems jedoch auch bei Bewegung des Roboters möglich.

Die Funktion "BackToPath" ist in der aktuellen Version der Software in diesem Zusammenhang nicht nutzbar, da der Drehtisch noch nicht automatisch zurückpositioniert wird, um die Bahn nach der Unterbrechung stetig fortzusetzen. Wird das SRL-Programm während des Verfahrens im Koordinatensystem "USER_ControlledByRobot" unterbrochen, muss das aktuelle SRL-Programm gestoppt und ein neues Programm z. B. zum Freifahren gestartet werden.

6.14 Physiksimulation

Die Funktion "Physiksimulation" simuliert die Kraft-Momenten-Belastungen auf die Antriebe von Robotern. Dabei werden die Antriebsmomente (und abhängig vom Kinematikmodell zusätzlich Kippmomente und Querkräfte) zyklisch abhängig von Massen, Beschleunigungen und Stellungen der Armglieder berechnet. Diese Belastungen können vom MOVI-C® CONTROLLER in die SEW-Workbench importiert werden, um dort eine Antriebsauswahl für den Roboter zu treffen (Projektierung). Die Funktion "Physiksimulation" bietet den Vorteil einer realitätsgetreuen Simulation der Kraft-Momenten-Belastungen mit exakt den Bewegungsprofilen, die an der realen Anlage abgefahren werden.

Die Funktion "Physiksimulation" ist eine Stunde nutzbar. Danach muss ein "Reset" des MOVI-C® CONTROLLER erfolgen, um die Funktion erneut für eine Stunde nutzen zu können. Die Funktion "Physiksimulation" ist nur bei den Kinematikmodellen sichtbar, die eine Physiksimulation unterstützen.

Die Projektierung in der SEW-Workbench ist nur für SEW-Mitarbeiter freigeschaltet. Kontaktieren Sie SEW-EURODRIVE für eine Antriebsauslegung und wenn Sie andere als die im Kapitel "Kinematikmodelle" (→ 🗎 44) als unterstützt gekennzeichneten Modelle simulieren möchten.

Weitere Informationen finden Sie im Kapitel zum dazugehörigen "Konfigurationsmenü" (→ 🗎 126) und im Kapitel "IEC-Programmierung" (→ 🗎 187).

26873346/DE – 07/2021

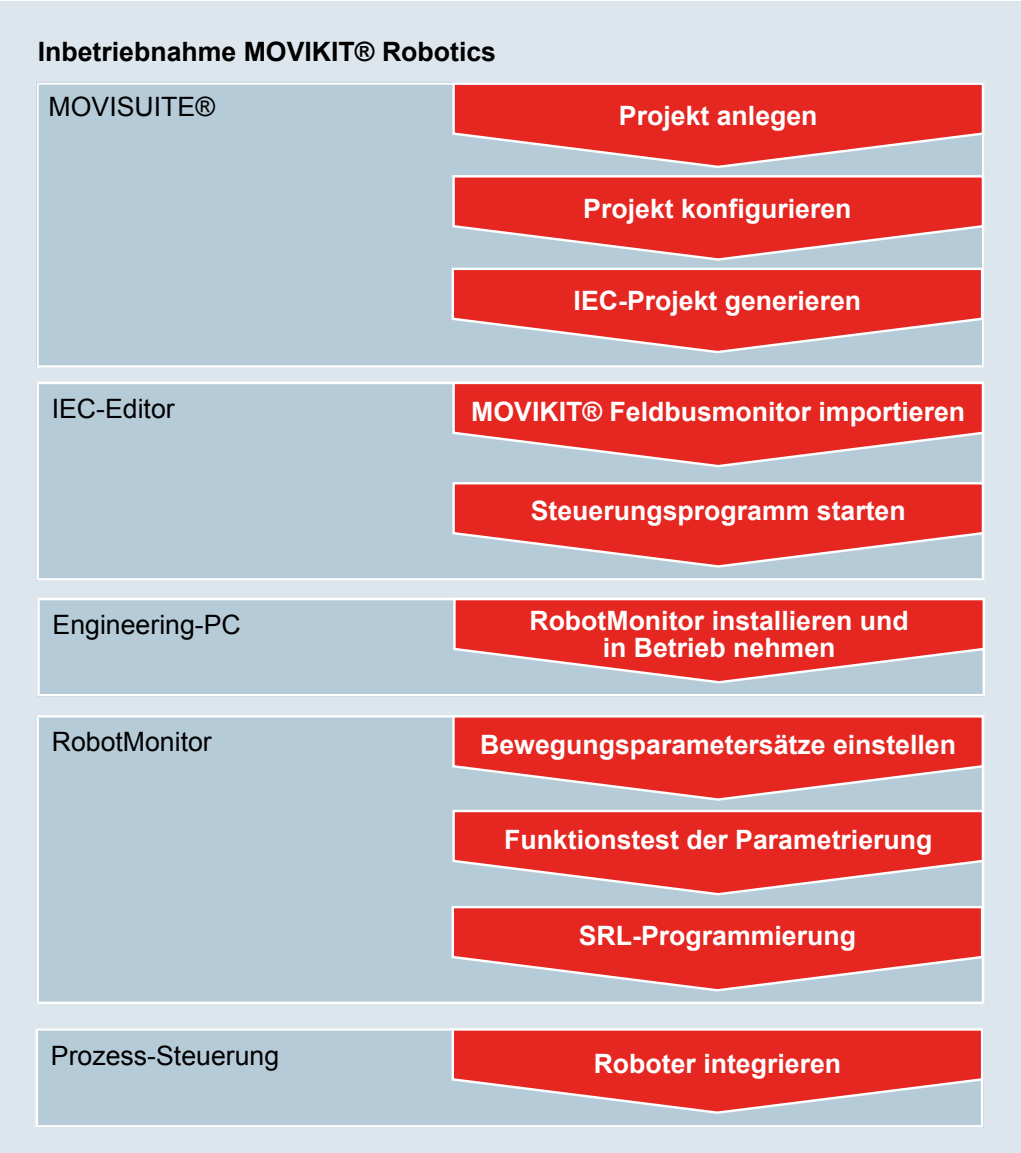
7 Inbetriebnahme

7.1 Voraussetzungen

- Beachten Sie die Installationshinweise in den Dokumentationen zu den verwendeten Geräten und Softwarekomponenten.
- In der MOVISUITE® werden die in Betrieb zu nehmenden Geräte angezeigt.

7.2 Inbetriebnahmeablauf

Folgendes Schaubild zeigt schematisch den Ablauf der Inbetriebnahme:



18014421790850187

In diesem Handbuch sind in den folgenden Kapiteln die für diese Software spezifischen Inbetriebnahmeschritte näher erläutert. Beachten Sie bei der Inbetriebnahme daher auch die Dokumentation aller weiteren verwendeten Komponenten.

26873346/DE – 07/2021

7.3 Projekt anlegen

HINWEIS

Detailliertere Informationen zur Bedienung der Engineering-Software MOVISUITE® finden Sie in der dazugehörigen Dokumentation.

✓ Ein neues MOVISUITE®-Projekt wurde erstellt und ist geöffnet.

1. Fügen Sie dem Projekt die benötigten Geräteknoten, Softwareknoten (MOVI-C® SoftwareNode) und Softwaremodule hinzu.

   ⇨ Siehe "Beispielprojekt".

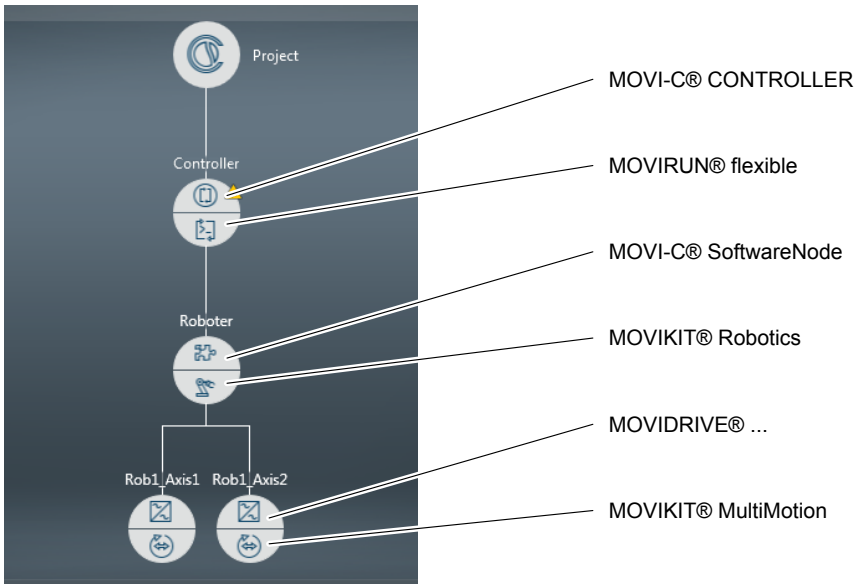
2. Konfigurieren Sie die hinzugefügten Geräte bzw. Softwaremodule. Beachten Sie dabei ggf. die für das MOVIKIT® Robotics spezifischen Erläuterungen in den nachfolgenden Kapiteln. Detaillierte Informationen zum Konfigurieren der Geräte bzw. anderer Softwaremodule finden Sie in der jeweils dazugehörigen Dokumentation.

7.3.1 Beispielprojekt

Folgende Abbildung zeigt ein entsprechendes Beispielprojekt:

HINWEIS

Dem MOVIKIT® Robotics untergeordnete Achsen benötigen das MOVIKIT® MultiMotion (und ggf. das MOVIKIT® MultiAxisController)



31222970507

7.4 MOVI-C® CONTROLLER konfigurieren

HINWEIS

Detailliertere Informationen zur Konfiguration des MOVI-C® CONTROLLER finden Sie in der dazugehörigen Dokumentation.

7.4.1 Zykluszeit einstellen

Stellen Sie durch Prüfen im Task-Manager sicher, dass es im Betrieb zu keiner Zykluszeitüberschreitung der Task *HighPrio* kommt und erhöhen Sie gegebenenfalls die eingestellte Zykluszeit auf den Achsen und auf dem MOVI-C® CONTROLLER. Spezifische Vorgaben hinsichtlich einzustellender Zykluszeiten finden Sie ggf. im Kapitel "Projektierungshinweise" (→ 🗎 15).

Das Einstellen der Zykluszeit erfordert folgende Teilschritte:

"Sollwertzyklus Steuerung" auf den Achsen einstellen

Führen Sie in MOVISUITE® folgende Schritte für alle untergeordneten Achsen durch:

1. Öffnen Sie die Konfiguration der Achse.
2. Öffnen Sie im Abschnitt "Funktionen" das Konfigurationsmenü "Sollwerte" und darin das Untermenü "Grundeinstellungen".
3. Stellen Sie im Bereich "Grundeinstellungen" im Eingabefeld "Sollwertzyklus Steuerung" den gewünschten Wert ein.

TaskHighPrio-Zykluszeit auf dem MOVI-C® CONTROLLER einstellen

Führen Sie in MOVISUITE® folgende Schritte für den MOVI-C® CONTROLLER durch:

4. Öffnen Sie die Konfiguration des MOVI-C® CONTROLLER.
5. Öffnen Sie im Abschnitt "MOVIRUN® flexible" das Konfigurationsmenü "Tasksystem".
6. Stellen Sie im Bereich "Tasksystem" im Eingabefeld "Zykluszeit der HighPrio Task" den gewünschten Wert ein.
7. Klicken Sie im Bereich "Tasksystem" im Eingabefeld "Sync Offset EtherCAT" zum Übernehmen des Vorschlagswerts auf den blauen Pfeil.

7.4.2 Feldbusanbindung einrichten

Führen Sie die folgenden Schritte durch, um am MOVI-C® CONTROLLER den Zugriff auf den Feldbus über IEC-Funktionsbausteine zu ermöglichen. Diese Einstellungen sind die Voraussetzung für die direkte Feldbusanbindung von Softwaremodulen.

✓ Ein MOVISUITE®-Projekt wurde erstellt und ist geöffnet.

✓ Das MOVISUITE®-Projekt beinhaltet einen MOVI-C® CONTROLLER.

1. Klicken Sie in der Funktionssicht in der MOVISUITE® auf den Knoten des MOVI-C® CONTROLLER.
   ⇨ Das Konfigurationsmenü des MOVI-C® CONTROLLER wird angezeigt.
2. Öffnen Sie im Konfigurationsmenü "MOVIRUN® flexible" das Untermenü "Feldbus".

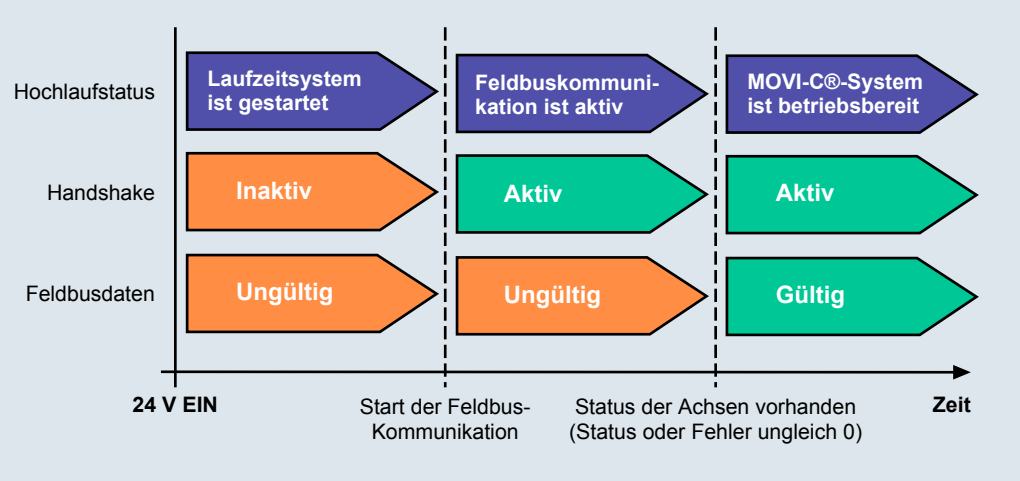
26873346/DE – 07/2021

3. Wählen Sie im Bereich "Feldbuskarte" das verwendete "Feldbusprotokoll" aus.
4. Setzen Sie im Bereich "Feldbusanbindung über IEC-Funktionsbausteine" den Wert des Felds "Feldbusanbindung aktivieren" auf "Ja".

⇨ Die Feldbusanbindung ist eingerichtet.

7.4.3 Hochlaufverhalten

Folgendes Diagramm veranschaulicht das Hochlaufverhalten des MOVI-C® CONTROLLER am Feldbus. Der Zeitraum von "24 V EIN" bis "Status der Achsen vorhanden" beträgt < 1 min.



9007232482850571

7.5 MOVIKIT® Robotics einfügen

HINWEIS

Detailliertere Informationen zur Bedienung der Engineering-Software MOVISUITE® finden Sie in der dazugehörigen Dokumentation.

✓ Ein MOVISUITE®-Projekt wurde angelegt und ist geöffnet.

1. Klicken Sie auf den leeren Softwaremodul-Bereich des gewünschten Knotens.
   ⇨ Der Katalog-Bereich klappt auf und die verfügbaren Softwaremodule werden angezeigt.
2. Klicken Sie im Katalog-Bereich auf MOVIKIT® Robotics.
   ⇨ Ein Kontextmenü wird geöffnet.
3. Wählen Sie im Kontextmenü über die entsprechende Auswahlliste die Version aus und bestätigen Sie Ihre Auswahl mit [Übernehmen].

⇨ Das MOVIKIT® Robotics wird dem Knoten zugeordnet, die Konfiguration angelegt und die Grundeinstellungen vorgenommen.

7.6 Untergeordnete Knoten konfigurieren

⚠ WARNUNG

Unvorhergesehen Bewegungen des Roboters durch das Konfigurieren einer Ruckbegrenzung an den Umrichtern

Körperverletzung und Tod

- Stellen Sie sicher, dass an den Umrichtern keine Ruckbegrenzung konfiguriert ist.

HINWEIS

Detailliertere Informationen zur Konfiguration der einzelnen Knoten finden Sie in der dazugehörigen Dokumentation.

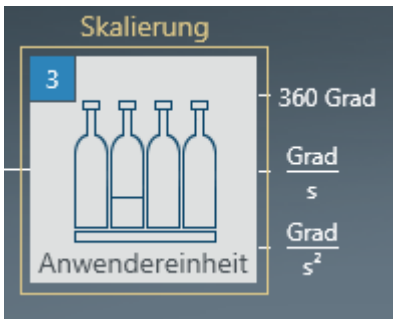
Beachten Sie für dem Softwaremodul untergeordnete Knoten folgende Hinweise:

- Folgende Achsgruppenteilnehmer dürfen unter dem Roboter verwendet werden:
  - MOVIDRIVE® mit MOVIKIT® MultiMotion (empfohlen) oder MOVIKIT® MulitMotion Camming
  - MOVI-C® Virtuelle Achse mit MOVIKIT® MultiMotion oder MOVIKIT® MulitMotion Camming
  - MOVI-C® SoftwareNode mit MOVIKIT® MultiAxisController
- Das Einstellen der Software- und Hardware-Endschalter muss vor der Antriebsstrangoptimierung erfolgen.

26873346/DE – 07/2021

Für den korrekten Betrieb des Softwaremoduls sind mindestens folgende Inbetriebnahme-Schritte für die untergeordneten Knoten notwendig. Beachten Sie dabei die jeweils dazugehörige Dokumentation.

1. Konfigurieren Sie den Antriebsstrang und definieren Sie dabei die Anwendereinheit. Siehe Kapitel "Standardeinheiten" (→ 🗎 28). Die Komponente "Anwendereinheit" muss zwingend im Antriebsstrang verwendet werden, um die korrekte Einstellung der Nachkommastellen zu gewährleisten.



9007230891832075

2. Bei den "Kinematikmodellen" (→ 🗎 44) ist angegeben, ob Linear- oder Drehachsen konfiguriert werden müssen. Bspw. müssen die Linearachsen eines kartesischen Portals im Antriebsstrang als Linearachsen konfiguriert werden. Die den umlaufenden Riemen antreibenden Achsen eines Roller Gantry müssen im Antriebsstrang jedoch als Drehachsen konfiguriert werden. Des Weiteren muss die Teilung sowie die Zähnezahl in der Kinematik-Konfiguration eingestellt werden
3. Abhängig vom mechanischen Einbau des jeweiligen Antriebs muss eine Drehrichtungsumkehr in der Parametergruppe "Regler" eingestellt werden. Die positive Bewegungsrichtung entnehmen Sie der Anzeige in der 3D-Simulation. Die Vorgehensweise hierzu ist im Kapitel "Achsen referenzieren und Funktionstest durchführen" (→ 🗎 110) erläutert.
4. Fügen Sie das Softwaremodul MOVIKIT® MultiMotion oder MOVIKIT® MultiAxisController ein und verwenden Sie für die Nachkommastellen die Grundeinstellung.

HINWEIS

Die Nachkommastellen der Position sind entscheidend für die maximale Position, die eine Achse erreichen kann. Wenn die Nachkommastellen der Position im Softwaremodul größer "0" sind, werden die Achspositionen für die interne Verarbeitung als Ganzzahl mit dem Faktor $10^{\text{Anzahl Nachkommastelllen}}$ multipliziert. Bei z. B. 4 Nachkommastellen und einer Sollposition der Achse von 10.0, wird der Wert für die interne Verarbeitung auf 100.000 skaliert und die größtmögliche erreichbare Position beträgt 214.748,3647. Der mögliche Bereich für die interne Verarbeitung geht von -2.147.483.648 bis 2.147.483.647. Bei größeren Positionen geht der Roboter in den Fehlerzustand.

HINWEIS

Wenn keine realen Achsen vorhanden sind, müssen Sie in MOVISUITE® in die Phase "Planung" wechseln, um das Softwaremodul MOVIKIT® MultiMotion einzufügen. Damit die Konfigurationsdaten auf den MOVI-C® CONTROLLER geladen werden, müssen Sie anschließend jedoch wieder in die Phase "Inbetriebnahme" wechseln.

5. Aktivieren Sie die vorhandenen Hardware-Endschalter (optional).

26873346/DE – 07/2021

6. Aktivieren und konfigurieren Sie die Software-Endschalter (Hierbei sollen die Software-Endschalter so gewählt werden, dass sie um den Bremsweg entfernt im Arbeitsraum vor den Hardware-Endschaltern liegen).
7. Konfigurieren Sie die Referenzfahrt.
8. Konfigurieren Sie die Not-Halt-Rampen. Setzen Sie dabei den Ruck auf "0".
9. Konfigurieren Sie die Drehmomentgrenze.
10. Optimieren Sie den "Regler".

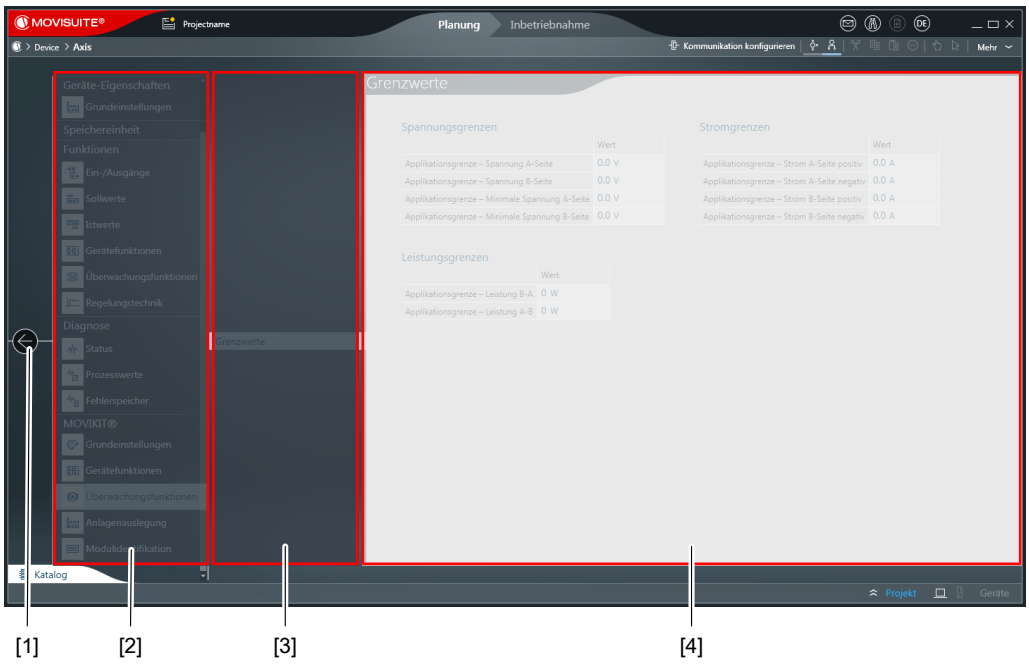
26873346/DE – 07/2021

7.7 MOVIKIT® Robotics konfigurieren

HINWEIS

Detailliertere Informationen zur Bedienung der Engineering-Software MOVISUITE® finden Sie in der dazugehörigen Dokumentation.

1. Klicken Sie in MOVISUITE® auf das MOVIKIT® Robotics.
   ⇨ Die Konfigurationsmenüs des Softwaremoduls werden angezeigt. Die Konfigurationsmenüs sind im Kapitel "Konfiguration" (→ 🗎 113) detailliert erläutert.



9007228165413771

[1] Schaltfläche zum Zurückkehren zur Projektübersicht
[2] Hauptmenü der Softwaremodul-Konfiguration (Abschnitt MOVIKIT®)
[3] Untermenüs der Konfiguration
[4] Einstellungsfelder der jeweiligen Untermenüs

2. Konfigurieren Sie das Softwaremodul über die entsprechenden Einstellungsfelder.

HINWEIS

Änderungen an der Konfiguration werden erst nach dem Aktualisieren der Konfigurationsdaten wirksam. Klicken Sie dazu in der entsprechenden Meldung am Knoten oder im Kontextmenü des MOVI-C® CONTROLLER auf [Konfigurationsdaten aktualisieren]. Für das Aktualisieren der Konfigurationsdaten wird der MOVI-C® CONTROLLER angehalten und neu gestartet.

3. Klicken Sie nach Abschluss der Konfiguration auf die Schaltfläche [1].
   ⇨ Die Projektübersicht wird angezeigt.

26873346/DE – 07/2021

7.7.1 Feldbusanbindung einrichten

Führen Sie zum Einrichten der Feldbusanbindung des Softwaremoduls folgende Schritte durch:

✓ Die "Konfiguration" (→ 🗎 113) des Softwaremoduls ist geöffnet.

1. Öffnen Sie das Konfigurationsmenü "Feldbus-Schnittstelle".
2. Setzen Sie im Bereich "Feldbuskonfiguration" den Wert des Feldes "Feldbusanbindung aktivieren" auf "Ja".
   ⇨ Weitere Einstellungsfelder werden angezeigt.
3. Passen Sie die Werte der weiteren Einstellungsfelder ggf. an die Gegebenheiten Ihres Systems an.

⇨ Die Feldbusanbindung ist eingerichtet und die resultierende Prozessdatenbelegung wird angezeigt.

26873346/DE – 07/2021

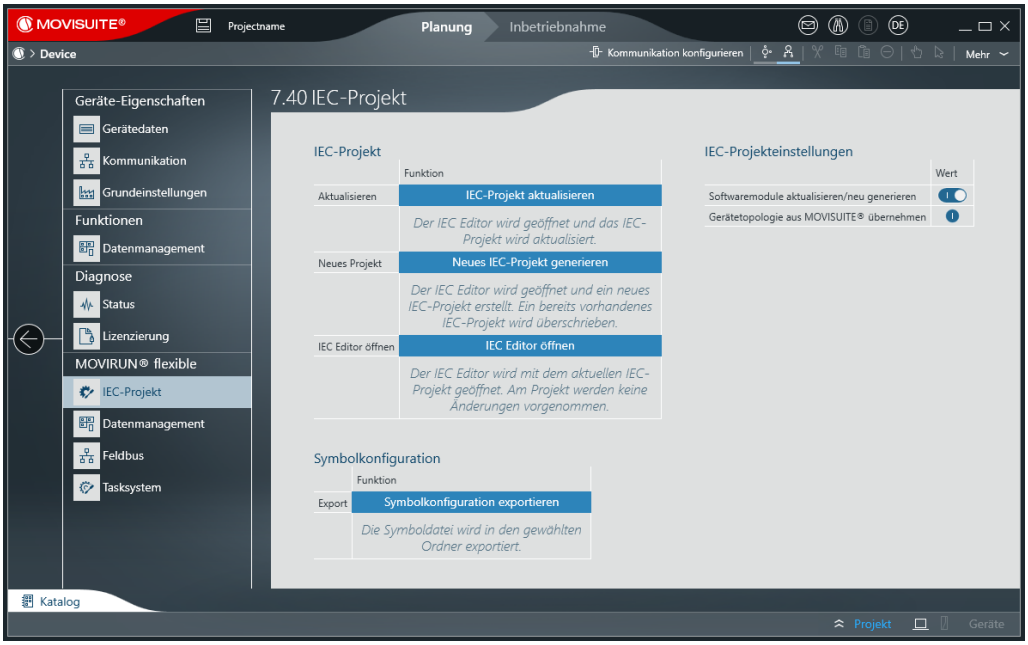
7.8 IEC-Projekt generieren

Führen Sie die folgenden Schritte durch, um mittels automatischer Codegenerierung ein IEC-Projekt basierend auf den Konfigurationen in der MOVISUITE® zu erstellen.

✓ Das Konfigurieren des MOVISUITE®-Projekts ist abgeschlossen.

1. Klicken Sie in der Funktionssicht in der MOVISUITE® auf den Softwaremodul-Bereich des MOVI-C® CONTROLLER.

   ⇨ Das Menü "IEC-Projekt" wird geöffnet.



36028817703378059

2. Klicken Sie auf [Neues IEC-Projekt generieren].

   ⇨ Der IEC-Editor wird geöffnet und ein neues IEC-Projekt wird erstellt.

HINWEIS

Werden nach dem erstmaligen Generieren des IEC-Projekts Änderungen an der Projekt-Struktur, in den Umrichterdatensätzen oder in der Softwaremodul-Konfiguration vorgenommen, wird ein Meldungssymbol am MOVI-C® CONTROLLER-Knoten angezeigt. Klicken Sie auf das Meldungssymbol, um nähere Informationen über die Änderung zu erhalten und eine Aktualisierung des IEC-Projekts durchzuführen.

26873346/DE – 07/2021

7.9 MOVIKIT® Feldbusmonitor importieren

Um den MOVIKIT® Feldbusmonitor zum Beobachten und Steuern der Feldbus-Schnittstelle zu nutzen, muss dieser importiert werden.

Öffnen Sie im IEC-Editor das Menü [Tools] > [Skripting] > [Scripts] > [F] und klicken Sie auf den Menüeintrag [Fieldbusmonitor.py]. Weitere Informationen zur Verwendung des MOVIKIT® Feldbusmonitors finden Sie im Kapitel "MOVIKIT® Feldbusmonitor" (→ 🗎 218).

7.10 Steuerungsprogramm starten

- ✓ Der IEC-Editor und das generierte Projekt sind geöffnet.
- ✓ MOVI-C® CONTROLLER und Engineering-PC sind verbunden.

1. Beheben Sie alle Fehler des Bibliotheksverwalters (siehe Meldungsfenster).
2. Beheben Sie alle Fehler der Vorkompilierung (siehe Meldungsfenster).
3. Klicken Sie im Menü [Erstellen] auf [Übersetzen], um das Projekt zu übersetzen.
4. Beheben Sie ggf. alle Fehler des Übersetzens (siehe Meldungsfenster).
5. Klicken Sie im Menü [Online] auf [Einloggen], um sich auf der Steuerung einzuloggen.
6. Klicken Sie im Menü [Debug] auf [Start], um die Steuerung zu starten.
7. Wechseln Sie zur MOVISUITE®, öffnen Sie im Modus "Inbetriebnahme" durch Klicken auf eine freie Stelle in der Funktionssicht das Kontextmenü und wählen Sie den Menüeintrag [Geräteadressen aktualisieren].

26873346/DE – 07/2021

7.11 RobotMonitor installieren

Für den Betrieb des MOVIKIT® Robotics ist die Installation des RobotMonitor notwendig. Führen Sie dazu folgende Schritte durch:

1. Rufen Sie die SEW-Homepage (www.sew-eurodrive.de) auf und öffnen Sie dort die Seite für den Softwaredownload (Online Support / Daten & Dokumente / Software)
2. Geben Sie in das Suchfeld den Begriff "RobotMonitor" ein und klicken Sie auf [Suchen].
3. Markieren Sie den Eintrag "RobotMonitor" in der Ergebnisliste und klicken Sie auf [Auswahl als ZIP herunterladen].
4. Entpacken Sie die heruntergeladene ZIP-Datei.
5. Starten Sie Im Ordner mit den entpackten Dateien die *.exe und befolgen Sie die Installationshinweise.

7.11.1 Systemvoraussetzungen

Folgende Hardware wird vorausgesetzt:

- Prozessor: 1 GHz 32-Bit- (x86) oder 64-Bit-Prozessor
- Arbeitsspeicher: 4 GB RAM (empfohlen)
- Festplatte: 40 MB freier Speicher insgesamt
- Grafikkarte: 2 GB Grafikspeicher

Der RobotMonitor ist eine 32-Bit-Anwendung.

Sie benötigen eines der folgenden Betriebssysteme:

- Windows 7 min. SP1 32 Bit / 64 Bit

  oder

  Windows 8.1 32 Bit / 64 Bit

  oder

  Windows 10 32 Bit / 64 Bit

Beachten Sie beim Betrieb folgende Empfehlungen hinsichtlich der Auflösung:

- Die beste Darstellung wird mit einer Auflösung von 1280 x 800 px oder höher erreicht
- Die Verwendung von großen Schriftarten und niedrigen Bildschirmauflösungen kann dazu führen, dass Informationen auf dem Display nicht angezeigt werden.

7.12 RobotMonitor starten

✓ Das Steuerungsprogramm mit MOVIKIT® Robotics ist auf dem MOVI-C® CONTROLLER gestartet ("RUN").

✓ Zwischen dem MOVI-C® CONTROLLER und dem Engineering-PC besteht eine Ethernet-Verbindung.

1. Öffnen Sie die Anwendungsdatei (RobotMonitor.exe) des Robotermonitors. Weitere Informationen zur Installation des Robotermonitors finden Sie im Kapitel "RobotMonitor installieren" (→ 🗎 107).

26873346/DE – 07/2021

7.13 Verbindung zum MOVI-C® CONTROLLER aufbauen

HINWEIS

Für den Fall das keine Verbindung aufgebaut werden kann, finden Sie im Kapitel "Kommunikation RobotMonitor/MOVI-C® CONTROLLER nicht möglich" (→ 🗎 238) Informationen zur Problembehebung.

1. Wählen Sie die IP-Adresse aus: Standard-IP-Adresse des MOVI-C® CONTROLLER (192.168.10.4) oder eine benutzerspezifische IP-Adresse.

9007224167506187

2. Klicken Sie auf die Schaltfläche [Start Communication].
   ⇨ Die Verbindung ist aufgebaut und meldet an drei Stellen im RobotMonitor den grün hinterlegten Status "Comm. OK".

⇨ Der RobotMonitor kommuniziert erfolgreich mit dem Controller.

7.14 Benutzer verwalten (optional)

Zur Verwaltung der Zugriffsberechtigungen verschiedener Benutzer besitzt der RobotMonitor eine Benutzerverwaltung. In der Benutzerverwaltung können beispielsweise Funktionen wie das Ändern von SRL-Programmen für bestimmte Benutzer erlaubt oder gesperrt werden. Siehe Kapitel "Benutzerverwaltung" (→ 🗎 139).

7.15 Bewegungsparametersätze einstellen

Sowohl für den Tippbetrieb als auch für die Bahn-Interpolation (z. B. für Geraden) von Gelenkachsen, kartesischen Achsen und Einzelachsen des Kinematikmodells, müssen vorab die Bewegungsparameter eingestellt werden.

Grundlegende Erläuterungen zu diesem Thema finden Sie in den Kapiteln "Bewegungsprofile" (→ 🗎 27), "Bewegungsparametersätze" (→ 🗎 28) und "Bewegungsparameter einstellen" (→ 🗎 235). Es gelten dabei die festgelegten "Standardeinheiten" (→ 🗎 28).

Tippbetrieb

Die Bewegungsparameter für das Tippen der Gelenkachsen und der Einzelachsen befinden sich unter "MotionSet.Kinematic.Joint". Die Bewegungsparameter für das Tippen der translatorischen und rotatorischen kartesischen Achsen befinden sich unter "MotionSet.Translation" und "MotionSet.Rotation".

Der Standardwert für den Tippbetrieb ist Bewegungsparametersatz "8". Der Bewegungsparametersatz für den Tippbetrieb kann über die "Anwenderschnittstelle im IEC-Programm" (→ 🗎 188) eingestellt werden. Dies ist vor allen dann wichtig, wenn die Gelenkachsen des Roboters nicht mit den Einzelachsen übereinstimmen. In diesen Fall sollte ein eigener Bewegungsparametersatz für die Einzelachsen angelegt (z. B. in Bewegungsparametersatz 7) und der Standartwert entsprechend angepasst werden.

26873346/DE – 07/2021

In der aktuellen Softwareversion muss das Verhältnis von Ruck zu Beschleunigung bzw. Verzögerung für die Erhöhung des resultierenden Rucks im Tippbetrieb vergrößert werden. In zukünftigen Versionen wird der resultierende Ruck dem eingestellten Wert entsprechen.

Automatik-Betrieb

Die Bewegungsparameter für die Bahninterpolation befinden sich unter "MotionSet.Translation" und "MotionSet.Rotation".

In der aktuellen Softwareversion wird mit der parametrierten Beschleunigung im Automatik-Betrieb auch verzögert. In zukünftigen Versionen wird die eingestellte Verzögerung auch im Automatik-Betrieb verwendet. Wenn z. B. stark beschleunigt und weich verzögert werden soll, ist das in der aktuellen Softwareversion durch Verwendung verschiedener Bewegungsparametersätze für die Bahnsegmente möglich.

Vorgehensweise

- ✓ Der RobotMonitor ist gestartet und verbunden.
- ✓ Die Zugriffsberechtigung ist angefordert.

1. Wechseln Sie zum Register "Variables".
2. Wechseln Sie im Register "Variables" zum Register "MotionSet Var".
   - ⇨ Die 8 Bewegungsparametersätze werden angezeigt.
3. Klappen Sie die Details des gewünschten Bewegungsparametersatzes ("MotionSet") durch Anklicken des Pfeils auf.
4. Klappen Sie unter "MotionSet" die Parametergruppe "Kinematic" auf.
5. Klappen Sie unter "Kinematic" die Parametergruppen der verwendeten Gelenkachsen ("Joint") auf.
6. Stellen Sie jeweils für die kinematische Gelenkachsen die Bewegungsparameter Geschwindigkeit, Beschleunigung, Verzögerung und Ruck ein.
7. Klappen Sie unter "MotionSet" die Parametergruppen "Translation" und "Rotation" auf.
8. Stellen Sie für die Translation und die Rotation jeweils die Bewegungsparameter Geschwindigkeit, Beschleunigung, Verzögerung und Ruck ein.
9. Klicken Sie zum Bestätigen Ihrer Einstellungen auf die gelb markierte Schaltfläche [Send].

HINWEIS

Die Bewegungsparametersätze sind zunächst im flüchtigen Speicher zwischengespeichert. Klicken Sie auf [Speichern], um die Bewegungsparametersätze dauerhaft auf der Speicherkarte zu speichern.

26873346/DE – 07/2021

7.16 Achsen referenzieren und Funktionstest durchführen

HINWEIS

Beachten Sie beim Verwenden eines Kinematikmodells des Typs ROLLER GANTRY die Besonderheit beim Referenzieren entsprechend der Beschreibung beim jeweiligen Kinematikmodell im Kapitel "Kinematic Model" (→ 🗎 113).

7.16.1 Funktionstest des simulierten Roboters

✓ Es liegt kein Fehler vor.

1. Klicken sie auf die Schaltfläche [Simulate Inverters].
   ⇨ Unten links auf der Benutzeroberfläche des RobotMonitors erscheint die Rückmeldung *Inverters connected*.
2. Führen Sie den Hochlauf mit dem RobotMonitor durch, indem Sie die Schritte aus dem Kapitel "Prozess-Start" (→ 🗎 133) durchführen.
3. Konfigurieren Sie die Bewegungsparametersätze.
4. Tippen Sie jede Achse des Roboters mit JOG_JOINT (+ und - Schaltflächen) in die Nullstellungen der Gelenkachsen. Die Werte der Gelenkachsen werden im RobotMonitor im Fenster der 3D-Simulation links oben angezeigt. Merken Sie sich die Stellung der Gelenkachsen in der Nullstellung sowie die positive Bewegungsrichtung der Gelenkachsen, wie sie in der 3D-Simulation angezeigt werden. Sie müssen die realen Achsen entsprechend referenzieren und auch die Drehrichtung der realen Achsen entsprechend einstellen. Siehe dazu "Untergeordnete Knoten konfigurieren" (→ 🗎 100).
5. Tippen Sie den Roboter mit *JOG_JOINT* (+ und - Schaltflächen) an die Arbeitsraumgrenzen und überprüfen Sie diese.
6. Tippen Sie den Roboter mit *JOG_CART* (+ und - Schaltflächen) an die Arbeitsraumgrenzen und überprüfen Sie diese.
7. Erstellen Sie folgendes Testprogramm

```
P   N10   ROB.MOTIONSET :=  1
    N20   LIN Grundstellung  BlendingDist :=  0
    N30   END_PROG
```

8. Starten Sie das Testprogramm und warten Sie bis das Programmende erreicht ist.
9. Ändern Sie mehrmals die Position der Grundstellung und testen Sie das Programm erneut.

7.16.2 Referenzieren und Funktionstest am realen Roboter

Führen Sie einen Funktionstest des Roboters durch, indem sie folgende Schritte durchführen:

1. Prüfen Sie die positive Bewegungsrichtung aller Achsen über den Handbetrieb der MOVISUITE® gemäß der Anzeige in der 3D-Simulation aus Schritt 4 in Kapitel "Funktionstest des simulierten Roboters" (→ 🗎 110). Falls die positive Bewegungsrichtung einer realen Achse nicht mit der positiven Bewegungsrichtung in der 3D-Simulation übereinstimmt, konfigurieren Sie eine Drehrichtungsumkehr in MOVISUITE®. Siehe dazu "Untergeordnete Knoten konfigurieren" (→ 🗎 100).

26873346/DE – 07/2021

2. Führen Sie den Hochlauf mit dem RobotMonitor durch, indem Sie die Schritte aus dem Kapitel "Prozess-Start" (→ 🗎 133) durchführen.
3. Referenzieren Sie die Achsen über die in MOVIKIT® Robotics integrierte Referenzierfunktion gemäß der Nullstellung, die Sie in Schritt 4 im Kapitel "Funktionstest des simulierten Roboters" (→ 🗎 110) ermittelt haben. Wechseln Sie dafür im RobotMonitor in den Referenzierbetrieb, indem Sie rechts im Dropdown-Menü unter "JOG CART", "JOG JOINT" und "JOG AXIS" den Referenzierbetrieb HOMING auswählen. Sie sehen nun, ob die jeweilige Achse bereits referenziert ist. Ist die Achse nicht oder falsch referenziert, drücken und halten Sie den Start-Button der jeweiligen Achse, um die Achse zu referenzieren. Die Achsen, die nicht referenziert werden sollen, wechseln für die Dauer des Referenzierens in den Not-Halt.

HINWEIS

Das Referenzieren von virtuellen Achsen ist mit dem Softwaremodul nicht möglich. Virtuelle Achsen melden dauerhaft, dass sie nicht referenziert sind. Simulierte Achsen hingegen sind immer referenziert. Reale Achsen sind während des Referenzierens nicht referenziert. Das Referenzieren ist auch im UI und über PD möglich (siehe Kapitel "IEC-Programmierung" (→ 🗎 187) bzw. "Prozessdatenbelegung" (→ 🗎 204)). Außerdem ist das Referenzieren der Achsen über den Handbetrieb in MOVISUITE® und oder über das MOVIKIT® MultiMotion (Zugriff anfordern, Referenzfahrt aktivieren und starten, nach erfolgreicher Ausführung Zugriff zurückgeben) möglich.

4. Tippen Sie jede Achse des Roboters mit JOG_JOINT (+ und - Schaltflächen) und vergleichen Sie die reale Bewegung mit der in der 3D-Simulation dargestellten Bewegung. Falls Sie eine Abweichung der Nullstellung oder der Bewegungsrichtung erkennen, führen Sie für die jeweilige Achse nochmals Schritt 1 oder Schritt 2 durch.
5. Tippen Sie jede Achse des Roboters mit JOG_JOINT (+ und - Schaltflächen) um eine bestimmte Strecke bei Linearachsen bzw. Winkel bei Drehachsen, z. B. 100 mm bzw. 90°. Prüfen Sie, ob die jeweilige Achse des Roboters genau diese Strecke bzw. diesen Winkel zurücklegt. Wenn dies nicht der Fall ist, korrigieren Sie bitte die Einstellungen im Antriebsstrang der Achse in MOVISUITE®.
6. Tippen Sie anschließend den Roboter mit *JOG_JOINT* (+ und - Schaltflächen) an die Arbeitsraumgrenzen und überprüfen Sie diese.
7. Tippen Sie anschließend den Roboter mit *JOG_CART* (+ und - Schaltflächen) an die Arbeitsraumgrenzen und überprüfen Sie diese.
8. Erstellen Sie folgendes Testprogramm

```
P   N10   ROB.MOTIONSET :=  1
    N20   LIN Grundstellung  BlendingDist :=  0
    N30   END_PROG
```

9. Starten Sie das Testprogramm und warten Sie bis das Programmende erreicht ist.
10. Ändern Sie mehrmals die Position der Grundstellung und testen Sie das Programm erneut. Die Bewegungen des realen Roboters müssen mit den in der 3D-Simulation dargestellten Bewegungen übereinstimmen.
11. Erzeugen Sie eine IEC-Bootapplikation.

26873346/DE – 07/2021

7.17 Roboter programmieren

Nachdem der Roboter den Funktionstest bestanden hat, kann das Programmieren der Bewegungsbahnen des Roboters erfolgen. Erstellen Sie hierzu mit dem RobotMonitor SRL-Programme. Detailierte Informationen dazu erhalten Sie im Kapitel "SRL-Programmierung" (→ 🗎 137).

7.18 Roboter integrieren

Nachdem die SRL-Programme mit dem RobotMonitor erstellt wurden, müssen diese von einer Prozess-Steuerung (RobotMonitor, MOVI-C® CONTROLLER oder übergeordnete Steuerung) gestartet werden.

Detailierte Informationen dazu erhalten Sie im Kapitel "Ansteuerung durch die Prozess-Steuerung" (→ 🗎 130).

26873346/DE – 07/2021

8 Konfiguration

8.1 Kinematic Model

8.1.1 Model Selection

Parameterbezeichnung	Beschreibung
Kinematic Model	
Model	Auswahl des Kinematikmodells

HINWEIS

Abhängig vom gewählten Kinematikmodell werden weitere Untermenüs mit spezifischen Einstellungsmöglichkeiten angezeigt. Detailliertere Informationen zur Konfiguration der Kinematikmodelle und zu den darin enthaltenen Parametern finden Sie im Kapitel "Kinematikmodelle" (→ 🗎 44).

8.2 Path Emergency Dynamics

⚠ WARNUNG

Unkontrollierte Bewegungen des Roboters durch Ruck-Einstellung am Umrichter

Sachschäden und Tod

- Deaktivieren Sie den Ruck für den Not-Halt der Umrichter (auf 0 setzen)

HINWEIS

Diese Parameter sind auch die Obergrenzen für die Verzögerung und den Ruck der Roboterbewegung im Tipp- und Programmbetrieb.

Für den Not-Halt auf der Bahn müssen die Bewegungsparameter definiert werden. Dabei wird nicht nach kartesischem Freiheitsgrad sondern zwischen Translation und Rotation unterschieden. Siehe auch "Bewegungsparametersätze" (→ 🗎 28) und "Bewegungsparameter einstellen" (→ 🗎 235).

Parameterbezeichnung	Beschreibung
Path Emergency Dynamics	
Trans EStop Dec	Bremsverzögerung, mit der die Translation beim Not-Halt auf der Bahn abgebremst wird.
Trans EStop Jerk	Ruck, mit der die translatorische Bremsverzögerung beim Not-Halt aufgebaut wird.
Rot EStop Dec	Bremsverzögerung, mit der die Rotation beim Not-Halt auf der Bahn abgebremst wird.
Rot EStop Jerk	Ruck, mit der die rotatorische Bremsverzögerung beim Not-Halt aufgebaut wird.

Die Not-Halt-Rampen müssen so gewählt werden, dass sie...

- jederzeit von den Umrichtern des Roboters durchgeführt werden können, ohne dass diese einen Fehler melden (Schleppfehler, Drehzahlüberwachung, ...).
- die Mechanik nicht beschädigen.

26873346/DE – 07/2021

- keine unerwünschten Schwingungen und Geräusche beim Bremsen verursachen.
- den maximal erlaubten Bremsweg einhalten.

8.3 Cartesian Limits

Parameterbezeichnung	Beschreibung
Cartesian SWLS	
Kartesische Software-Endschalter Weitere Informationen finden Sie im Kapitel "Software-Endschalter" (→ 🗎 43).	

8.4 Transformations

Parameterbezeichnung	Beschreibung
Tool Transformation	
Koordinatentransformation vom Flanschkoordinatensystem zum Werkzeugkoordinatensystem Das Flanschkoordinatensystem ist dabei das kartesisches Koordinatensystem, das sich aus dem Basiskoordinatensystem mithilfe der direkten kinematischen Transformation ergibt. Mittels dieser Transformation können Versätze und Verdrehungen durch das Werkzeug des Roboters berücksichtigt werden, die nicht bereits im Kinematikmodell berücksichtigt sind. Diese Transformation kann im Betrieb nicht mehr verändert werden.	

26873346/DE – 07/2021

8.5 Axis-Joint Transformations

8.5.1 Couplings Tabelle

Konfigurationsmenü zum Konfigurieren von Kopplungsfaktoren zwischen Einzelachsen (Axis) und Gelenkachsen (Joint). Der Kopplungsfaktor gibt dabei an, wie viel Einfluss eine Einzelachse auf eine Gelenkachse hat, die ihr nicht direkt zugeordnet ist, d. h. nicht den gleichen Index hat. Wenn keine Kopplung vorliegt, genügt es, die Anzahl der Kopplungen auf 0 einzustellen. Wenn man diese größer als 0 einstellt, wird eine Tabelle mit einer Zeile pro Kopplung eingeblendet. Für jede Kopplung kann man den Eingangsindex, den Ausgangsindex und den Koppelfaktor einstellen.

Beispiel

Dreht sich die Einzelachse 4 um 20 °, würde sich bei einer Kopplung mit Eingangsindex 4, Ausgangsindex 5 und einem (nur theoretisch möglichen) Koppelfaktor von 0 nur die Gelenkachse 4 um 20 ° drehen. Ist der Koppelfaktor jedoch 1, werden sowohl die Gelenkachse 4 als auch die Gelenkachse 5 um 20 ° gedreht. Gelenkachse 5 wird zusätzlich um den Wert von Einzelachse 5 gedreht. Kopplungen zwischen den Achsen werden z. B. mittels Parallelogrammen oder Riemenkonstruktionen realisiert. Auch die Handachsen typischer ARTICULATED Knickarmroboter sind in der Regel gekoppelt.

Parametrierung

Parameter	Beschreibung
Couplings	
Number of Couplings	Anzahl der Kopplungen Bei einer Einstellung größer 0, wird die Kopplungsfaktorentabelle mit den Spalten "Faktor", "Eingangsindex" und "Ausgangsindex" eingeblendet.
Eingangsindex	Einzelachse, von der die Kopplung den Positionswert nehmen soll.
Ausgangsindex	Gelenkachse, zu der die Kopplung den Positionswert der Einzelachse (mit dem Faktor gewichteten) addieren soll.
Faktor	Faktor, mit dem der Positionswert der Einzelachse gewichtet wird, bevor er dann auf die Gelenkachse addiert wird.

26873346/DE – 07/2021

8.5.2 Cranks Tabelle

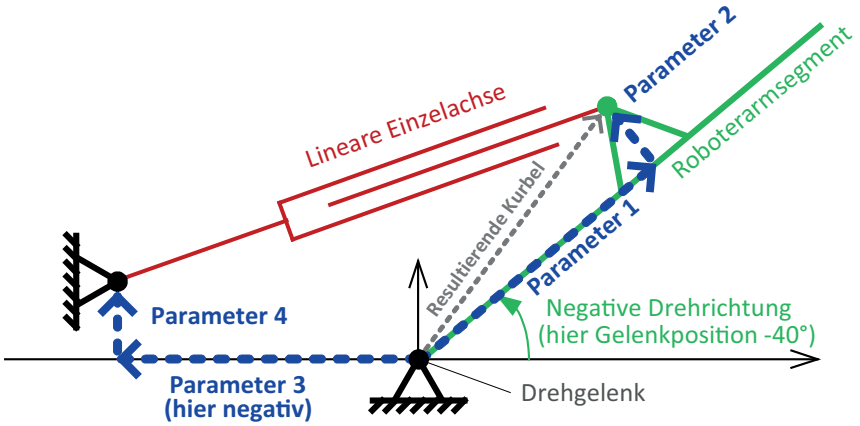
Cranks (Kurbeln) werden auf vielfältige Arten zur Übertragung von Bewegungen verwendet. In diesem Konfigurationsmenü kann die Übertragung der Bewegung einer Einzelachse (Axis) auf die zugehörige Gelenkachse (Joint) parametriert werden.

Wenn die Anzahl der Cranks auf 0 eingestellt wird, entsprechen alle Einzelachsen direkt den Gelenkachsen mit dem gleichen Index. In diesem Fall ist einer linearen Einzelachse mit Index j also eine lineare Gelenkachse mit dem gleichen Index j zugeordnet. Entsprechend ist einer rotativen Einzelachse mit Index k eine rotative Gelenkachse mit Index k zugeordnet.

Wird die Anzahl der Cranks auf einen Wert n größer 0 eingestellt, können n Cranks parametriert werden. Hierzu wird eine Tabelle mit je einer Zeile pro Crank eingeblendet. Für jeden Crank können "Index", "Crank-Typ" und die für den Typ erforderlichen Parameter eingestellt werden.

Linear axis to rotary joint

Bei diesem Crank-Typ wird die Bewegung einer linearen Einzelachse in die Bewegung einer rotativen Gelenkachse übertragen.



9007228304930315

Beispiel:

Im abgebildeten Beispiel sind Parameter 1, 2 und 4 positiv, Parameter 3 ist negativ. Da Parameter 2 positiv ist, bewegt sich die lineare Einzelachse in der Abbildung oberhalb der rotativen Gelenkachse. Wäre Parameter 2 negativ, würde sich die lineare Einzelachse unterhalb der rotativen Gelenkachse bewegen. Der Roboterarm hat in dem Beispiel den Gelenkachswert -40°. Zeigt der Roboterarm z. B. horizontal nach rechts, hat er den Gelenkachswert 0°. Zeigt der Roboterarm z. B. senkrecht nach oben, hat er den Gelenkachswert -90°. Die lineare Einzelachse müsste also zum Anfahren von 0° gegenüber der Stellung in der Abbildung verlängert und zum Anfahren von -90° verkürzt werden.

Parametrierung

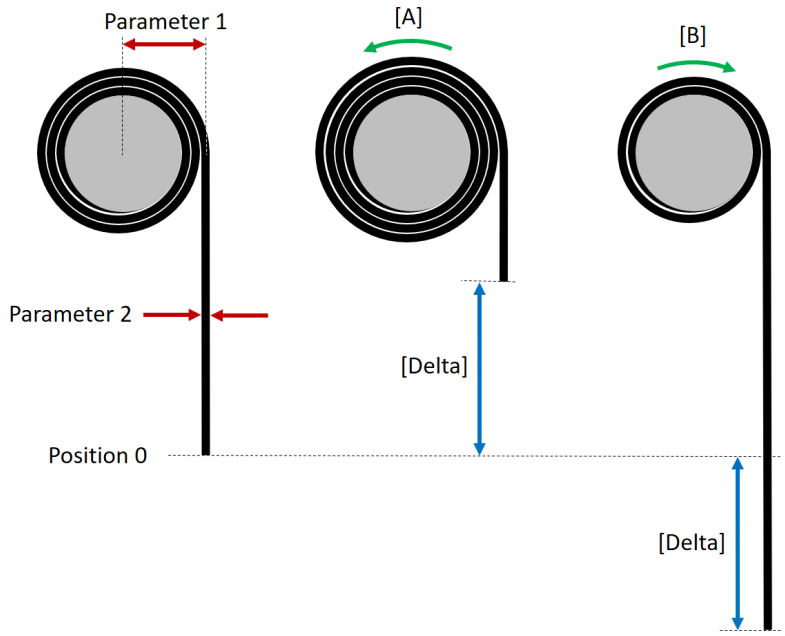
Parameter	Beschreibung
Cranks	
Index	Index von Einzelachse und Gelenkachse, zwischen denen der Crank wirkt.
Parameter 1	Versatz, ausgehend von der rotativen Gelenkachse, entlang des Roboterarms, der um die rotative Gelenkachse gedreht wird, hin zum Scharniergelenk zwischen linearer Einzelachse und Roboterarm (Parameter 1 > 0)

26873346/DE – 07/2021

Parameter	Beschreibung
Parameter 2	Zusätzlicher Versatz, senkrecht zum Roboterarm, in Bewegungsebene des Roboterarms, hin zum Scharniergelenk zwischen linearer Einzelachse und Roboterarm. Abhängig vom Vorzeichen von Parameter 2 bewegt sich die lineare Einzelachse auf der einen oder der anderen Seite der rotativen Gelenkachse.
Parameter 3	Versatz, ausgehend von der rotativen Gelenkachse, entlang der Achse des Roboterarms bei Gelenkachswert 0°, hin zum Scharniergelenk zwischen der Verankerung und der linearen Einzelachse. (Parameter 3 < 0)
Parameter 4	Zusätzlicher Versatz, senkrecht zur Achse von Parameter 3, in Bewegungsebene des Roboterarms, hin zum Scharniergelenk zwischen der Verankerung und der linearen Einzelachse.
Parameter 5..8	nicht verwendet

Winding axis to linear joint

Bei diesem Crank-Typ wird die Bewegung einer rotativen Einzelachse in die Bewegung einer linearen Gelenkachse übertragen. Insbesondere wird z. B. ein Riemen übereinander auf die Einzelachse aufgewickelt, so dass sich der effektive Durchmesser der rotativen Einzelachse während der Drehung verändert.



Beispiel

Im abgebildeten Beispiel ist links die Nullstellung der Achse dargestellt, mit Parameter 1 als effektivem Radius in dieser Stellung und Parameter 2 als Dicke des Wickelmaterials.

In Fall [A] nimmt der effektive Radius in positiver Drehrichtung (grüner Pfeil) zu. Die positive Bewegungsrichtung der linearen Gelenkachse zeigt in dem Fall in der Abbildung nach oben.

In Fall [B] nimmt der effektive Radius in positiver Drehrichtung (grüner Pfeil) ab. Die positive Bewegungsrichtung der linearen Gelenkachse zeigt in dem Fall in der Abbildung nach unten.

26873346/DE – 07/2021

Parametrierung

Parameter	Beschreibung
Cranks	
Index	Index von Einzelachse und Gelenkachse, zwischen denen der Crank wirkt.
Parameter 1	**Nur verwendet, wenn Parameter 8 = 0 oder 1** Effektiver Radius der rotativen Einzelachse in der Nullstellung der Achse (Parameter 1 > 0)
Parameter 2	**Nur verwendet, wenn Parameter 8 = 0** Dicke des gewickelten Materials • Parameter 2 > 0: Fall [A], wobei der effektive Radius in positiver Drehrichtung zunimmt • Parameter 2 = 0: Effektiver Radius ändert sich nicht • Parameter 2 < 0: Fall [B], wobei der effektive Radius in positiver Drehrichtung abnimmt. Die Materialdicke ist hierbei der Betrag von Parameter 2.
Parameter 3..7	Partiell verwendet, wenn Parameter 8 ≠ 0 (siehe Beschreibung "Inbetriebnahme")
Parameter 8	Kennzeichnung der Eingabe von Parametern (siehe Beschreibung "Inbetriebnahme")

Inbetriebnahme

Die rotative Einzelachse muss im Antriebsstrang mit der Einheit Grad [°] konfiguriert werden (obwohl eine lineare Gelenkachse des ausgewählten Kinematikmodells angetrieben wird). Die Umrechnung von der rotativen Einzelachse in eine lineare Gelenkachse erfolgt durch den Crank.

Die Software-Endschalter müssen so eingestellt sein, dass das Wickelmaterial nicht weiter abgewickelt wird als mechanisch möglich. Für die korrekte Funktion der Software-Endschalter der Gelenkachsen und der kartesischen Achsen im Softwaremodul, müssen die Parameter gemäß folgender Beschreibung korrekt eingestellt sein.

Wenn die erforderlichen Parameter 1 und 2 bekannt sind, werden diese einfach eingegeben. Hierzu muss in Parameter 8 der Wert 0 eingegeben werden.

Wenn jedoch Unsicherheiten bestehen und eine hohe Genauigkeit erzielt werden soll (insbesondere zur Berücksichtigung der Längung und Verringerung der Dicke des Wickelmaterials unter Last), lässt sich der Crank gemäß der im Folgenden beschriebenen Vorgehensweisen auf verschiedene Arten vermessen. In allen Varianten können Sie die effektiv verwendeten Parameter 1 (effektiver Radius in Nullstellung) und Parameter 2 (Dicke des gewickelten Materials) zu Diagnosezwecken nach Aktualisierung der Konfigurationsdaten und Start des MOVI-C® CONTROLLER im IEC-Editor auslesen:

```
SEW_GVL_Internal.MyRobot.fbAxisJointTransformation.
stOut.stConfig.astCrank[Index].alrPar[1 bzw. 2]
```

26873346/DE – 07/2021

Vermessung bei Einstellung Parameter 8 = 1 (Nullstellung anfahrbar)

Wenn die Nullstellung in der Nähe des negativen Endes des Verfahrbereichs festgelegt ist, bietet sich folgende Vorgehensweise an.

Bei dieser Vorgehensweise werden neben Parameter 8 (Wert = 1) 3 Parameter konfiguriert. Parameter 1 wird dabei explizit eingegeben. Parameter 2 wird beim Starten des MOVI-C® CONTROLLER automatisch berechnet.

1. Verfahren Sie die Achse z. B. mittels JOG_AXIS in die Nullstellung, die möglichst nah am negativen Ende des Verfahrbereichs der Achse festgelegt ist.
2. Markieren Sie die Position der linearen Gelenkachse im Raum in dieser Stellung.
3. Messen Sie den effektiven Wickelradius in dieser Stellung mit einem Maßband
   ⇨ Parameter 1
4. Verfahren Sie die Achse möglichst weit bis zum positiven Ende des Verfahrbereichs der Achse.
5. Lesen Sie die neue "Achsposition" aus:
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Axis`**`[Index]`
   ⇨ Parameter 3
6. Messen Sie die (positive) Strecke "Delta" (blauer Pfeil in Abbildung) zwischen der zuvor angebrachten Markierung und der neuen Position der linearen Gelenkachse mit einem Maßband
   ⇨ Parameter 4
7. Geben Sie die ermittelten Werte für Parameter 1, 3 und 4 ein sowie den Wert 1 für Parameter 8 und aktualisieren Sie die Konfiguration.
   ⇨ Der Crank ist korrekt parametriert, wenn der Gelenkachswert in der Position, in der "Delta" gemessen wurde, mit "Delta" übereinstimmt:
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Joint`**`[Index]`

Alternative Vorgehensweise bei Einstellung Parameter 8 = 1

Wenn die Nullstellung in der Nähe des **positiven** Endes des Verfahrbereichs der Achse liegt, ergeben sich zur beschriebenen Vorgehensweise die folgenden Unterschiede. Die anderen Schritte sind identisch.

Schritt 1: Verfahren Sie die Achse in die Nullstellung in der Nähe des **positiven** Endes des Verfahrbereichs der Achse.

Schritt 4: Verfahren Sie die Achse möglichst weit bis zum **negativen** Ende des Verfahrbereichs der Achse.

Schritt 5: Messen Sie die (negative) Strecke "Delta" und geben Sie den Wert als **negative** Zahl in Parameter 3 ein.

Vermessung bei Einstellung Parameter 8 = 2 (Nullstellung mittig oder nicht anfahrbar)

Wenn die Nullstellung etwa mittig im Verfahrbereich liegt, sodass die Strecke für die Vermessung verkürzt wäre oder die Nullstellung außerhalb des Verfahrbereichs liegt, bietet sich die folgende Vorgehensweise an.

26873346/DE – 07/2021

Bei dieser Vorgehensweise werden neben Parameter 8 (Wert = 1) 4 Parameter konfiguriert. Parameter 1 und Parameter 2 werden beim Starten des MOVI-C® CONTROLLER automatisch berechnet.

1. Verfahren Sie die Achse z. B. mittels JOG_AXIS möglichst weit bis zum negativen Ende des Verfahrbereichs der Achse.
2. Lesen Sie die Position der Achse aus:
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Axis`**`[Index]`
   ⇨ Parameter 3
3. Markieren Sie die Position der linearen Gelenkachse im Raum in dieser Stellung.
4. Messen Sie den effektiven Wickelradius "Radius" in dieser Stellung mit einem Maßband
   ⇨ Parameter 4
5. Verfahren Sie die Achse nun möglichst weit bis zum positiven Ende des Verfahrbereichs der Achse.
6. Lesen Sie die neue Position der Achse aus:
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Axis`**`[Index]`
   ⇨ Parameter 5
7. Messen Sie die (positive) Strecke "Delta" zwischen der zuvor angebrachten Markierung und der neuen Position der linearen Gelenkachse mit einem Maßband
   ⇨ Parameter 6
8. Geben Sie die ermittelten Werte für Parameter 3 bis 6 sowie den Wert "2" für Parameter 8 ein und aktualisieren Sie die Konfiguration.
9. Die korrekte Parametrierung des Cranks ist plausibel, wenn die Differenz der Gelenkachswerte in den beiden Messpunkten dem Wert "Delta" entspricht.

Alternative Vorgehensweise bei Einstellung Parameter 8 = 2

Wenn der effektive Wickelradius im Gegensatz zur "Vermessung bei Einstellung Parameter 8 = 2" am **positiven** Ende des Verfahrbereichs der Achse gemessen werden soll, ergeben sich zur beschriebenen "Vermessung bei Einstellung Parameter 8 = 2" die folgenden Unterschiede. Die anderen Schritte sind identisch.

Schritt 1: Verfahren Sie die Achse möglichst weit bis zum **positiven** Ende des Verfahrbereichs der Achse.

Schritt 5: Verfahren Sie die Achse möglichst weit bis zum **negativen** Ende des Verfahrbereichs der Achse.

Schritt 6: Messen Sie die (negative) Strecke "Delta" und geben Sie den Wert als **negative** Zahl in Parameter 6 ein.

Vermessung bei Einstellung Parameter 8 = 3 (Wickelradius nicht messbar)

Wenn der effektive Wickelradius nicht messbar ist, z. B. weil mechanisch nicht zugänglich, bietet sich die folgende Vorgehensweise an.

26873346/DE – 07/2021

Bei dieser Vorgehensweise werden neben Parameter 8 (Wert = 1) 5 Parameter konfiguriert. Parameter 1 und Parameter 2 werden beim Starten des MOVI-C® CONTROLLER automatisch berechnet.

1. Verfahren Sie die Achse z. B. mittels JOG_AXIS möglichst weit bis zum negativen Ende des Verfahrbereichs der Achse.
2. Lesen Sie die Position der Achse aus.
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Axis`**`[Index]`
   ⇨ Parameter 3
3. Markieren Sie die Position der linearen Gelenkachse im Raum in dieser Stellung.
4. Verfahren Sie die Achse nun etwa bis zur Mitte des Verfahrbereichs der Achse.
5. Lesen Sie die neue Position der Achse aus.
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Axis`**`[Index]`
   ⇨ Parameter 4
6. Messen Sie die (positive) Strecke "Delta 1" zwischen der zuvor angebrachten Markierung und der neuen Position der linearen Gelenkachse mit einem Maßband
   ⇨ Parameter 5
7. Verfahren Sie die Achse nun möglichst weit bis zum positiven Ende des Verfahrbereichs der Achse.
8. Lesen Sie die neue Position der Achse aus.
   ⇨ `SEW_GVL.Interface_MyRobot.Basic.OUT.SetpointPose.alrKinematic`**`Axis`**`[Index]`
   ⇨ Parameter 6
9. Messen Sie die (positive) Strecke "Delta 2" zwischen der zuvor in Schritt 3 angebrachten Markierung und der neuen Position der linearen Gelenkachse mit einem Maßband
   ⇨ Parameter 7
10. Geben Sie die ermittelten Werte für Parameter 3 bis 7 sowie den Wert "3" für Parameter 8 ein und aktualisieren Sie die Konfiguration.
11. Die korrekte Parametrierung des Cranks ist plausibel, wenn die Differenzen der Gelenkachswerte in den Messpunkten 2 und 3 vom Messpunkt 1 den Werten "Delta 1" und "Delta 2" entsprechen.

26873346/DE – 07/2021

8.6 Feldbus-Schnittstelle

8.6.1 Allgemeine Einstellungen

Parameterbezeichnung	Beschreibung
Feldbuskonfiguration	
Feldbusanbindung aktivieren	Aktivierung der Feldbusanbindung. Weitere Informationen finden Sie im Kapitel "Feldbus-Schnittstelle" (→ 🗎 204)
Startadresse	Festlegung der Startadresse des Softwaremoduls in der Feldbus-Schnittstelle des MOVI-C® CONTROLLER
Prozessdatenlänge	Basierend auf der Konfiguration automatisch berechnete Prozessdatenlänge
Nachkommastellen über Feldbus	
Tabelle zum Einstellen der Nachkommastellen für Position, Geschwindigkeit, Beschleunigung und Ruck für die Translation und die Rotation über den Feldbus. Da die Posen- und Geschwindigkeitswerte über den Feldbus nur als ganze Zahlen übertragen werden können, wird so eine höhere Genauigkeit erzielt. Diese Einstellungen gelten für alle Variablen in der Feldbus-Schnittstelle des gewählten Typs. Wenn die Nachkommastellen im Softwaremodul größer 0 eingestellt sind, werden die Eingangswerte im MOVI-C® CONTROLLER vor der Verarbeitung als Gleitkommazahl mit dem Faktor $10^{\text{Anzahl Nachkommastellen}}$ dividiert. Andererseits werden die Ausgangswerte im MOVI-C® CONTROLLER mit diesem Faktor multipliziert, bevor sie zur übergeordneten Steuerung übertragen werden. Wenn z. B. die Nachkommastellen auf 2 eingestellt sind und in einem Eingangswort des MOVI-C® CONTROLLER der Wert 1 steht, wird im Controller mit dem Wert 0.01 gearbeitet. Wenn die Nachkommastellen auf 2 eingestellt sind und in einem Ausgangswort des MOVI-C® CONTROLLER der Wert 1 steht, muss dieser Wert in der übergeordneten Steuerung als 0.01 interpretiert werden.	
Feldbusprofilvorlage	
Feldbusprofil	Auswahl des Feldbusprofils: • Standardprofil für die Positionierung • Flexibles, parametrierbares Profil Weitere Informationen finden Sie im Kapitel "Feldbusprofile" (→ 🗎 204).

Abhängig vom gewählten Feldbusprofil stehen weitere Einstellungsmöglichkeiten zur Verfügung.

Feldbusprofil: Standardprofil für die Positionierung

Parameterbezeichnung	Beschreibung
Feldbuskonfiguration	

26873346/DE – 07/2021

Parameterbezeichnung	Beschreibung
Maximale Anzahl der Bahnsegmente pro Telegramm (Zielpose sowie Warte- und Endsignale)	Die maximale Anzahl von Bahnsegmenten wird in einem Telegramm übertragen und kann ohne Genau-Halt zwischen den Bahnsegmenten abgefahren werden. Mittels der in den Bahnsegmenten enthaltenen Endsignale kann die Anzahl der Bahnsegmente im Betrieb von Bahn zu Bahn zwischen 1 und der maximalen Anzahl angepasst werden.

Feldbusprofil: Flexibles, parametrierbares Profil

Parameterbezeichnung	Beschreibung
Konsistenz eines Telegramms	
Größe der konsistenten Blöcke	Einstellung der von der übergeordneten Steuerung verwendeten Größe der konsistenten Blöcke im Bereich der Feldbus-Schnittstelle des Softwaremoduls. Die Größe ist dabei im Bereich der Feldbus-Schnittstelle des Softwaremoduls auf der übergeordneten Steuerung für alle konsistenten Blöcke gleich zu halten (bis auf den letzten = Parameter "Anzahl der konsistenten Blöcke")
Anzahl der konsistenten Blöcke	Einstellung der von der übergeordneten Steuerung verwendeten Anzahl der konsistenten Blöcke im Bereich der Feldbus-Schnittstelle des Softwaremoduls
Erweiterungsmodule	
Aktivierung Roboter-Diagnose-Modul	Zuordnung des Erweiterungsmoduls "Roboter-Diagnose-Modul" auf dem Bus. Es wird empfohlen, das Erweiterungsmodul zu aktivieren.
Größe einer Posenvariablen	
Tabelle zum Einstellen der Größe der einzelnen Koordinaten aller Posen in der Feldbus-Schnittstelle des Softwaremoduls. Dabei kann jeweils zwischen "Nein (0 Bit)", "WORD (16 Bit)" und "DWORD (32 Bit)" gewählt werden. Die Summe der Bits über alle Koordinaten ergibt die Posengröße auf dem Bus. Standardmäßig werden nur X, Y, Z, A als WORD übertragen. Bei B und C ist "Nein" ausgewählt. Dies gilt auch für das "Standardprofil für die Positionierung". Nur wenn sich die gesamte Posengröße verändert (z. B. durch die Hinzunahme von B als WORD), verschieben sich weitere Signale in der Schnittstelle (im Beispiel dann jeweils um ein WORD pro Pose nach hinten).	

26873346/DE – 07/2021

8.6.2 Standard-Bahnsegmente

HINWEIS

Nur sichtbar bei der Auswahl des Feldbusprofils "Flexibles, parametrierbares Profil".

Parametergruppe / Beschreibung
Sollwerte
Anzahl der Standard-Bahnsegmente pro konsistentem Block Einstellung zum Hinzukonfigurieren von Standard-Bahnsegmente in jedem konsistenten Block. Die maximale Anzahl hängt dabei von der Größe des konsistenten Blocks, der Posengröße und den bereits durch andere Signale verbrauchten Prozessdatenwörtern ab. Wenn mehrere konsistente Blöcke verwendet werden sollen, muss der Parameter "Anzahl der konsistenten Blöcke" unter "Allgemeine Einstellung" entsprechend angepasst werden.

8.6.3 Weitere Posen

HINWEIS

Nur sichtbar bei der Auswahl des Feldbusprofils "Flexibles, parametrierbares Profil".

Parametergruppe / Beschreibung
Variablenanzahl auf dem Bus
Einstellung zum Hinzukonfigurieren von SRL-Programmvariablen dieses Typs als Soll- oder Istwert in jedem konsistenten Block. Die maximale Anzahl hängt dabei von der Größe des konsistenten Blocks, der Größe der Variable und den bereits durch andere Signale verbrauchten Prozessdatenwörtern ab. Wenn mehrere konsistente Blöcke verwendet werden sollen, muss der Parameter "Anzahl der konsistenten Blöcke" unter "Allgemeine Einstellung" entsprechend angepasst werden.
Versätze in der Bool-Variablenliste
Parameter zum Verschieben des Versatzes der Soll- bzw. Istwerte in der Robotervariablen-Liste diesen Typs. Der Wert kann zwischen 1 und 100 gewählt werden, wobei es nicht zu Überschneidung mit anderweitig verwendeten Variablen kommen darf (zwischen den Soll- und Istwerten, den Variablen aus den Bahnsegmenten - beginnen mit 1 - sowie Variablen die nicht über den Bus übertragen werden). Genügend Abstand zu den anderweitig verwendeten Variablen wird empfohlen, damit beim Hinzufügen weiterer Variablen das SRL-Programm nicht angepasst werden muss.

26873346/DE – 07/2021

8.6.4 Weitere Reals

HINWEIS

Nur sichtbar bei der Auswahl des Feldbusprofils "Flexibles, parametrierbares Profil".

Parametergruppe / Beschreibung
Variablenanzahl auf dem Bus
Einstellung zum Hinzukonfigurieren von SRL-Programmvariablen dieses Typs als Soll- oder Istwert in jedem konsistenten Block. Die maximale Anzahl hängt dabei von der Größe des konsistenten Blocks, der Größe der Variable und den bereits durch andere Signale verbrauchten Prozessdatenwörtern ab. Wenn mehrere konsistente Blöcke verwendet werden sollen, muss der Parameter "Anzahl der konsistenten Blöcke" unter "Allgemeine Einstellung" entsprechend angepasst werden.
Versätze in der Real-Variablenliste
Parameter zum Verschieben des Versatzes der Soll- bzw. Istwerte in der Robotervariablen-Liste diesen Typs. Der Wert kann zwischen 1 und 100 gewählt werden, wobei es nicht zu Überschneidung mit anderweitig verwendeten Variablen kommen darf (zwischen den Soll- und Istwerten, den Variablen aus den Bahnsegmenten - beginnen mit 1 - sowie Variablen die nicht über den Bus übertragen werden). Genügend Abstand zu den anderweitig verwendeten Variablen wird empfohlen, damit beim Hinzufügen weiterer Variablen das SRL-Programm nicht angepasst werden muss.
Sollwerte in der Real-Variablenliste
Tabelle zum Einstellen der Größe auf dem Bus sowie des Typs der Nachkommastellen der konfigurierten Real-Variablen. Die Einstellung der Anzahl Nachkommastellen wird unter "Allgemeine Einstellungen" vorgenommen, siehe Parameter "Nachkommastellen über Feldbus".
Istwerte in der Real-Variablenliste
Tabelle zum Einstellen der Größe auf dem Bus sowie des Typs der Nachkommastellen der konfigurierten Real-Variablen. Die Einstellung der Anzahl Nachkommastellen wird unter "Allgemeine Einstellungen" vorgenommen, siehe Parameter "Nachkommastellen über Feldbus".

26873346/DE – 07/2021

8.6.5 Weitere Bools

HINWEIS

Nur sichtbar bei der Auswahl des Feldbusprofils "Flexibles, parametrierbares Profil".

Parametergruppe / Beschreibung
Variablenanzahl auf dem Bus
Einstellung zum Hinzukonfigurieren von SRL-Programmvariablen dieses Typs als Soll- oder Istwert in jedem konsistenten Block. Die maximale Anzahl hängt dabei von der Größe des konsistenten Blocks, der Größe der Variable und den bereits durch andere Signale verbrauchten Prozessdatenwörtern ab. Wenn mehrere konsistente Blöcke verwendet werden sollen, muss der Parameter "Anzahl der konsistenten Blöcke" unter "Allgemeine Einstellung" entsprechend angepasst werden.
Versätze in der Bool-Variablenliste
Parameter zum Verschieben des Versatzes der Soll- bzw. Istwerte in der Robotervariablen-Liste diesen Typs. Der Wert kann zwischen 1 und 100 gewählt werden, wobei es nicht zu Überschneidung mit anderweitig verwendeten Variablen kommen darf (zwischen den Soll- und Istwerten, den Variablen aus den Bahnsegmenten - beginnen mit 1 - sowie Variablen die nicht über den Bus übertragen werden). Genügend Abstand zu den anderweitig verwendeten Variablen wird empfohlen, damit beim Hinzufügen weiterer Variablen das SRL-Programm nicht angepasst werden muss.

8.7 Physics

HINWEIS

Nur bei den Kinematikmodellen sichtbar, die eine Physiksimulation unterstützen. Bei jedem "Kinematikmodell" (→ 🗎 44) ist angegeben, ob es die Physiksimulation unterstützt. Weitere Informationen finden Sie im Kapitel "Physiksimulation" (→ 🗎 95).

Die Physiksimulation ermittelt die zu erwartenden Kraft-Momenten-Belastungen auf die Antriebe von Robotern. Dabei werden die Antriebsmomente (und abhängig vom Kinematikmodell zusätzlich Kippmomente und Querkräfte) zyklisch abhängig von Massen, Beschleunigungen und Stellungen der Armglieder berechnet. Diese Belastungen können vom MOVI-C® CONTROLLER exportiert und in die SEW-Workbench importiert werden, um dort eine Antriebsauswahl für den Roboter zu treffen (Projektierung).

Die Physiksimulation bietet den Vorteil einer realitätsgetreuen Ermittlung der Kraft-Momenten-Belastungen mit exakt den Bewegungsprofilen, die an der realen Anlage abgefahren werden.

Parameterbezeichnung	Beschreibung
General Settings	
Physiksimulation aktivieren	Aktivierung der Physiksimulation

Abhängig vom gewählten Kinematikmodell werden bei aktivierter Physiksimulation folgende Untermenüs mit spezifischen Einstellungsmöglichkeiten angezeigt:

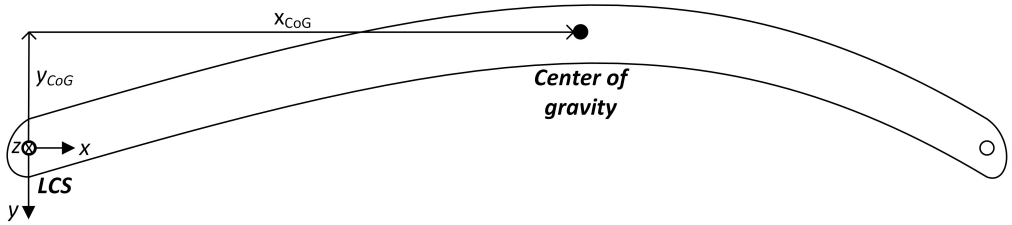
Für alle Armglieder des gewählten Kinematikmodells müssen die folgenden Trägheitseigenschaften definiert werden.

26873346/DE – 07/2021

HINWEIS

Beim TRIPOD_RRR_M10 wird davon ausgegangen, dass alle drei Oberarme und alle drei Unterarme identische Massen, Schwerpunkte und Trägheiten aufweisen.

Parameterbezeichnung	Beschreibung
[Kinematikmodell] DynamicModel	
Mass	Masse des Armgliedes.
Center of gravity (X, Y, Z)	Schwerpunkt des Armgliedes. X, Y und Z beziehen sich auf das LCS (Link Coordinate System) des Armgliedes.
Inertia (XX, YY, ZZ, XY, XZ, ZX)	Trägheitsmatrix um den Schwerpunkt. X, Y und Z beziehen sich auf das LCS (Link Coordinate System) des Armgliedes.



33795862539

Der Ursprung des LCS (Link Coordinate System) jedes Armgliedes liegt am Anfang des Armglieds im Gelenk (siehe Abbildung). Die Z-Achse zeigt in positive Achs-Richtung. Die x-Richtung zeigt in der Regel zum nächsten Gelenk.

Die LCS aller Kinematikmodelle sind im Kapitel "Funktionsbeschreibung" (→ 🗎 44) beim jeweiligen Kinematikmodell dargestellt und können auch in der 3D-Simulation eingeblendet werden.

Für die folgenden Kinematikmodelle gibt es zudem Besonderheiten und spezifische Konfigurationsparameter.

Parameterbezeichnung	Beschreibung
ROLLER GANTRY: General Dynamic Model Settings	
Roller Gantry Mechanische Ausführung	Varianten • H: Mechanik sieht aus wie ein „H“ • Cross: Mechanik sieht aus wie ein Kreuz oder ein Plus – Top Fixed: Riemen ist oben ohne Riemenrad fest mit der Mechanik verbunden. – Bottom Fixed: Riemen ist unten ohne Riemenrad fest mit der Mechanik verbunden.
Riemenbreite	Breite des Riemens. Mit steigender Riemenbreite steigt i. A. die Reibung im System. Diese gesteigerte Reibung wird in der Physiksimulation berücksichtigt.

26873346/DE – 07/2021

Parameterbezeichnung	Beschreibung
Manuelle Vorgabe der Riemenvorspannung	• Deaktiviert: Riemenvorspannung wird automatisch anhand der vorliegenden Belastungen mittels Riemenprojektierungsrichtlinien ermittelt (Grob abgeschätzt, i. A. falsch) • Aktiviert: Siehe Parameter "Riemenvorspannung"
Riemenvorspannung	Riemenvorspannung. Nur wirksam, wenn die manuelle Vorgabe der Riemenvorspannung aktiviert ist.
ROLLER_GANTRY_RR_M10: DynamicModel	
Belt wheels: Inertia	Trägheitsmoment aller Riemenräder zusammen
Belt: Inertia	Trägheitsmoment des Riemens
1. Cartesian link: Mass	Masse des ersten kartesischen Gelenkes, das nur in eine Richtung bewegt werden kann: • inklusive der daran montierten Riemenräder • ohne die Masse des zweiten kartesischen Gelenkes
2. Cartesian link: Mass	Masse des zweiten kartesischen Gelenkes, das in zwei Richtungen bewegt werden kann: • inklusive der daran montierten Riemenräder und Greifer • ohne die Masse des ersten kartesischen Gelenkes

Der Roboter kann am TCP mit einem Werkstück zur Laufzeit beladen werden. Für die Berücksichtigung in der Physiksimulation können die Trägheitseigenschaften im Untermenü "Workpiece Settings" konfiguriert werden.

Parameterbezeichnung	Beschreibung
WorkpieceSettings	
Mass	Masse des Werkstücks
Center of gravity (X, Y, Z)	Schwerpunkt des Werkstücks. X, Y und Z beziehen sich auf das TCS (Tool Coordinate System) des Roboters.
Inertia (XX, YY, ZZ, XY, XZ, ZX)	Trägheitsmatrix um den Schwerpunkt. X, Y und Z beziehen sich auf das TCS (Tool Coordinate System) des Roboters.

26873346/DE – 07/2021

8.8 Additional Functions

Parameterbezeichnung	Wert
Touchprobe Measure + Positioning	
Use Touchprobe Functions	Aktivierung bzw. Deaktivierung der Touchprobe-Funktion (Messung und/oder Restwegpositionierung) Wenn die Funktion aktiviert wird, kann diese im RobotMonitor ausgewählt werden und die entsprechende Lizenz wird vom MOVI-C® CONTROLLER abgerufen. Weitere Informationen finden Sie im Kapitel "Touchprobe" (→ 🗎 86).
	Index: 50020.24
	IEC-Name: -
Circular Interpolation	
Use Circular Motion Commands	Aktivierung bzw. Deaktivierung des MOVIKIT® Robotics addon Circle. Weitere Informationen finden Sie im Kapitel "Add-ons" (→ 🗎 21).
	Index: 50020.17
	IEC-Name: -
Conveyor Tracking oder Rotary Table Tracking	
Use Conveyor Tracking	Aktivierung bzw. Deaktivierung des MOVIKIT® Robotics addon ConveyorTracking. Weitere Informationen finden Sie im Kapitel "Add-ons" (→ 🗎 20).
	Index: 50020.16
	IEC-Name: -

8.9 Modulidentifikation

Parametergruppe	Beschreibung
Modulidentifikation	Angabe u. a. des Namens und der Version zur Identifikation des Softwaremoduls.

26873346/DE – 07/2021

9 Ansteuerung durch die Prozess-Steuerung

9.1 Prozess-Steuerung

Nach der Inbetriebnahme und der SRL-Programmierung ist das Softwaremodul betriebsbereit. Das Softwaremodul kann über das IEC-Anwenderprogramm auf dem MOVI-C® CONTROLLER, über den Feldbus von der SPS oder vom Bediener über den RobotMonitor angesteuert werden. In diesem Handbuch werden diese daher zusammengefasst als Prozess-Steuerung bezeichnet.

9.2 Prozess-Steuerung in Betrieb nehmen

- ✓ Der Funktionstest der Parametrierung war erfolgreich. Weitere Informationen finden Sie im Kapitel "Achsen referenzieren und Funktionstest durchführen" (→ 🗎 110).
- ✓ Ein SRL-Programm wurden erstellt. Weitere Informationen finden Sie im Kapitel "SRL-Programmierung" (→ 🗎 137).

1. Programmieren Sie den "Prozess-Start" (→ 🗎 133).
2. Programmieren Sie den "Prozessablauf" (→ 🗎 133).
3. Programmieren Sie die "Handhabung von Fehlersituationen" (→ 🗎 135).

9.3 Wichtige Statussignale

Zugriff

Über das Signal *Get Access Control* kann der Zugriff auf den Roboter angefordert werden. Wenn das Anfordern erfolgreich ist, wird das Signal *Control Active* zurückgegeben. Weitere Informationen dazu erhalten Sie im Kapitel "Zugriffsverwaltung" (→ 🗎 41).

Fehler

Wenn Fehler vorliegen, beseitigen Sie zunächst die Fehlerursachen. Zur Ermittlung der Fehlerursache können die im RobotMonitor oder im IEC-Editor angezeigten Fehlernachrichten helfen. Den Fehlerzustand beheben Sie durch setzen des Signals *Reset*.

Sollwerte aktiv

Für die Steuerung des Roboters muss *Setpoints Active* "TRUE" sein. Falls nicht, ermitteln Sie die Ursache:

1. Prüfen Sie, ob ein Fehler anliegt.
2. Prüfen Sie, ob die Freigabe gesetzt ist.
3. Prüfen Sie, ob der Zugriff angefordert ist.

26873346/DE – 07/2021

9.4 Voraussetzungen für Bewegungen des Roboters

Allgemein

- Sollwerte aktiv
- Override > 0
- Steuerungsartabhängige Voraussetzungen

Programmbetrieb

- Bewegungsprogramm beinhaltet Bewegungsbefehle zu Posen, die nicht der aktuellen Pose entsprechen.
- Programmpause deaktiviert
- Programmstopp deaktiviert
- Steigende Flanke von Programmstart

  Damit die steigende Flanke das Programm startet, muss die Flanke mindestens einen Zyklus nach der steigenden Flanke von Sollwert aktiv sowie den fallenden Flanken von Programmpause und Programmstopp sein.
- In der Betriebsart "Manuell mit hoher Geschwindigkeit" muss Programmstart während der kompletten Ausführung des Programms aktiviert bleiben. Weitere Informationen erhalten Sie im Kapitel "Programmbetrieb" (→ 🗎 37).

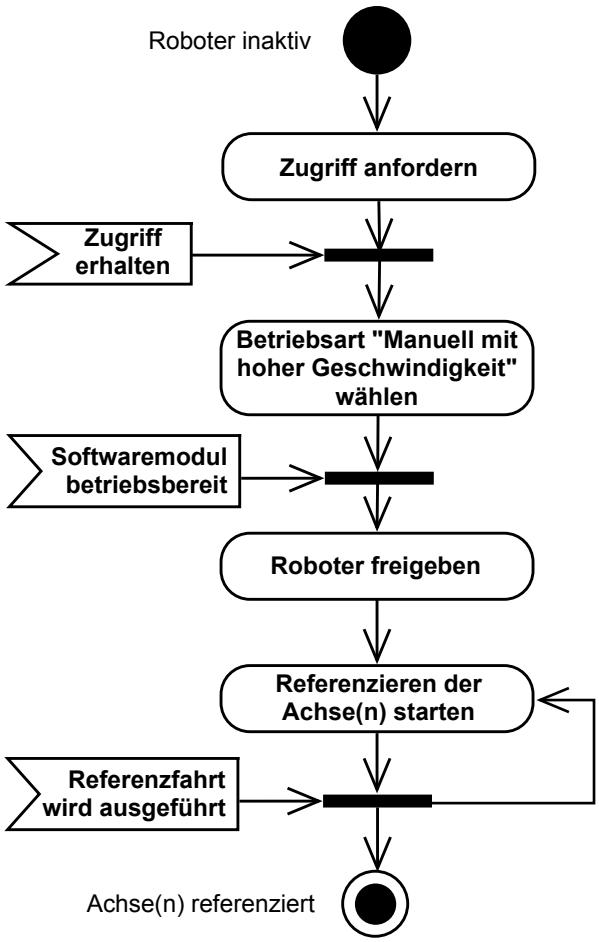
Tippbetrieb

- Im Tippbetrieb muss mindestens ein Tippsignal aktiviert sein (positiv oder negativ).

26873346/DE – 07/2021

9.5 Referenzierung

Das Referenzieren der Achse(n) vor Prozessstart kann folgendermaßen aussehen. Abhängig von der in der Einzelachse konfigurierten Referenzfahrt muss die Roboterachse vor dem Referenzieren mittels unreferenziertem (achsweisem) Tippen (*JOG_AXIS*) in Referenzstellung gebracht werden.

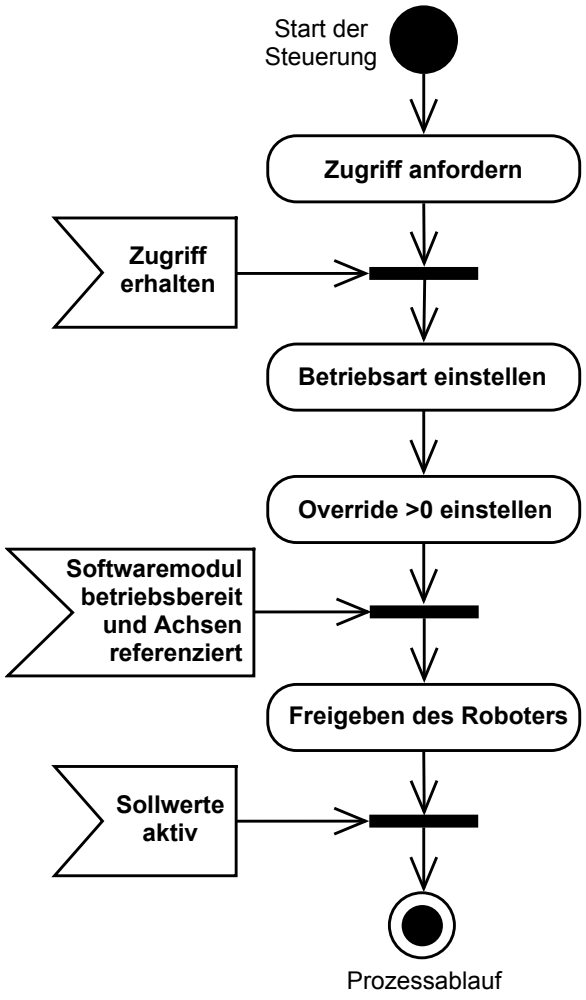


34808485259

26873346/DE – 07/2021

9.6 Prozess-Start

Der Prozess-Start in der Prozesssteuerung nach dem Einschalten der Anlage kann folgendermaßen aussehen. Dabei können applikativ weitere Abfragen hinzukommen.



27021621098059403

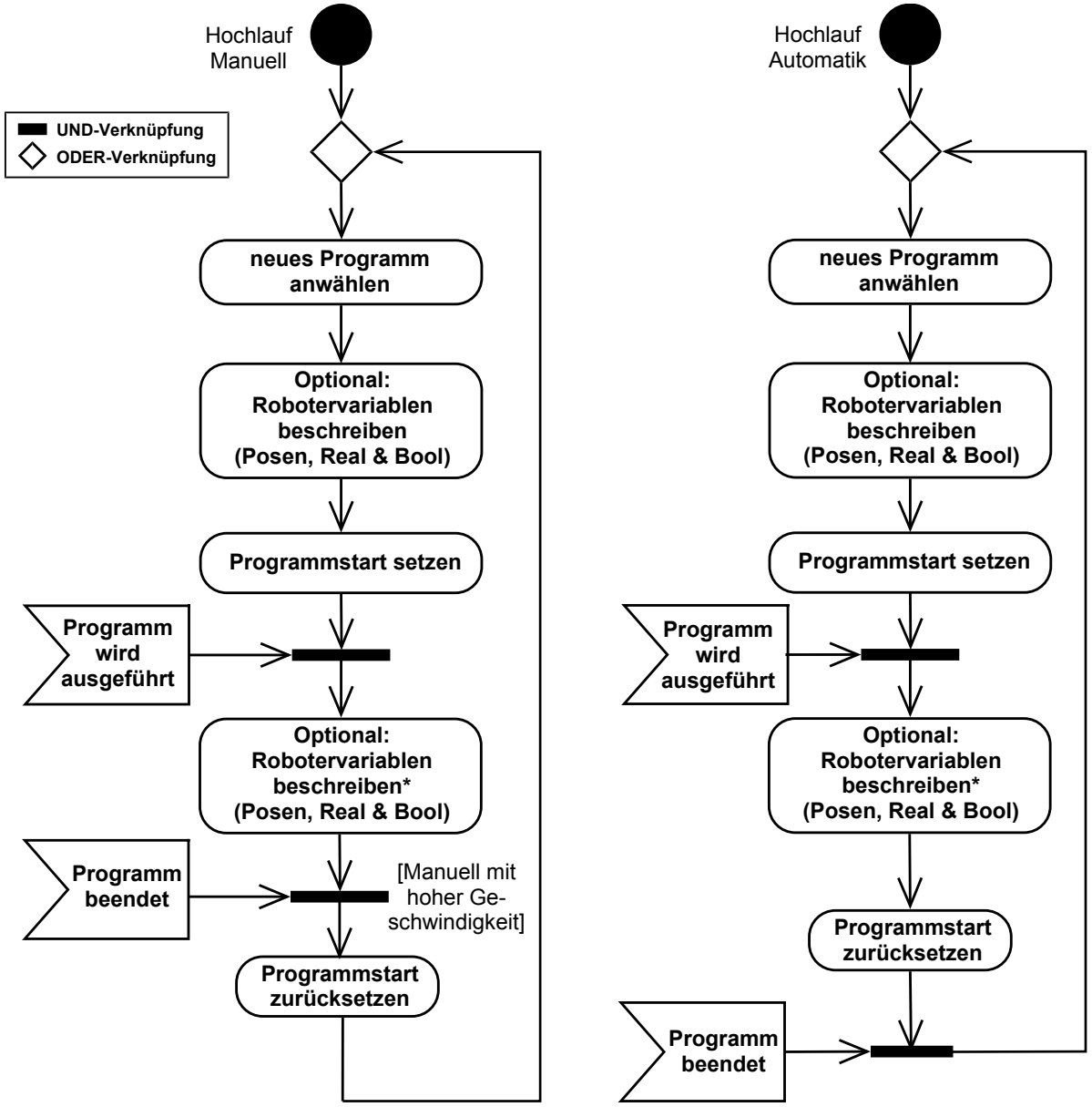
9.7 Prozessablauf

Programme können folgendermaßen gestartet werden:

- Zuerst wird die Programmnummer vorgegeben. Falls gewünscht, können die SRL-Programmvariablen neu parametriert werden (Bahnposen, REAL- und BOOL-Variablen).
- Ist das Programm initialisiert, kann es mit einem Programmstart gestartet werden.
- In der Betriebsart "Manuell mit hoher Geschwindigkeit" muss der Start-Knopf während der kompletten Programmabarbeitung aktiviert sein.
- Bei jedem Programmablauf können zusätzlich die Wartesignale während der Programmabarbeitung beschaltet werden.
- Wenn es erkannt wurde, dass das Programm abgeschlossen ist, beginnt der Ablauf wieder von vorne.

26873346/DE – 07/2021

Das folgende Ablaufdiagramm zeigt, wie das kontinuierliche Ausführen von Programmen realisiert werden kann. Bevor die gezeigte Sequenz ausgeführt werden kann, müssen die Voraussetzungen für eine Bewegung im Programmbetrieb erfüllt sein, siehe Kapitel "Voraussetzungen für Bewegungen des Roboters" (→ 🗎 131).



27021621098062219

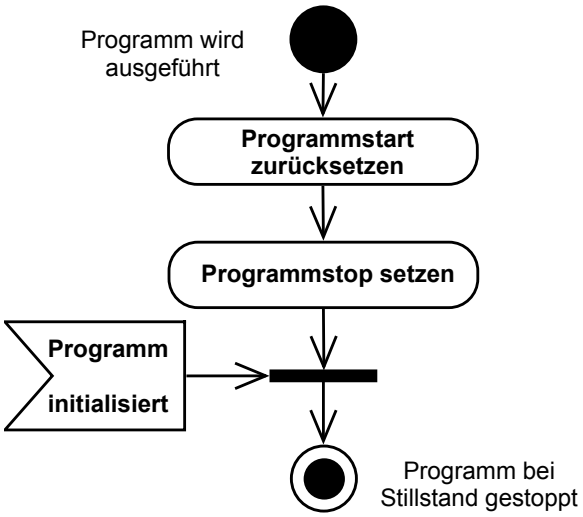
Anmerkungen

- Die neuen Werte werden wirksam, wenn der Roboter einen Befehl abarbeitet, in dem die Variable verwendet wird. Siehe Kapitel "SRL-Programm ausführen" (→ 🗎 163).
- Zum Beschreiben der Robotervariablen bietet sich die Verwendung des SRL-Befehls *CallFunction* an. Die *CallFunction* führt den enthaltenen IEC-Code klar definiert und konsistent im Programmablauf an der Stelle aus, wo die *CallFunction* im SRL-Programm platziert ist. In der *CallFunction* können beispielsweise auch MotionSets beschrieben werden (mittels Methode im UserInterface).

26873346/DE – 07/2021

9.8 Prozess-Stopp

Wenn das aktuelle Programm gestoppt werden soll und wieder von Beginn des Programms gestartet werden soll, muss dafür Programmstopp ausgeführt werden.



9007230624602763

9.9 Fehlersituationen handhaben

Ob das MOVIKIT® Robotics im Fehlerzustand ist, kann am Signal Fehler erkannt werden. Der Fehler kann vom Controller ausgelöst oder von zugeordneten Umrichtern gemeldet werden. Ob es sich um einen Fehler eines Umrichters handelt, kann an den Fehlerausgängen der Einzelachsen erkannt werden. Im Fehlerfall wird das Signal Sollwert aktiv auf "FALSE" gesetzt.

Wenn das Softwaremodul im Fehlerzustand ist, wird automatisch mit den konfigurierten Not-Halt-Rampen abgebremst. Wenn die Fehlerursache beseitigt und der Fehler zurückgesetzt ist, kann weiter verfahren werden. Das Zurücksetzen des Fehlers geschieht über das Signal Reset.

Um herauszufinden, was die Ursache für einen Fehler ist, wird im Signal MessageID eine eindeutige Fehlernummer und ein Fehlertext ausgegeben. Im Falle eines Umrichterfehlers finden Sie weitere Informationen zur Fehlernummer in der Dokumentation des Geräts.

9.9.1 Begrenzung des Arbeitsraums

Jeder Roboter hat einen Arbeitsraum, in dem er sich bewegen darf. Der Arbeitsraum resultiert aus dem Kinematikmodell und aus den Begrenzungen durch die Software. Diese Begrenzungen werden vom Anwender mit Kenntnis über die Umgebung und Mechanik des realen Roboters parametriert.

Bei Überschreitung einer Softwarelimitierung wird automatisch ein Not-Halt ausgelöst und eine Fehlermeldung (0x7e60) ausgegeben. Nach anschließendem Fehler-Reset kann der Roboter im Tippbetrieb verfahren werden. In den Ausgangssignalen (IEC: *Basic.Out.xOutOfWorkspace*, PD PA 7:1) und in einer Warnung (0x17e60) wird weiterhin darauf hingewiesen, dass sich der Roboter außerhalb des Arbeitsraums befindet. Sobald sich der Roboter innerhalb des Arbeitsraums befindet, erlöschen die Hinweise und es kann wieder ein Programm gestartet werden.

Folgende Softwarelimitierungen können für den Arbeitsraum parametriert werden:

26873346/DE – 07/2021

Gelenkachsen-Limitierungen

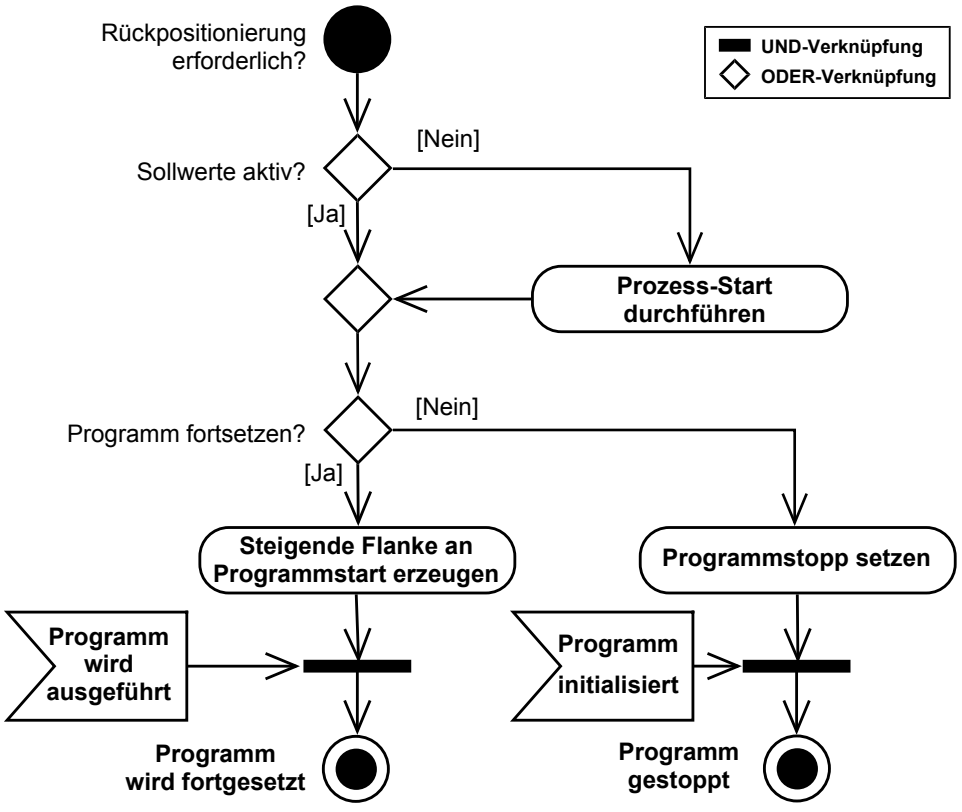
- Grenzwerte für aktiv durch einen Antrieb bewegte Dreh- und Schubgelenke. Sie können u. a. aus den folgenden Randbedingungen resultieren:
  - Eingeschränkte Beweglichkeit des Gelenks
  - Aufwickeln von Kabeln
  - Endlagen der Linearführung

Kartesische Limitierungen

- Grenzwerte für die kartesischen Koordinaten des Werkzeugarbeitspunkts und der Werkzeugorientierung. Räumlich spannen die 3 translatorischen Limitierungen einen Quader auf, innerhalb dessen sich das Werkzeug bewegen darf. Die rotatorische Limitierung schränkt zusätzlich die erlaubte Orientierung des Werkzeugs ein.

9.10 Rückpositionierung (BackToPath) erforderlich

Wenn eine Rückpositionierung (BackToPath) als erforderlich gemeldet wird, gibt es zwei Möglichkeiten dies zu handhaben. Entweder bricht man mit Programmstopp das vorhandene Programm ab und startet ein neues (siehe vorheriges Kapitel) oder man startet mit Programmstart die Rückpositionierung und setzt das Programm fort.



31381916043

26873346/DE – 07/2021

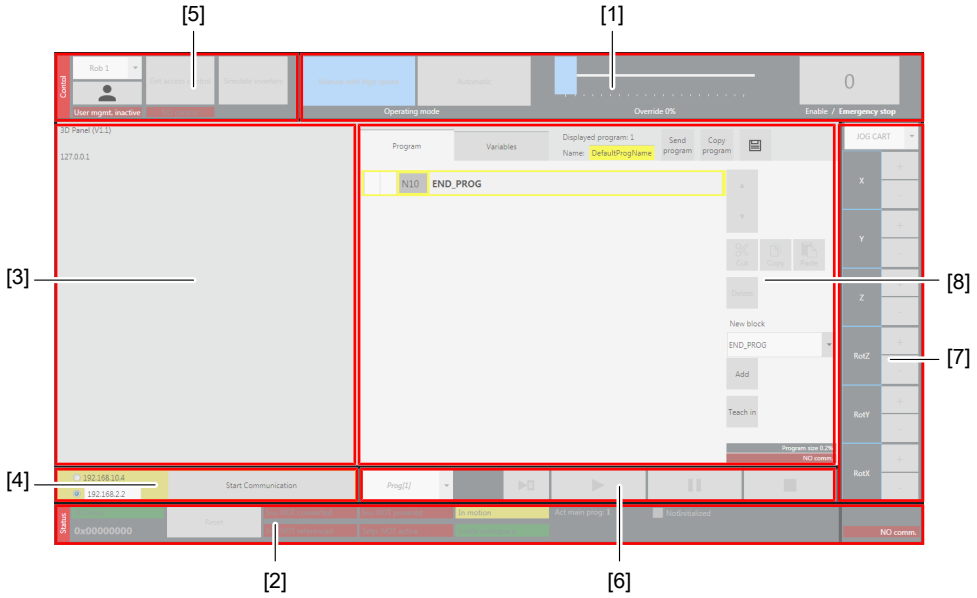
10 SRL-Programmierung

In diesem Kapitel sind die Signale des RobotMonitors und das Erstellen von SRL-Programmen mittels der SEW-Robot-Language (SRL) beschrieben. In welcher Reihenfolge die Signale angesteuert werden sollen, ist im Kapitel "Ansteuerung durch die Prozess-Steuerung" (→ 🗎 130) beschrieben.

10.1 Voraussetzungen

- Das IEC-Projekt mit dem Softwaremodul ist auf den MOVI-C® CONTROLLER heruntergeladen und gestartet. Weitere Informationen dazu finden Sie im Kapitel "IEC-Projekt generieren" (→ 🗎 105).
- Der RobotMonitor ist "gestartet" (→ 🗎 107) und eine Verbindung wurde "aufgebaut" (→ 🗎 108).

10.2 Benutzeroberfläche



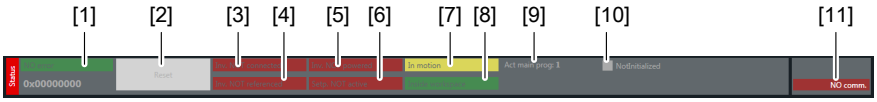
31688038411

Nr.	Beschreibung
[1]	"Allgemeine Steuerung des Roboters" (→ 🗎 138)
[2]	"Allgemeiner Status und Diagnosemeldungen des Roboters" (→ 🗎 138)
[3]	"3D-Simulation des Roboters" (→ 🗎 142)
[4]	Steuerung der "Kommunikationsverbindung" (→ 🗎 108)
[5]	"Zugriffsverwaltung/Benutzerverwaltung" (→ 🗎 108)
[6]	"Programmstatus und Steuerung des Programmbetriebs" (→ 🗎 144)
[7]	"Steuerung des Tippbetriebs" (→ 🗎 144) / "Referenzierbetriebs" (→ 🗎 144)
[8]	Register zum "Erstellung des SRL-Programms" (→ 🗎 147)

10.2.1 Allgemeine Steuerung des Roboters

Nr.	Beschreibung	IEC-Name
[1]	Zugriff auf den Roboter (HMI) anfordern/zurückgeben. Siehe "Zugriffsverwaltung" (→ 🗎 41).	xGetAccessControl
[2]	Status der Steuerung des Roboters (Aktive Steuerquelle bei MouseOver)	xControlActive
[3]	Simulation der Roboterachsen aktivieren/deaktivieren. Siehe "Simulation" (→ 🗎 42). **HINWEIS:** Die Achsen können nur in Simulation geschaltet werden, wenn der Roboter nicht freigegeben ist.	Inverter.IN.xSimulation
[4]	Deaktivieren der kartesischen und gelenkbezogenen Arbeitsraumüberwachung	SoftwareLimitSwitch.IN.xDisable
[5]	Betriebsart	Basic.IN.eOperatingMode
[6]	Skalierung der Geschwindigkeit in [%]	Basic.IN.usiOverride
[7]	Not-Halt (Enable / EmergencyStop)	Basic.IN.xEnable_EmergencyStop

10.2.2 Allgemeiner Status und Diagnosemeldungen des Roboters



Nr.	Beschreibung	IEC-Name
[1]	Fehlerstatus	xError
	Direkt darunter: ID der Meldung	udiMessageID
[2]	Fehler zurücksetzen	xReset
[3]	Verbindungsstatus	Inverter.OUT.xConnected
[4]	Status der Referenzierung	Inverter.OUT.xReferenced
[5]	Freigabestatus	Inverter.OUT.xPowered
[6]	Status der Sollwertverarbeitung	Inverter.OUT.xSetpointActive
[7]	Bewegungsstatus	Basic.OUT.xSetpointStandstill
[8]	Arbeitsraum verlassen	Basic.OUT.xOutOfWorkspace
[9]	Aktuelle Programmnummer	Prg.OUT.uiProgramNumber
[10]	Programmstatus	Prg.OUT.eProgramState
[11]	Verbindungsstatus des RobotMonitors	-

26873346/DE – 07/2021

10.2.3 Benutzerverwaltung

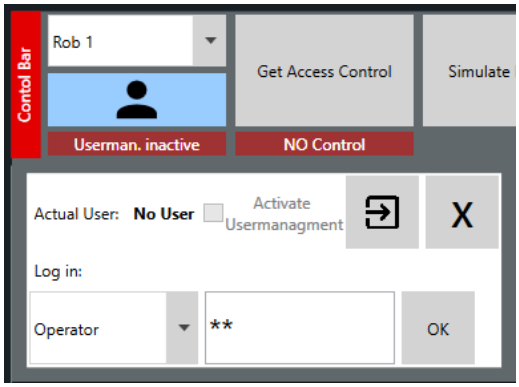
HINWEIS

Die Benutzerverwaltung ist standardmäßig deaktiviert. Das Anlegen und Konfigurieren von Benutzern ist nicht zwingend notwendig. Wenn Sie den RobotMonitor nicht an der Maschine verwenden wollen, können Sie diesen Schritt überspringen.

Benutzerverwaltung öffnen

✓ Der RobotMonitor ist geöffnet.

1. Klicken Sie zum Öffnen der Benutzerverwaltung auf die Schaltfläche .
   ⇨ Die Benutzerverwaltung wird geöffnet.



31659964171

Als Benutzer anmelden

✓ Die Benutzerverwaltung ist geöffnet.

1. Wählen Sie im Auswahlfeld "Log In:" den gewünschten Benutzer aus.
2. Geben Sie im Eingabefeld rechts daneben das dazugehörige Passwort ein.
3. Bestätigen Sie Ihre Eingabe mit [OK].
   ⇨ Im Feld "Actual User:" wird der Benutzername angezeigt.
4. Klicken Sie zum Schließen der Benutzerverwaltung auf die Schaltfläche [X].

Als Benutzer abmelden

✓ Die Benutzerverwaltung ist geöffnet.

✓ Sie sind als Benutzer angemeldet.

1. Klicken Sie zum Abmelden des Benutzers auf die Schaltfläche .
   ⇨ Im Feld "Actual User:" wird *No User* angezeigt.
2. Klicken Sie zum Schließen der Benutzerverwaltung auf die Schaltfläche [X].

26873346/DE – 07/2021

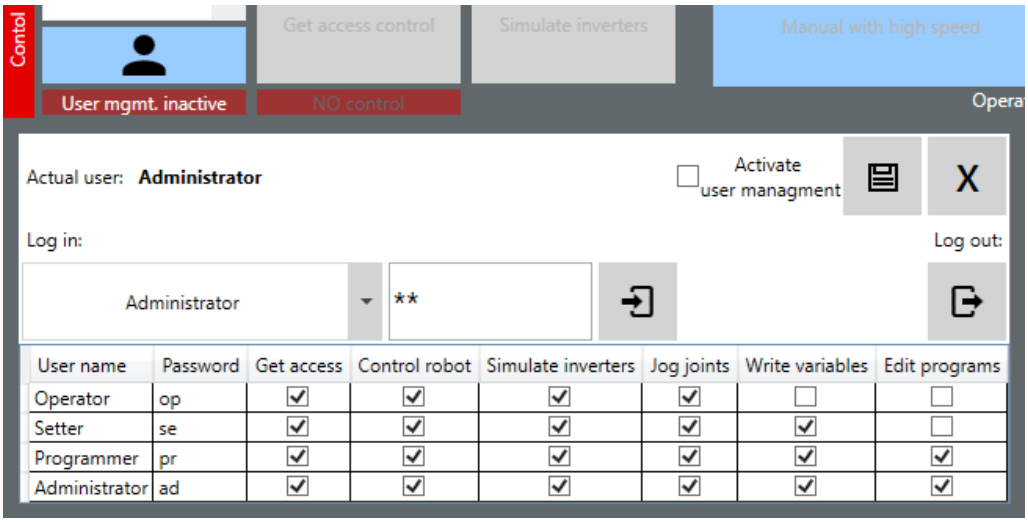
Benutzerverwaltung aktivieren

HINWEIS

Die Benutzerverwaltung ist standardmäßig deaktiviert. Der Status der Aktivierung wird unterhalb der Schaltfläche zum Öffnen der Benutzerverwaltung angezeigt. Nur Administratoren können die Benutzerverwaltung aktivieren und deaktivieren.

✓ Die Benutzerverwaltung ist geöffnet.

1. Melden Sie sich in der Benutzerverwaltung als Administrator an (Standardwert des Passworts: "ad").
   ⇨ Im Feld "Actual User:" wird *Administrator* angezeigt.
   ⇨ Eine Auflistung der angelegten Benutzer wird angezeigt.



9007225108439691

2. Aktivieren Sie das Kontrollfeld "Activate Usermanagement".
   ⇨ Die Benutzerverwaltung ist aktiviert. Der Status unterhalb der Schaltfäche zum Öffnen der Benutzerverwaltung zeigt *Userman. active* an.
3. Melden Sie sich von der Benutzerverwaltung ab. Siehe Kapitel "Als Benutzer abmelden" (→ 🗎 139).
4. Klicken Sie zum Schließen der Benutzerverwaltung auf die Schaltfläche [X].

26873346/DE – 07/2021

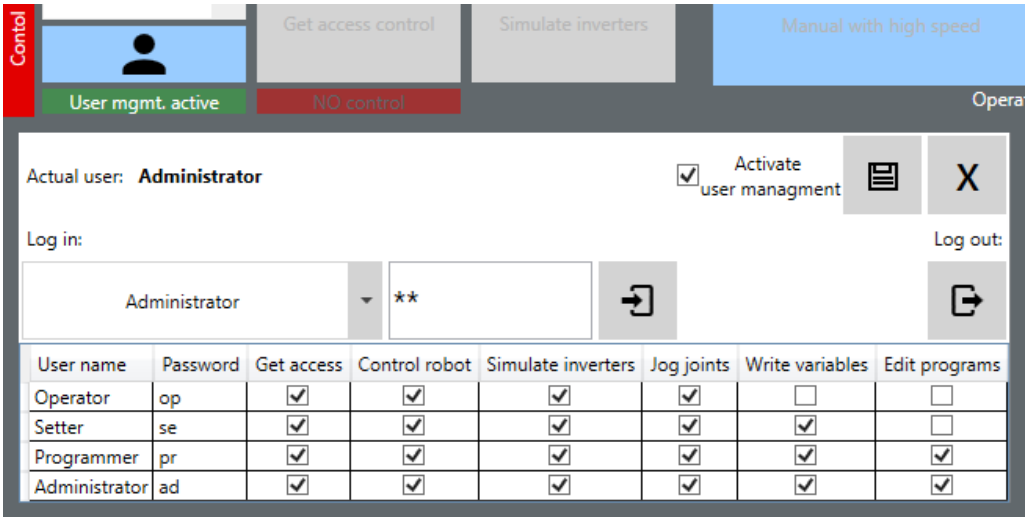
Benutzer konfigurieren

HINWEIS

Nur Administratoren können Passwörter und Zugriffsberechtigungen konfigurieren.

✓ Die Benutzerverwaltung ist geöffnet und aktiviert.

1. Melden Sie sich in der Benutzerverwaltung als Administrator an (Standardwert des Passworts: "ad").
   ⇨ Im Feld "Actual User:" wird *Administrator* angezeigt.
   ⇨ Eine Auflistung der angelegten Benutzer wird angezeigt.



9007225112801803

2. Aktivieren oder deaktivieren Sie über die entsprechenden Kontrollfelder die Zugriffsberechtigung der einzelnen Benutzer. Folgende Zugriffsberechtigungen sind verfügbar:

Kontrollfeld	Funktion
GetAccess	Zugriff anfordern
ControlRobot	Roboter im Tipp- und Automatik-Betrieb steuern
WriteVariables	SRL-Programmvariablen ändern
EditPrograms	SRL-Programme ändern
SimulateInverters	Umschalten der am Roboter angeschlossenen Umrichter auf Simulationsbetrieb (setzt die Berechtigung "ControlRobot" voraus)
JogJoints	Tippen der einzelnen Robotergelenke (setzt die Berechtigung "ControlRobot" vorraus)

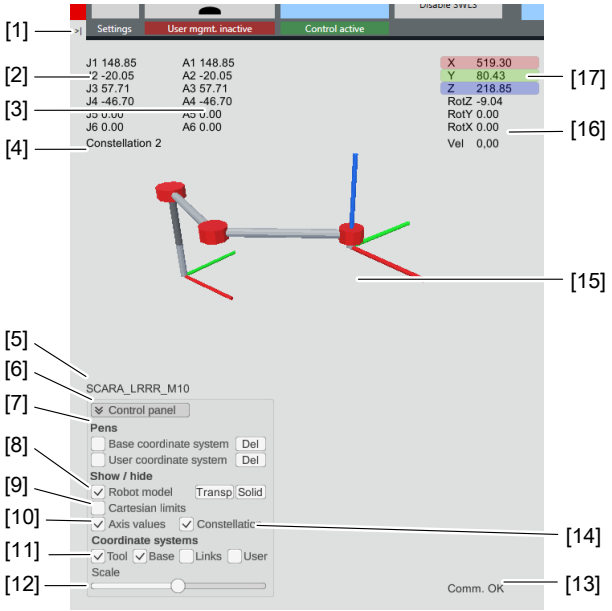
3. Klicken Sie zum Speichern der Konfiguration auf die Schaltfläche .
   ⇨ Die konfigurierten Zugriffsberechtigungen der Benutzer werden wirksam.
4. Klicken Sie zum Schließen der Benutzerverwaltung auf die Schaltfläche [X].

10.2.4 3D-Simulation

In der im RobotMonitor integrierten 3D-Simulation wird die aktuelle Stellung der Roboters angezeigt und mit dem "Pen"-Werkzeug kann die Bahn des TCP aufgezeichnet werden.

Die Ansicht in der 3D-Simulation kann über folgende Aktionen beeinflusst werden:

- Halten der linken Maustaste und Bewegen der Maus: Drehen der Ansicht um den eingeblendeten grünen Referenzpunkt.
- Halten der rechten Maustaste und Bewegen der Maus: Verschieben der Ansicht in der Bildschirmebene, d.h. der grüne Referenzpunkt wird verschoben. Dies bewirkt auch, dass die Ansicht mit der linken Maustaste um den nun verschobenen Referenzpunkt gedreht wird.
- Drehen des Mausrades: Zoomen (Vergrößern und Verkleinern) der Ansicht. Es kann maximal bis zum grünen Referenzpunkt hineingezoomt werden. Wenn Objekte hinter dem grünen Referenzpunkt aus der Nähe betreachtet werden sollen, muss zunächst der Referenzpunkt mit der rechten Maustaste dahin verschoben werden.



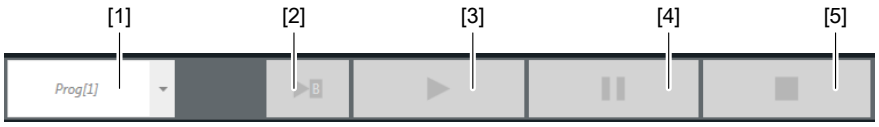
Nr.	Beschreibung
[1]	Umschalten der 3D-Ansicht (Splitscreen, Fullscreen, Minimiert)
[2]	Aktuelle Positionen der Gelenke (*Basic.OUT.SetpointPose.aIrKinematicJOint*)
[3]	Aktuelle Positionen der Achsen (*Basic.OUT.SetpointPose.aIrKinematicAxis*) Nur sichtbar, wenn die Anzeige über das Steuerelement [10] aktiviert ist.
[4]	Aktuelle Konstellation des Roboters (*Basic.OUT.SetpointPose.usiConstellation)* Nur sichtbar, wenn die Anzeige über das Steuerelement [14] aktiviert ist.
[5]	Aktuell gewähltes Kinematikmodell
[6]	Ein- und Ausklappen der Bedienelemente

26873346/DE – 07/2021

Nr.	Beschreibung
[7]	Steuerelemente von "Pen" (Funktion zum Mitzeichnen der TCP-Bahn) Das Kontrollfeld [Pen Base CoordSys] zum Zeichnen der Bahn relativ zum statischen Base-Koordinatensystem des Roboters aktivieren. Das Kontrollfeld [Pen User CoordSys] zum Zeichnen der Bahn relativ zum evtl. bewegten Koordinatensystem des Roboters (gezeichnete Bahn bewegt sich mit dem Koordinatensystem) aktivieren. Über die [Del]-Schaltflächen werden die gezeichneten Linien gelöscht.
[8]	3D-Modell ein- und ausblenden. Über [Transp] und [Solid] kann zwischen halbtransparente und solider Darstellung umgeschaltet werden.
[9]	Kartesischen Arbeitsraum ein- und ausblenden
[10]	Positionswerte der Achsen ein- und ausblenden
[11]	Verschiedenen Koordinatensysteme ein- und ausblenden
[12]	Koordinatensysteme größer und kleiner skalieren
[13]	Status der Kommunikationsverbindung zwischen 3D-Simulation und MOVI-C® CONTROLLER
[14]	Konstellation des Roboters ein- und ausblenden.
[15]	3D-Simulation des Kinematikmodells
[16]	Aktuelle translatorische Geschwindigkeit des TCP
[17]	Aktuelle Koordinaten des TCP bezogen auf das Base-Koordinatensystem

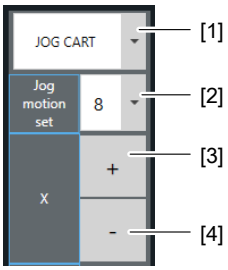
26873346/DE – 07/2021

10.2.5 Programmstatus und Steuerung des Programmbetriebs



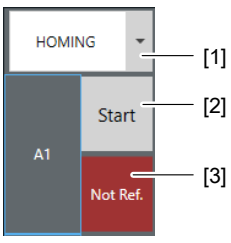
Nr.	Beschreibung	IEC-Name
[1]	Auswahlliste für Programme	Prg.IN.uiProgramNumber
[2]	Schrittbetrieb	PRG.IN.eMode
[3]	Start	Prg.IN.xStart
[4]	Pause	Prg.IN.xPause
[5]	Stop	Prg.IN.xPause

10.2.6 Steuerung des Tippbetriebs



Nr.	Beschreibung	IEC-Name
[1]	Auswahl JOG-Koordinatensystem	Jog.IN.Kinematic.eCoordinateSystem
[2]	Auswahl der Bewegungsparameter	Jog.IN.usiMotionSet (JOG_JOINT/CART) Jog.IN.usiMotionSet_Axes (JOG_AXES)
[3]	Plus-Button für den Tipp-Betrieb	Jog.IN.Kinematic.axPositive
[4]	Minus-Button für den Tipp-Betrieb	Jog.IN.Kinematic.axNegative

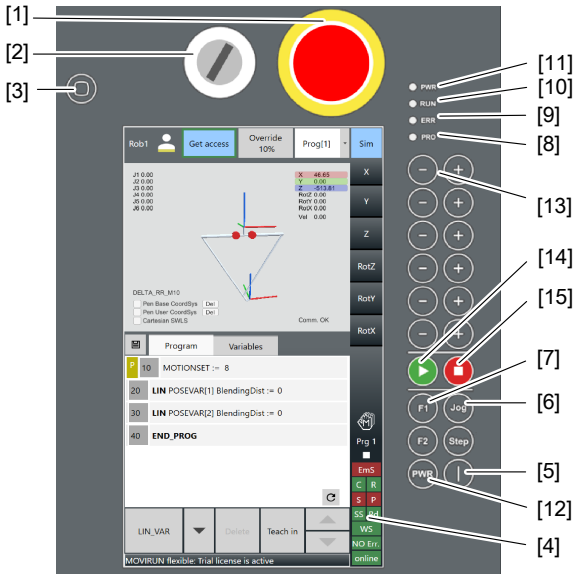
10.2.7 Steuerung des Referenzierbetriebs



Nr.	Beschreibung	IEC-Name
[1]	Auswahl des Referenzierbetriebs	-
[2]	Start-Button für den Referenzierbetrieb	Homing.In.axStart
[3]	Status des Referenzierens der Achse	Inverter.Out.axReferenced

26873346/DE – 07/2021

10.2.8 Bedienpanel



Nr.	Beschreibung
[1]	Nothalt-Schalter des Bedienpanels - Schalter muss extern verdrahtet werden. Weitere Informationen dazu finden Sie in der Dokumentation des Bedienpanels.
[2]	Schlüsselschalter des Bedienpanels zum Umschalten der Betriebsart des Roboters (Betriebsartenwahlschalter) - Schalter muss mit dem MOVI-C® CONTROLLER und ggf. der Sicherheitssteuerung verdrahtet werden. Mehr dazu in der Dokumentation des Bedienpanels und im Kapitel "Externen Anwahlschalter für die Betriebsarten verwenden" (→ 🗎 232).
[3]	Taster zum Umschalten zwischen folgenden Ansichten des RobotMonitors: - Splitscreen zwischen 3D-Simulation und Programmanzeige - Fullscreen 3D-Simulation - Fullscreen Programmanzeige
[4]	Feld mit Statusinformationen zum Roboter Durch Antippen des Feldes wird ein detailliertes Statusfeld aufgeklappt.
[5]	Taster zum Ein- und Ausschalten des Bedienpanels Die Aktion (Herunterfahren, Reboot, …), die beim Drücken des Tasters ausgeführt werden soll, kann in den Systemeinstellungen des Bedienpanels eingestellt werden. Mehr dazu in der Dokumentation des Bedienpanels.
[6]	Taster zum Umschalten des JOG-Koordinatensystems und Anwahl des Referenzierbetriebs
[7]	Freie Taster zum Auswerten im IEC-Programm - Weitere Informationen dazu finden Sie im Kapitel "Freie Funktionstasten des Bedienpanels verwenden" (→ 🗎 233).
[8]	Status LED: Programmbetrieb aktiv
[9]	Status LED: Fehlerstatus
[10]	Status LED: Verbindung zwischen RobotMonitor und MOVI-C® CONTROLLER
[11]	Status LED: Bedienpanel ist bestromt
[12]	PWR: Taster zum Freigeben des Roboters

26873346/DE – 07/2021

Nr.	Beschreibung
[13]	Taster zum Tippen des Roboters ("+" und "-" Tasten) Die Art des Tippens (kartesisch, gelenkweise, achsweise) ist abhängig von der über den JOG-Taster [6] ausgewählten Betriebsart und wird in der Beschriftung links neben den Tastern angezeigt. Wenn über den JOG-Taster der Referenzierbetrieb angewählt wurde, kann über die Tippen-Taster das Referenzieren der entsprechenden Achse gestartet werden.
[14]	Taster zum Starten des SRL-Programms Im Automatikbetrieb durch erneutes Betätigen das Programm pausieren.
[15]	Taster zum Stoppen und Zurücksetzen des SRL-Programms.

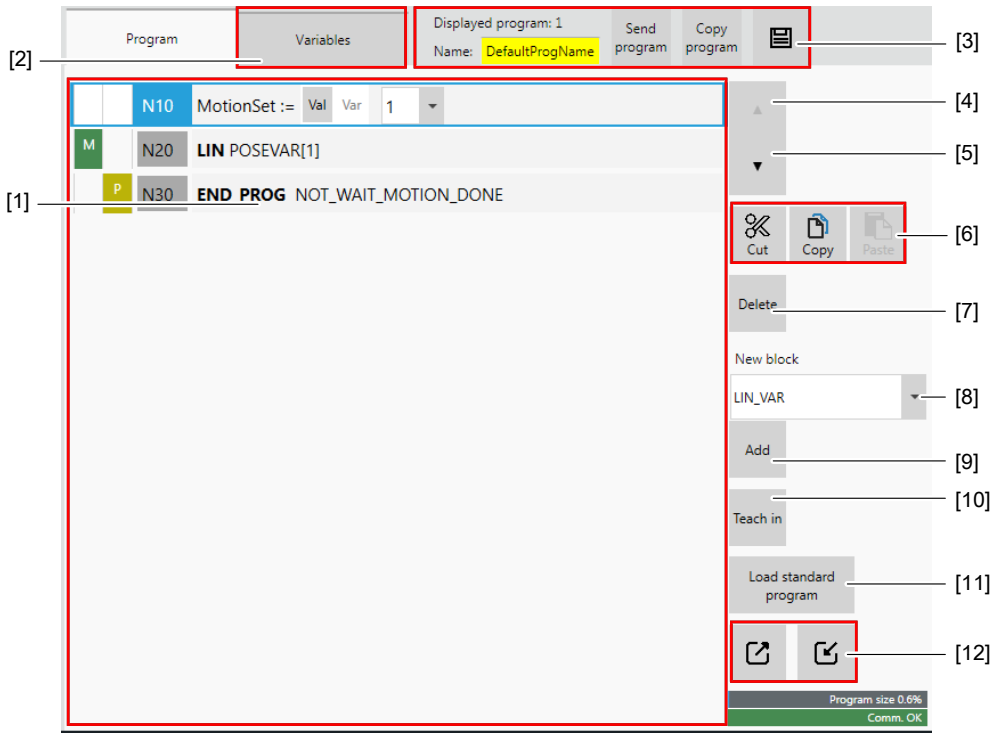
26873346/DE – 07/2021

10.3 SRL-Programm erstellen

Die Erstellung von Robotergrogrammen erfolgt im Register "Program".

HINWEIS

Um das Verlieren von ungespeicherten Änderungen zu vermeiden, speichern Sie unbedingt das SRL-Programm, bevor Sie die Codegenerierung, ein Konfigurations-Download oder ein Neustart des MOVI-C® CONTROLLER durchführen.



9007230943365131

Nr.	Beschreibung
[1]	Dem SRL-Programm hinzugefügte Zeilen
[2]	Register zum Bearbeiten der "Variablen" (→ 🗎 186)
[3]	Optionen zum "Speichern" (→ 🗎 152)/"Kopieren" (→ 🗎 149) des Programms
[4]	Befehl eine Zeile nach oben verschieben
[5]	Befehl eine Zeile nach unten verschieben
[6]	Befehl ausschneiden/kopieren/einfügen **Hinweis:** Auch über Strg+x, Strg+c und Strg+v ansteuerbar.
[7]	Ausgewählten Befehl löschen
[8]	Auswahl des hinzuzufügenden Blocks
[9]	Ausgewählten Block hinzufügen
[10]	Aktuelle Position speichern ("teachen")
[11]	"Standardprogramm laden" (→ 🗎 150)
[12]	"SRL-Programm importieren/exportieren" (→ 🗎 157)

10.3.1 Neues Programm erstellen

✓ Sichtbares Programm ist beendet oder initialisiert. Siehe Kapitel "Programmbetrieb" (→ 🗎 37).

1. Wechseln Sie auf die Registerkarte "Program".
2. Wechseln Sie das aktive Programm auf das neue Programm.

20621267083

   ⇨ Das Programm wird angezeigt.

3. Geben Sie dem neuen Programm einen Namen.
4. Fügen Sie die Bewegungsbefehle, Zuweisungen und Kontrollstrukturen ein, indem Sie in der Auswahlliste "New Block" den entsprechenden Wert auswählen und anschließend auf die Schaltfläche [Add] klicken. Wiederholen Sie diesen Vorgang für alle Befehle, die Ihr Programm beinhalten soll. Weitere Informationen über die Befehle finden Sie in den nachfolgenden Kapiteln.
5. Fügen Sie ein Programmende ein. Wählen Sie in der Auswahlliste "New Block" den Wert "END_PROG". Klicken Sie auf die Schaltfläche [Add].
6. Um das Programm und die SRL-Programmvariablen dauerhaft auf der Speicherkarte zu speichern, klicken Sie auf das Speichersymbol.

10.3.2 Vorhandenes Programm bearbeiten

✓ Angezeigtes Programm ist beendet oder initialisiert. Siehe Kapitel "Programmbetrieb" (→ 🗎 37).

1. Wechseln Sie zum Register "Program".
2. Wechseln Sie zu dem Programm, das Sie bearbeiten möchten.

   ⇨ Das Programm wird angezeigt.

3. Fügen Sie die gewünschten Befehle hinzu: Markieren Sie die Zeile, unter der Sie den neuen Befehl einfügen wollen. Wählen Sie in der Auswahlliste "New Block" den entsprechenden Wert. Klicken Sie auf die Schaltfläche [Add].
4. Konfigurieren Sie die markierte Zeile.

   ⇨ Mit den Pfeiltasten verschieben Sie die markierte Zeile nach oben oder unten.

   ⇨ Mit der Schaltfläche [Delete] löschen Sie markierte Zeilen.

5. Um das Programm und die SRL-Programmvariablen dauerhaft auf der Speicherkarte zu speichern, klicken Sie auf das Speichersymbol.

26873346/DE – 07/2021

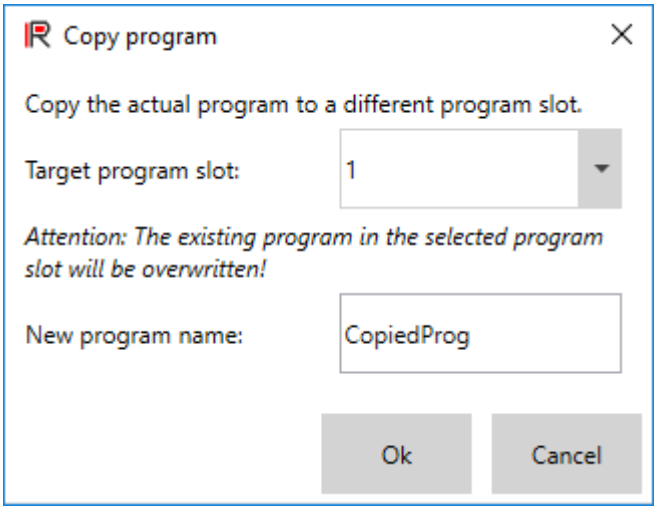
10.3.3 Vorhandenes Programm kopieren

✓ Angezeigtes Programm ist beendet oder initialisiert. Siehe Kapitel "Programmbetrieb" (→ 🗎 37).

1. Wechseln Sie zum Register "Program".
2. Wechseln Sie zu dem Programm, das Sie kopieren möchten.
   ⇨ Das Programm wird angezeigt.
3. Klicken Sie auf die Schaltfläche [Copy program].

18014423429246987

⇨ Das Dialogfenster "Copy program" wird angezeigt.



31368176139

4. Wählen Sie im Auswahlfeld "Target program slot" die Programmnummer aus, auf welche das aktuell angezeigte Programm kopiert werden soll. Das bestehende Programm mit dieser Nummer wird dabei überschrieben.
5. Geben Sie einen Namen für das neue Programm ein.
6. Bestätigen sie den Kopiervorgang mit [Ok]
   ⇨ Das Programm wird auf die neue Nummer kopiert.

10.3.4 Befehle hinzufügen

1. Markieren Sie die Zeile, unter der Sie den neuen Befehl einfügen wollen.
2. Wählen Sie in der Auswahlliste "New Block" den passenden Wert.
3. Klicken Sie auf die Schaltfläche [Add].
4. Parametrieren Sie den Befehl bei Bedarf. Weitere Informationen dazu erhalten Sie in den folgenden Kapiteln.

26873346/DE – 07/2021

10.3.5 Standardprogramm laden

HINWEIS

Das erzeugte Standard-Programm überschreibt das derzeit angezeigte Programm und die Variablennamen.

Mit der Schaltfläche [Load standard program] kann ein Standard-SRL-Programm erzeugt werden.

Es öffnet sich ein Dialog, in welchem ausgewählt werden kann, wie viele Standard-Bahnsegmente im erstellten Programm enthalten sein sollen. Nach dem Bestätigen des Dialogs mit [OK], wird das Standardprogramm automatisch erzeugt und die Namen der darin verwendeten Variablen gesetzt.

Funktionsweise

Ein Standard-Programm enthält eine bestimmte Anzahl von Standard-Bahnsegmenten. Diese Segmente können über Roboter-Programmvariablen parametriert werden. So ist es möglich, das Programm durch eine Prozess-Steuerung von außen zu bedienen, z. B. über die Feldbus-Schnittstelle oder die IEC-Anwenderschnittstelle des Softwaremoduls.

Jedes Standard-Bahnsegment umfasst folgende Parameter, die über Variablen jeweils vorgegeben werden können:

- Zielpose des Bahnsegments
- Überschleifdistanz auf das Bahnsegment
- Nummer des zu verwendenden MotionSets für das Bahnsegment
- Ende-Signal um zu signalisieren, dass nach diesem Bahnsegment das Programm beendet werden soll
- Warte-Signal zum Veranlassen, dass der Roboter das Bahnsegment noch nicht ausführt bis das Wartesignal weggenommen wird

26873346/DE – 07/2021

```
N10  MotionSet := MotionSet_Segment1
N20  BlendingDistance := Blending_Segment1 [mm]
N30  WAIT NOT Wait1
N40  LIN Pose_Segment1
N50  IF End1
N60      END_PROG
N70  END_IF
N80  MotionSet := MotionSet_Segment2
N90  BlendingDistance := Blending_Segment2 [mm]
N100 WAIT NOT Wait2
N110 LIN Pose_Segment2
N120 IF End2
N130     END_PROG
N140 END_IF
N150 MotionSet := MotionSet_Segment3
```

Standard-Bahnsegment 1

Standard-Bahnsegment 2

32232154379

HINWEIS

Um wirksam zu sein, müssen die Variablen zum Zeitpunkt des Programmstarts gesetzt sein. Auch Wartesignale müssen von Anfang an gesetzt sein, damit der Roboter an der gewünschten Stelle pausiert. Die Wartesignale können während der Programmabarbeitung zurückgesetzt werden.

26873346/DE – 07/2021

10.4 SRL-Programm speichern

HINWEIS

Wenn Sie Programme und Variablen nicht speichern, gehen die vorgenommenen Änderungen bei einem Reset bzw. Reboot der Steuerung, beim "Generieren des IEC-Projekts" (→ 🗎 105) oder beim Aktualisieren der Konfigurationsdaten verloren und können nicht wiederhergestellt werden. Wenn noch nicht gespeicherte Änderungen an Programmen oder Variablen vorliegen, ist dies mit einem gelben Punkt an der Schaltfläche [Speichern] (Diskettensymbol) gekennzeichnet.

1. Klicken Sie auf die Schaltfläche [Speichern] (Diskettensymbol)

⇨ Alle Programme und Variablen inklusive Motionsets des Roboters werden permanent gespeichert. Dadurch werden die erstellten Programme und Variablen nach dem Aus- bzw. Einschalten des MOVI-C® CONTROLLER oder einem Reset des Steuerungsprogramms wieder neu geladen. Die Daten werden dazu in Dateien auf der Speicherkarte des MOVI-C® CONTROLLER gesichert und beim erneuten Start des MOVI-C® CONTROLLER wieder eingelesen.

26873346/DE – 07/2021

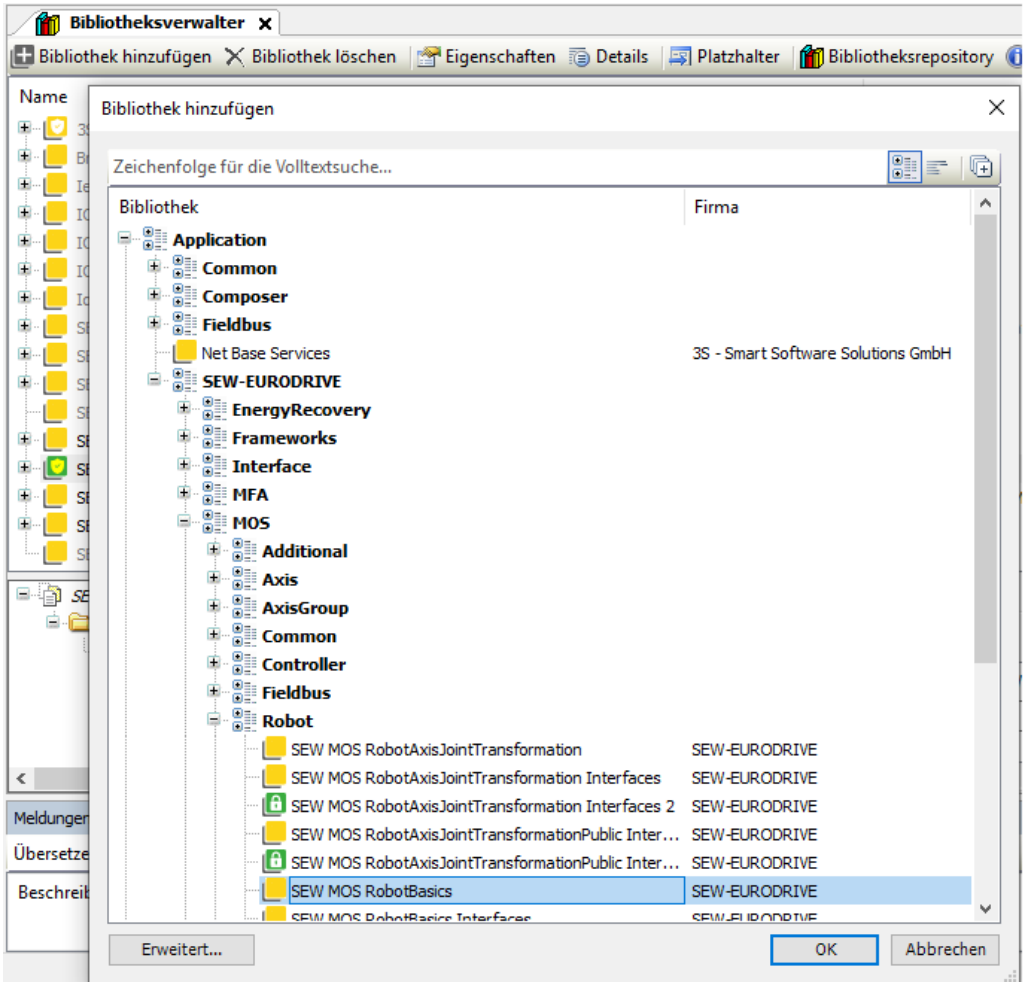
10.5 Zusätzliche (große) SRL-Programme

Standardmäßig stehen dem Anwender 20 SRL-Programme zur Verfügung. Jedes SRL-Programm kann eine bestimmte Menge an Befehlen enthalten. Der durch das aktuelle Programm belegte Speicherplatz wird auf der Benutzeroberfläche des Robot-Monitor in der rechten unteren Ecke des Programm-Editors angezeigt. Programme, die größer sind, können zwar angezeigt, exportiert und importiert, aber nicht in einem der standardmäßig vorhandenen Programm-Slots 1-20 auf dem MOVI-C® CONTROLLER gespeichert werden.

Für SRL-Programme, die nicht in einen der Programm-Slots 1-20 passen, können zusätzliche SRL-Programme angelegt und größer konfiguriert werden. Anzahl und Größe dieser zusätzlichen Programme sind durch den freien Arbeitsspeicher des MOVI-C® CONTROLLER begrenzt.

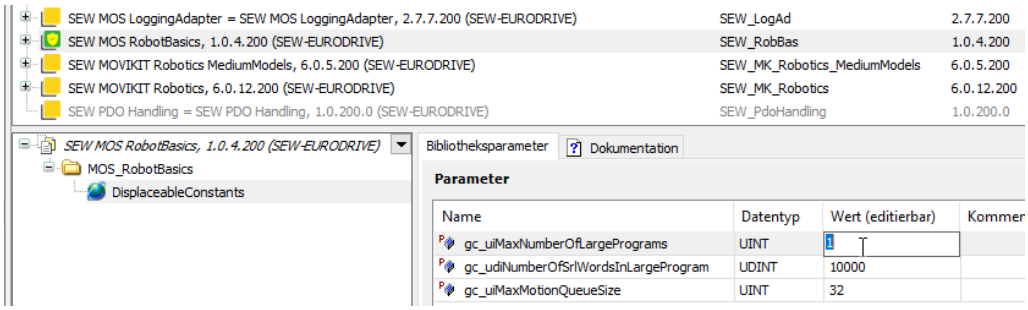
10.5.1 Zusätzliche SRL-Programme anlegen

Um ein zusätzliches SRL-Programm anzulegen, muss in MOVISUITE® der IEC-Editor geöffnet und die Programmbibliothek "SEW MOS RobotBasics" auf höchster Ebene mithilfe des Bibliotheksverwalters wie in folgender Grafik zu sehen eingebunden werden.



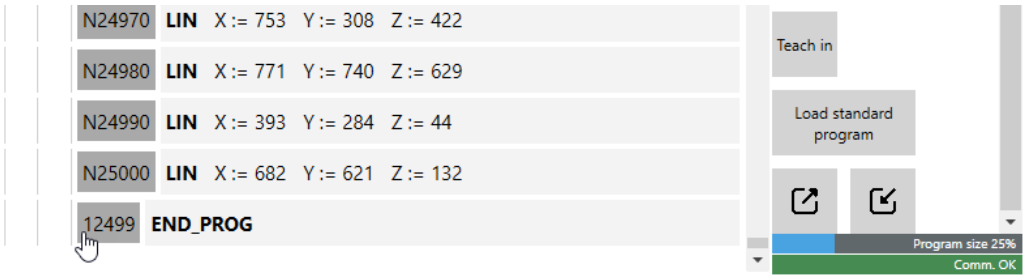
34939178251

Nach dem Einbinden kann die eingebundene Programmbibliothek im Bibliotheksverwalter angewählt und die Konfiguration der zusätzlichen Programme verändert werden. Die zusätzlichen SRL-Programme befinden sich im Eintrag "DisplaceableConstants". Das Editieren der zusätzlichen SRL-Programme erfolgt im Register "Bibliotheksparameter".



34939180683

Mit der Konstante *gc_uiMaxNumberOfLargePrograms* wird die Anzahl der zusätzlichen SRL-Programme und mit der Konstante *gc_udiNumberOfSrlWordsInLargePrograms* die Anzahl der Worte je zusätzlichem SRL-Programm festgelegt. Jeder eingefügte Befehl verbraucht (abhängig von seiner Konfiguration) eine unterschiedliche Anzahl an Worten. Die minimale Anzahl der Worte die eingestellt werden sollten, kann ermittelt werden, indem der Anwender mit der Maus über den Blockindex fährt. Die ohne den Buchstaben "N" angezeigte Zahl ist die Position des Befehls in Worten. In der nachfolgenden Grafik sind im IEC-Editor beispielsweise 50000 Worte konfiguriert. Das SRL-Programm im Programm-Editor verbraucht aktuell mit dieser Einstellung mindestens 12500 Worte. Daraus ergibt sich eine Programmbelegung von 25 %.



34939183115

Werden nachträglich mehr Worte benötigt, kann die Anzahl in der Konstante *gc_udiNumberOfSrlWordsInLargePrograms* noch erhöht werden. Ein nachträgliches Verkleinern ist allerdings nur über die im nachfolgenden Kapitel erläuterte Vorgehensweise möglich.

Wenn bereits ein zusätzliches SRL-Programm auf der Speicherkarte vorhanden ist und dieses größer als in den *DisplacebalConstants* eingestellt ist, erhält der Anwender die eine entsprechende Fehlermeldung, sobald er sich mit dem RobotMonitor verbindet. In diesem Fall muss die Anzahl an Worten in der Konstanten *gc_udiNumberOfSrlWordsInLargePrograms* erhöht und ein erneuter Download des IEC-Programms auf den MOVI-C® CONTROLLER durchgeführt werden.

34939185547

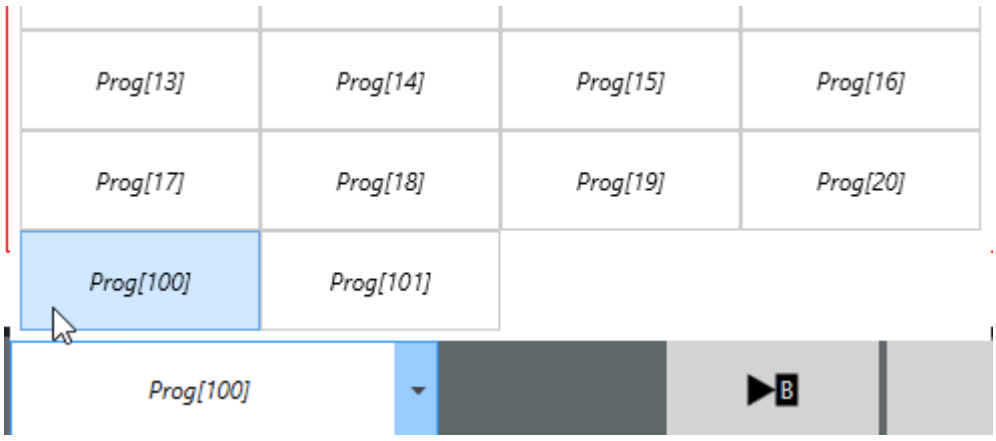
26873346/DE – 07/2021

Die Vorgehensweise zum Hinzufügen zusätzlicher SRL-Programme im Überblick (Wenn Sie die Anzahl an benötigten Worten bereits kennen, können Sie ab Punkt 4 beginnen):

1. Erstellen und exportieren Sie ein SRL-Programme ohne Verbindung zum MOVI-C® CONTROLLER in einem der Programm-Slots 1-20. Alternativ können Sie auch ein bereits vorhandenes SRL-Programm importieren.
2. Stellen Sie den Mauszeiger auf den letzten Programmbefehl des SRL-Programms, um im Tooltip den Wortindex abzulesen (z. B. N12510).
3. Addieren Sie zum abgelesenen Wortindex 250 Wörter (Puffer für folgende Bearbeitung).
4. Öffnen Sie über die MOVISUITE® den IEC-Editor.
5. Öffnen Sie den "Bibliotheksverwalter" und klicken Sie auf [Bibliothek hinzufügen]
6. Navigieren Sie im Bibliotheksverwalter zur Bibliothek "SEW MOVIKIT Robotics" > "SEW MOS Robot" > "SEW MOS RobotLanguage" > "SEW MOS RobotBasics" und notieren Sie die Versionsnummer.
7. Fügen Sie die Bibliothek "SEW MOS RobotBasics" in der Version aus dem vorherigen Schritt hinzu (Application > SEW-EURODRIVE > MOS > Robot).
8. Klicken Sie auf die hinzugefügte Bibliothek und passen Sie in *DisplacebalConstants* die Konstanten *gc_uiMaxNumberOfLargePrograms* und *gc_udiNumberOfSrlWordsInLargePrograms* an.
9. Führen Sie im IEC-Editor ein Einloggen mit Download aus.
10. Starten Sie das IEC-Programm.
11. Verbinden Sie den RobotMonitor mit dem MOVI-C® CONTROLLER, fordern Sie Zugriff an (Get access control) und wählen Sie das gewünschte Zielprogramm aus.
12. Importieren bzw. erstellen und senden Sie das SRL-Programm zum MOVI-C® CONTROLLER.

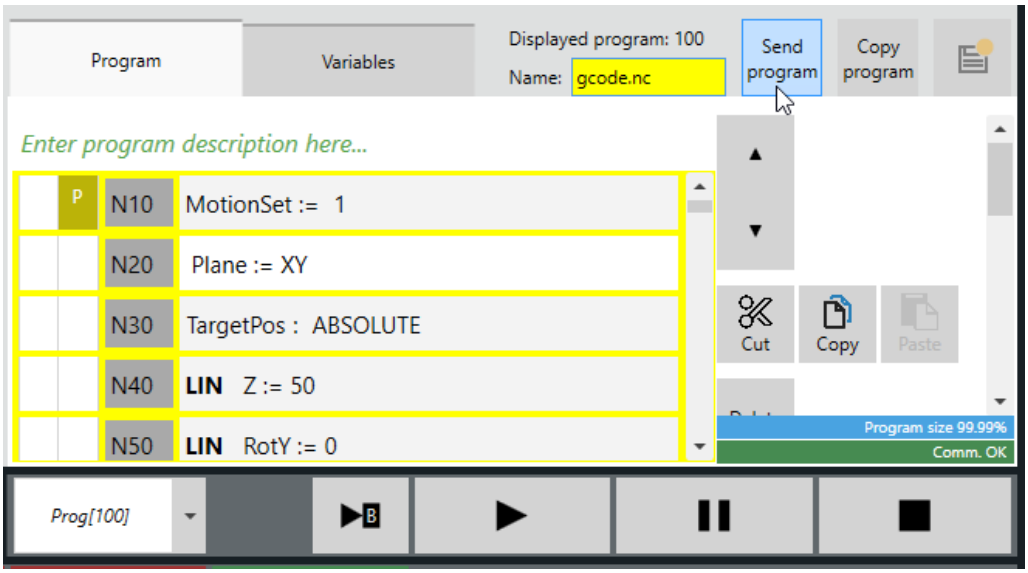
10.5.2 Zusätzliche SRL-Programmen verwenden

Die zusätzlichen SRL-Programme werden zur Auswahl im RobotMonitor ab dem Index 100 angezeigt und können auch über die IEC-Anwenderschnittstelle sowie die Feldbusschnittstelle angewählt werden (Index 100+)



34939187979

Wird eine SRL-Programm-Datei auf einen der Programm-Slots 1-20 importiert, wird es automatisch auf den MOVI-C® CONTROLLER übertragen. Voraussetzung dafür ist, dass kein Fehler im SRL-Programm vorliegt. Die zusätzlichen SRL-Programme werden nicht automatisch auf den MOVI-C® CONTROLLER übertragen. Nach einer Änderung oder nach dem Importieren eines zusätzlichen SRL-Programms, muss das Senden auf den MOVI-C® CONTROLLER manuell ausgelöst werden. Ein nicht gesendetes oder geändertes zusätzliches SRL-Programm wird auf der Benutzeroberfläche des RobotMotitors gelb gekennzeichnet. Das Senden des zusätzlichen SRL-Programms an den MOVI-C® CONTROLLER erfolgt über die Schaltfläche [Send Program].



34939190411

10.5.3 Zusätzliche SRL-Programme nachträglich verkleinern

Wenn für weitere zusätzliche SRL-Programme der Arbeitsspeicher nicht mehr ausreicht, besteht die Möglichkeit die vorhandenen zusätzlichen SRL-Programme zu verkleinern. Hierfür müssen alle vorhandenen zusätzlichen SRL-Programme vom MOVI-C® CONTROLLER auf den Engineering-PC verschoben werden, um die *DisplaceableConstants* anzupassen. Die neu eingestellte Größe der zusätzlichen SRL-Programme sollte mindestens so groß sein wie das größte zusätzliche SRL-Programm. Die Größen müssen vorher im RobotMonitor ermittelt werden.

Nach dem Anpassen der Konstanten müssen die zusätzlichen SRL-Programme mithilfe des RobotMonitors auf den gewünschten Programmindex importiert, gesendet und gespeichert werden.

26873346/DE – 07/2021

10.6 SRL-Programm importieren/exportieren

Der RobotMonitor bietet die Möglichkeit das aktuell angezeigte SRL-Programm sowie alle Variablen auf die Festplatte des Engineering-PCs zu exportieren bzw. dorthin exportierte SRL-Programme zu importieren.

Das Exportieren bzw. Importieren von SRL-Programmen und Variablen kann dabei unabhägig voneinander ausgeführt werden. Das SRL-Programm sowie die Variablen werden beim Export im selben Format abgespeichert, wie diese auf der Speicherkarte des MOVI-C® CONTROLLER abgelegt werden. Somit können exportierte SRL-Programme und Variablen auch mithilfe des IEC-Editors und dessen Dateimanager, direkt auf die Speicherkarte des MOVI-C® CONTROLLER übertragen werden.

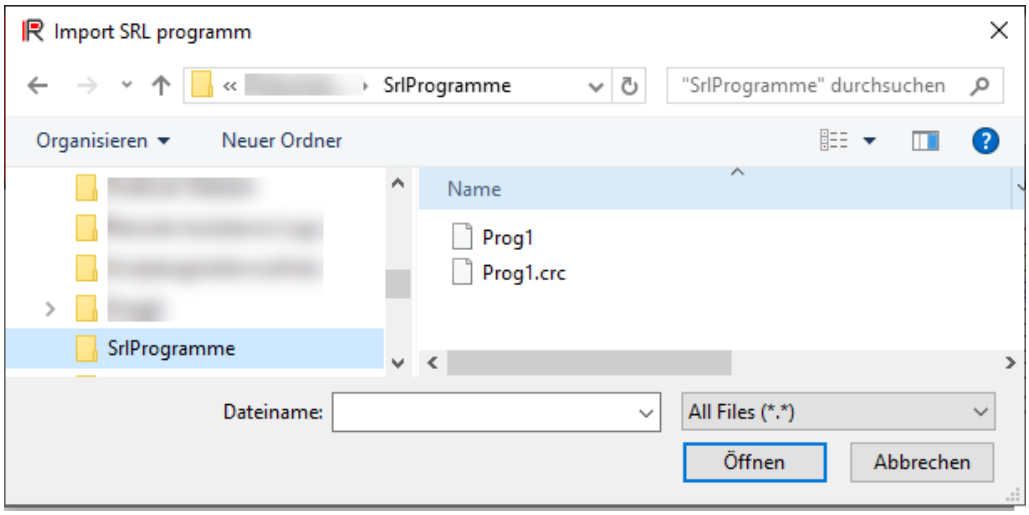
Im- und Exporte können auch ohne bestehende Verbindung mit dem MOVI-C® CONTROLLER ausgeführt werden. Damit ist es möglich SRL-Programme und Variablen ohne MOVI-C® CONTROLLER zu speichern, zu laden, zu bearbeiten und zu versionieren. Damit ist ein komfortabler Austausch aller Daten des RobotMonitor zwischen beispielsweise Mitwirkenden eines Projekts gewährleistet.

26873346/DE – 07/2021

10.6.1 SRL-Programm importieren

✓ Der RobotMonitor ist geöffnet.

1. Öffnen Sie im RobotMonitor das Register "Program".
2. Klicken Sie im Register "Program" auf die Schaltfläche  . Siehe auch "SRL-Programm erstellen" (→ 🗎 147).
3. Wählen Sie die Datei des zu importierende SRL-Programms aus und klicken Sie auf [Öffnen].
   ⇨ Die zum SRL-Programm gehörende CRC-Datei muss sich im selben Verzeichnis wie das zu importierende SRL-Programm befinden.
   ⇨ Wenn eine Kommentardatei (*.srlcomments) vorhanden ist und ebenfalls importiert werden soll, muss sich diese auch im selben Verzeichnis befinden. Weitere Informationen finden Sie im Kapitel "Kommentare importieren/exportieren" (→ 🗎 162).



33223936779

4. Bestätigen Sie ggf. die Meldung, dass das bestehende Roberterprogramm überschrieben werden soll.

10.6.2 SRL-Programm exportieren

✓ Der RobotMonitor ist geöffnet.

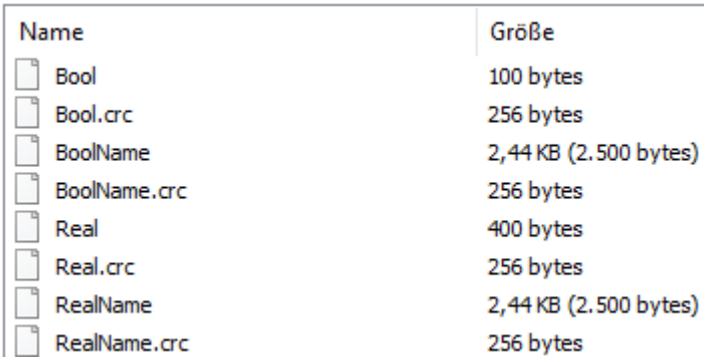
1. Öffnen Sie im RobotMonitor das Register "Program".
2. Klicken Sie im Register "Program" auf die Schaltfläche  . Siehe auch "SRL-Programm erstellen" (→ 🗎 147).
3. Wählen Sie den Speicherort für die Datei des zu exportierenden Roberterprogramms aus, geben Sie einen Dateinamen ein und klicken Sie auf [Öffnen].
   ⇨ Exportierte SRL-Programme werden als Datei ohne Dateiendung auf der Festplatte gespeichert.
4. Bestätigen Sie ggf. die Meldung, dass die bei einem vorherigen SRL-Programm-Export erstellte Datei überschrieben werden soll.

26873346/DE – 07/2021

10.6.3 SRL-Variablen importieren

✓ Der RobotMonitor ist geöffnet.

1. Öffnen Sie im RobotMonitor das Register "Variables".
2. Klicken Sie im Register "Variables" auf die Schaltfläche  . Siehe auch "Variablen" (→ 🗎 186).
3. Wählen Sie das Verzeichnis mit den zu importierenden SRL-Variablen aus und klicken Sie auf [Öffnen].
   ⇨ Im selben Verzeichnis wie die Datei mit den SRL-Variablen muss sich die dazugehörige CRC-Datei sowie die Namens-Datei, ebenfalls mit dazugehöriger CRC-Datei, befinden. Folgender Screenshot zeigt beispielhaft die korrekte Benennung der SRL-Variablen-Dateien.



9007232478755211

4. Bestätigen Sie ggf. die Meldung, dass die bestehenden Variablen überschrieben werden sollen.

10.6.4 SRL-Variablen exportieren

✓ Der RobotMonitor ist geöffnet.

1. Öffnen Sie im RobotMonitor das Register "Variables".
2. Klicken Sie im Register "Variables" auf die Schaltfläche  . Siehe auch "Variablen" (→ 🗎 186).
3. Wählen Sie ein Ziel-Verzeichnis für die zu exportierenden Dateien aus, geben Sie einen Dateinamen ein und klicken Sie auf [Öffnen].
   ⇨ Die exportierten SRL-Variablen werden als Datei ohne Dateiendung auf der Festplatte gespeichert.
   ⇨ Die CRC-Datei und die Namens-Datei der exportierten SRL-Variablen werden automatisch im selben Verzeichnis mit demselben Namen wie die Variablen-Datei gespeichert.
4. Bestätigen Sie ggf. die Meldung, dass die bei einem vorherigen Variablen-Export erstellte Datei überschrieben werden soll.

26873346/DE – 07/2021

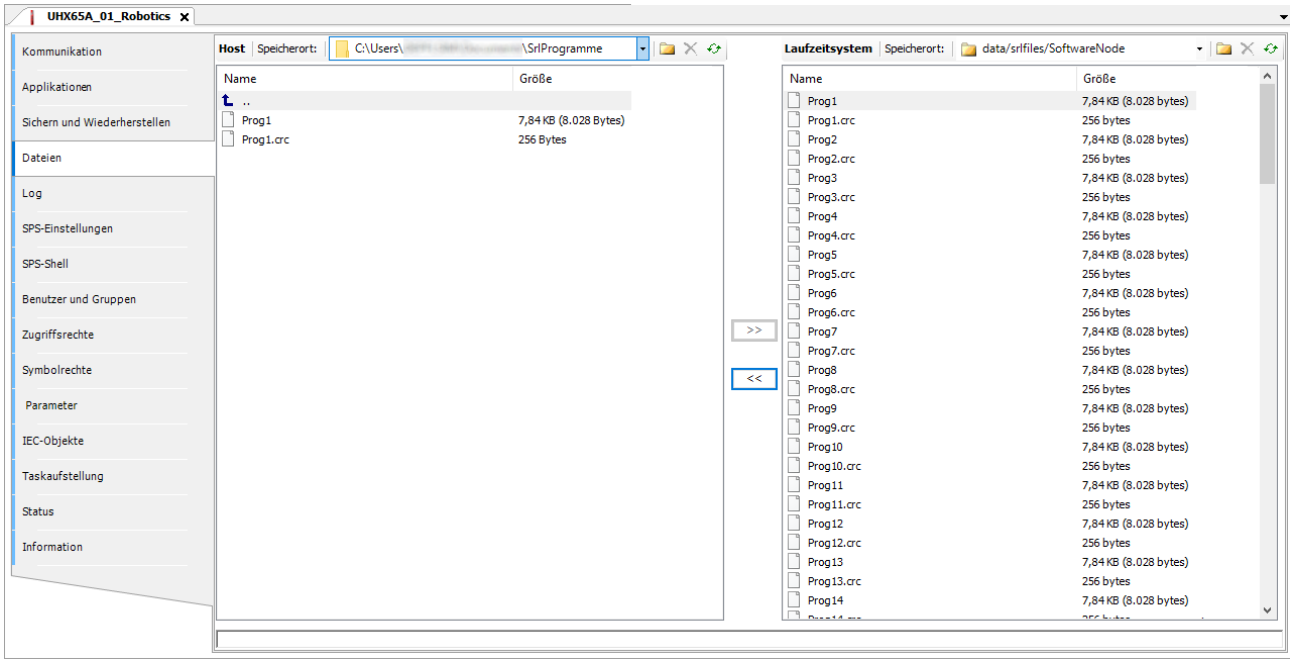
10.6.5 Übertragung zwischen Speicherkarte und Festplatte

Mit dem CODESYS-Dateimanager im IEC-Editor können vom RobotMonitor exportierte Dateien von der Festplatte auf die Speicherkarte des MOVI-C® CONTROLLER übertragen werden. Umgekehrt können auch Dateien von der Speicherkarte auf die Festplatte übertragen und dann in den RobotMonitor importiert werden.

Dabei gilt es folgendes zu beachten:

- Beim Übertragen von Engineering-PC auf die Speicherkarte muss die CRC-Datei jeweils auch übertragen werden und genauso benannt sein, wie die Datei mit dem SRL-Programm bzw. den SRL-Variablen.
- Beim Speichern auf die Speicherkarte müssen die Dateien mit den SRL-Programmen bzw. den SRL-Variablen und die dazugehörigen CRC-Dateien die ursprünglich gewählten Namen (z. B. `Prog1` bis `Prog20`) haben.
- Beim Übertragen von Variablen-Dateien muss zusätzlich beachtet werden, dass die Names- und Names.crc-Datei des jeweiligen Variablentyps ebenfalls übertragen wird.
- Die SRL-Programme und SRL-Variablen werden beim Speichern durch den RobotMonitor an folgendem Speicherort abgelegt:

  `/data/srlfiles/ <Name des Softwareknotens in MOVISUITE>`



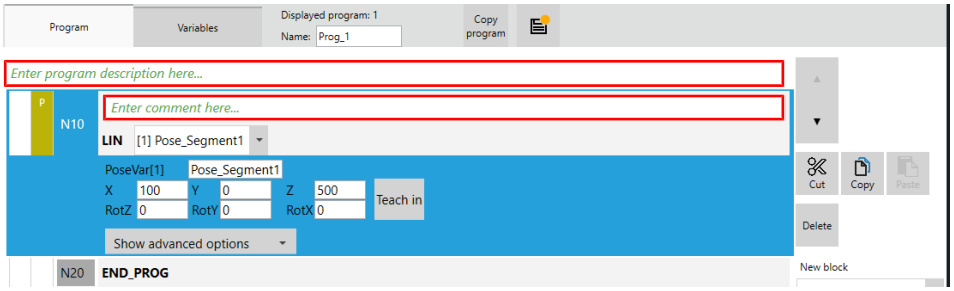
33224157579

26873346/DE – 07/2021

10.7 SRL-Programm kommentieren

Kommentare dienen dazu das SRL-Programm zu dokumentieren und es somit verständlicher zu machen. Die Kommentare sind standardmäßig eingeblendet, können aber über die Einstellungen unter dem Register "Editor" ausgeblendet werden.

Der RobotMonitor bietet die Möglichkeit einen Kommentar für das gesamte SRL-Programm und Kommentare je Block anzulegen. Dafür befinden sich auf der Benutzeroberfläche entsprechende Eingabefelder. Sind noch keine Kommentare angelegt, enthalten die für die Kommentare vorgesehenen Eingabefelder einen entsprechenden Hinweistext. Das Kommentar-Eingabefeld eines Blocks wird jeweils sichtbar, wenn der Block selektiert ist.



33774344331

10.7.1 Kommentare anlegen

Gehen Sie zum Anlegen von Kommentaren in Roberprogrammen wie folgt vor:

✓ Der Benutzer hat Zugriff auf den Roboter.

✓ Die Ausführung des SRL-Programms ist gestoppt.

✓ Das Register "Program" ist geöffnet.

1. Klicken Sie in das Kommentar-Eingabefeld des Roberprogramms (Hinweistext "*Enter program describiton here...*") oder selektieren Sie einen Block im Roboterprgramm und klicken Sie anschließend in dessen Kommentar-Eingabefeld (Hinweistext "*Enter comment here...*").
2. Geben Sie den gewünschten Text in das Kommentar-Eingabefeld ein.
   - ⇨ Zeilenumbrüche werden über die Tastenkombination [SHIFT]+[ENTER] erzeugt.
   - ⇨ Lange Block-Kommentare werden auf der Benutzeroberfläche auf 3 Zeilen zusammengeklappt und können über die Schaltfläche [Expand comment] auf- und die Schaltfläche [Shrink comment] zusammengeklappt werden.
   - ⇨ Wenn die maximal zulässige Anzahl an Zeichen aller Kommentare erreicht ist, erscheint eine Meldung mit der Info, wie viele Zeichen zu viel verwendet wurden. Wenn die maximal zulässige Anzahl überschritten ist, können die Kommentare nicht gespeichert werden.
3. Bestätigen Sie Ihre Eingabe mit [ENTER].
   - ⇨ Der eingegebene Kommentar wird an den MOVI-C® CONTROLLER übertragen.
4. Speichern Sie zum persistenten Speichern der Kommentare auf der Speicherkarte das SRL-Programm ab. Siehe "SRL-Programm speichern" (→ 🗎 152).

10.7.2 Kommentare importieren/exportieren

Beachten Sie beim Importieren/Exportieren von SRL-Programmen hinsichtlich der Kommentare folgende Hinweise:

- Wenn im SRL-Programm Kommentare angelegt wurden, wird beim Export von SRL-Programmen automatisch eine .srlcomments-Datei mit demselben Namen wie das exportierte Programm angelegt.
- Beim Import von SRL-Programmen muss die .srlcomments-Datei am selben Speicherort wie das SRL-Programm abgelegt sein. Wird beim Import keine .srlcomments-Datei erkannt, wird der Benutzer durch eine entsprechende Meldung darauf hingewiesen. Die Kommentare bleiben in diesem Fall leer bzw. werden ignoriert.
- Beim Übertragen von SRL-Programmen von der Speicherkarte des MOVI-C® CONTROLLER auf den Engineering-PC und umgekehrt muss die .srlcomments-Datei ebenfalls übertragen werden.

26873346/DE – 07/2021

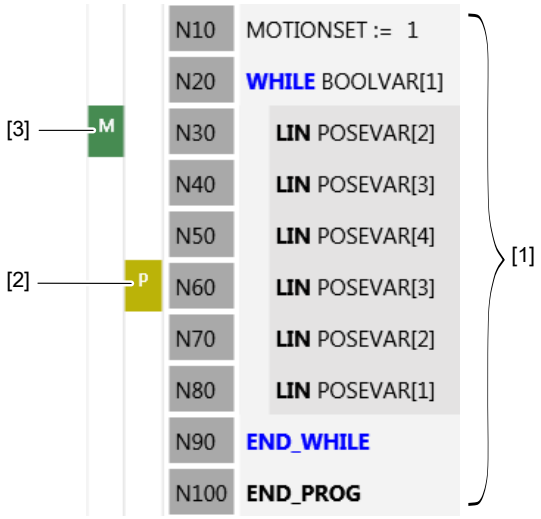
10.8 SRL-Programm ausführen

✓ Ein SRL-Programm wurde erstellt. Siehe Kapitel "SRL-Programm erstellen" (→ 🗎 147).

1. Wechseln Sie zum Register "Program".
2. Wechseln Sie zu dem Programm, das ausgeführt werden soll.

   ⇨ Das Programm wird angezeigt.
3. Starten Sie das SRL-Programm durch Setzen des Signals "Start". Weitere Informationen zur Programmausführung erhalten Sie im Kapitel "Programmbetrieb" (→ 🗎 37).

⇨ Die Ausführung des SRL-Programms startet:



18014423429273995

[1] Befehle des SRL-Programms
[2] Programmzeiger ("P" für "Program")
Zeigt an, welcher Befehl gerade abgearbeitet wird.
[3] Bewegungszeiger ("M" für "Motion")
Zeigt an, welche Bewegung die Bewegungssteuerung gerade bearbeitet, d. h. welches Bahnsegment der Roboter gerade abfährt.

HINWEIS

Beim Starten des Programms wird der "Grundzustand" (→ 🗎 38) hergestellt.

10.9 Posen teachen

Der Robotermonitor bietet eine sogenannte Teach-Funktion. Dabei können im SRL-Programm verwendete Posen in das SRL-Programm (explizites teachen) oder in die Posenvariable (Variable teachen) übernommen werden. In beiden Fällen kann die Pose danach mit dem SRL-Programm angefahren werden.

HINWEIS

Es gelten die festgelegten "Standardeinheiten" (→ 🗎 28).

10.9.1 Posenvariable teachen

Die Teach-Funktion schreibt immer die Pose in der sich der Roboter gerade befindet in die gewünschte Posenvariable. Dabei gibt es folgende Möglichkeiten:

Posenvariable in der Variablenliste teachen

1. Wechseln Sie auf die Registerkarte "Variables".
2. Wechseln Sie auf der Registerkarte "Variables" in die Registerkarte der gewünschten Posenvariablen.
3. Verfahren Sie den Roboter mittels Tippbetrieb zur gewünschten Pose.
4. Markieren Sie in der Liste die Posevariable, die übernommen werden soll
5. Klicken Sie auf [ActPos to Selected PosVar].
   ⇨ Die aktuelle Pose des Roboters wird auf die markierte Variable geschrieben. Die Änderung ist sofort wirksam.
   ⇨ Sie können die Pose nun im SRL-Programm verwenden. Weitere Informationen dazu finden Sie im Kapitel "Linearsegment mit Posenvariable hinzufügen" (→ 🗎 165).
   ⇨ Bei Bedarf können Sie die Variable verändern. Weitere Inormationen dazu erhalten Sie im Kapitel "SRL-Programmvariablen editieren" (→ 🗎 186).

Posenvariable im SRL-Programm teachen

✓ Es gibt ein Linearsegment mit Posevariablen im zu bearbeitenden SRL-Programm.

1. Wechseln Sie auf die Registerkarte "Programm".
2. Verfahren Sie den Roboter mittels Tippbetrieb zur gewünschten Pose.
3. Markieren Sie das Linearsegment mit der Posevariable, die übernommen werden soll.
4. Wählen Sie in der Auswahlliste des Linearsegmentes die Posenvariable aus, die beschrieben werden soll.
5. Klicken Sie auf [Edit].
6. Klicken Sie auf [TeachIn].
   ⇨ Die aktuelle Pose des Roboters wird auf die ausgewählte Variable geschrieben. Die Änderung ist sofort wirksam.
   ⇨ Bei Bedarf können Sie den Befehl verändern. Weitere Informationen dazu finden Sie im Kapitel "Linearsegment mit Posenvariable hinzufügen" (→ 🗎 165).

10.9.2 Explizites teachen mit Einfügen eines Linearsegmentes

1. Verfahren Sie den Roboter mittels Tippbetrieb zur gewünschten Pose.
2. Markieren Sie die Zeile, unter der Sie das neue Linearsegment einfügen möchten.
3. Klicken Sie auf [TeachIn]
   ⇨ Ein Linearsegment wird eingefügt und die aktuelle Pose des Roboters wird übernommen. Die Änderung ist sofort wirksam.
   ⇨ Bei Bedarf können Sie den Befehl verändern. Weitere Informationen dazu finden Sie im Kapitel "Linearsegment mit Posenvariable hinzufügen" (→ 🗎 165).

26873346/DE – 07/2021

10.10 Bewegungsbefehle

Bewegungsbefehle sind Befehle zum Anfahren einer Position.

HINWEIS

Es gelten die festgelegten "Standardeinheiten" (→ 🗎 28).

10.10.1 Linearsegmente

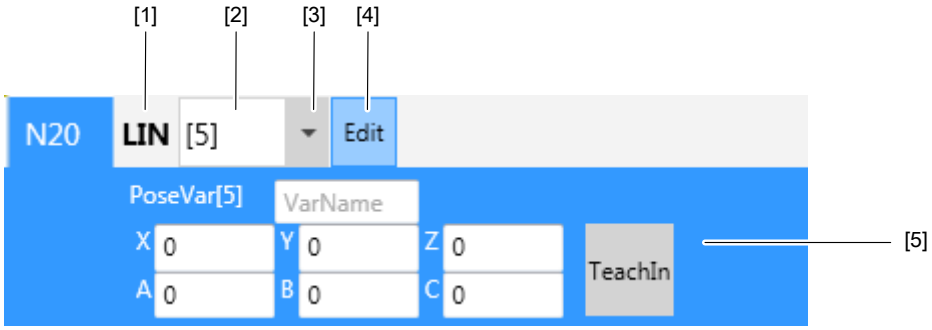
Befehle zum Anfahren einer Zielpose auf einer Geraden mit vorherigem Überschleifen. Die Zielkoordinaten und die Überschleifdistanz können dabei explizit im SRL-Programm oder über IEC-Variablen vorgegeben werden.

LIN Zielpose BlendingDist := Überschleifdistanz

Linearsegment mit Posenvariable hinzufügen

Linearsegment, bei dem die Zielpose über eine SRL-Programmvariable vorgegeben wird.

"New Block"-Wert: "LIN_VAR".



18014423319144459

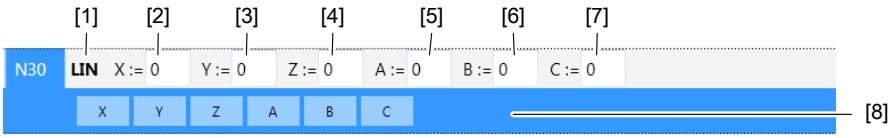
[1] Kennung für ein Linearsegment
[2] Zielpose
[3] Auswahlliste zum Austauschen der Zielpose
[4] Menü zum Verändern des Namens und der Koordinaten der Zielpose
[5] Namen und Koordinaten der Zielpose

26873346/DE – 07/2021

Linearsegment mit expliziter Koordinatenvorgabe hinzufügen

Linearsegment, bei dem die Zielpose explizit im SRL-Programm vorgegeben wird.

"New Block"-Wert: "LIN_EXPLICIT".



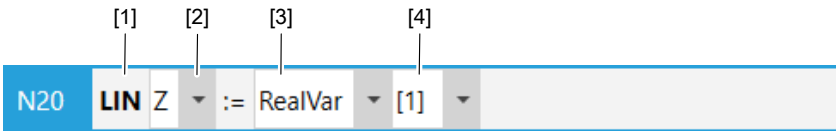
18014423320833035

[1] Kennung für ein Linearsegment
[2] X-Koordinate der Zielpose
[3] Y-Koordinate der Zielpose
[4] Z-Koordinate der Zielpose
[5] Koordinate der Zielpose für die Rotation um Z
[6] Koordinate der Zielpose für die Rotation um Y
[7] Koordinate der Zielpose für die Rotation um X
[8] Aktivieren/Deaktivieren einzelner Koordinaten. Bei Deaktivierung bleibt der Wert auf dem des vorherigen Bewegungsbefehls stehen.

Linearsegment mit Verfahren einer einzelnen Koordinate hinzufügen

Linearsegment, bei dem nur in die Richtung einer Koordinatenachse verfahren werden soll. Die Richtung, in die gefahren werden soll kann ausgewählt werden. Als Zielkoordinate kann ein expliziter Wert, eine REAL-Variable oder die Koordinate einer POSE-Variablen verwendet werden.

"New Block"-Wert: "LIN_SINGLE_COORD".



9007228307326475

[1] Kennung für ein Linearsegment
[2] Auswahl, welche Koordinate verändert werden soll.
[3] Auswahl, ob ein direkter Wert, eine REAL-Variable oder eine Koordinate aus einer POSE-Variablen verwendet werden soll.
[4] Auswahl der Variablen für die Zielkoordinate

26873346/DE – 07/2021

10.10.2 PTP-Segmente

Befehle zum Anfahren einer Zielpose über Punkt-zu-Punkt-Interpolation der Gelenkachsen.

Bestimmte Kinematikmodelle unterstützen mehrere Konstellationen (z. B. Ellbogen links/rechts) und Gelenkachsphasen. Die Konfigurationsmöglichkeiten finden Sie im Kapitel "Kinematikmodelle" (→ 🗎 44). Die kartesischen Zielkoordinaten, die Konstellation und die Gelenkachsphasen können dabei explizit im SRL-Programm oder über IEC-Variablen vorgegeben werden.

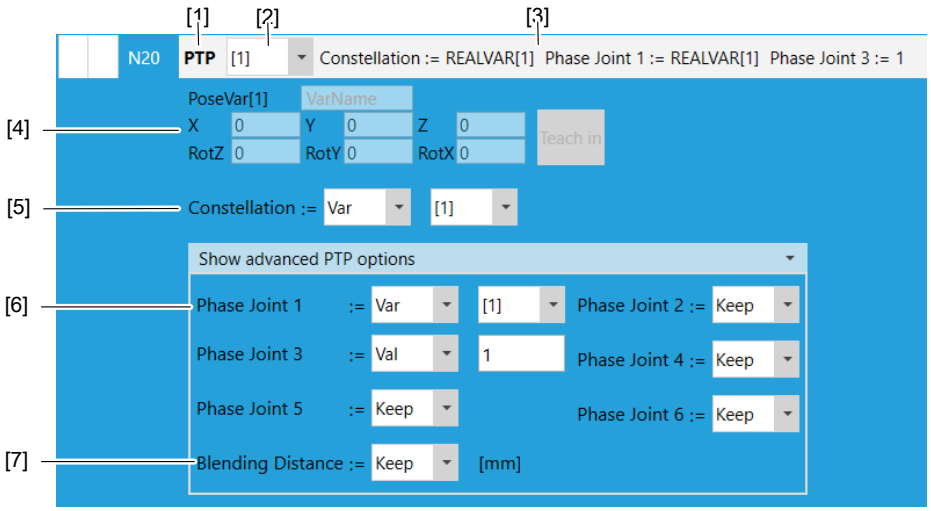
HINWEIS

Bei Programmstart werden "Constellation", "Phasen Joint 1-6" und "BlendingDistance" automatisch auf "0" gesetzt. Siehe auch Kapitel "Herstellen des Grundzustands" (→ 🗎 38).

PTP-Segment mit Posenvariable hinzufügen

PTP-Segment mit Vorgabe der Zielpose über eine SRL-Programmvariable.

"New Block"-Wert: "PTP_VAR".



[1] Kennung für PTP-Segment
[2] Auswahl der Zielposenvariablen
[3] Kompakte Anzeige der selektierten Werte
[4] Anzeige und Auswahl der selektierten Zielposenvariablen.
Mit der TeachIn Funktion wird die aktuelle Pose in der Variablen der Zielpose gespeichert. Die Konstellation und die Gelenkachsphasen werden dabei jedoch nicht mit abgespeichert.
[5] Auswahl der Konstellation über ...
- Zuweisungsart (Var = SRL-Variable, Val = angegebener Wert, Keep = keine Zuweisung soll erfolgen und der bisherige Wert der Eigenschaft bleibt erhalten)
- Zuweisungswert (Var = Auswahlfeld mit dem Index der verfügbaren SRL-Variablen, Val = Eingabefeld für absolute Werte mit dem Typ Gleitkommazahl)

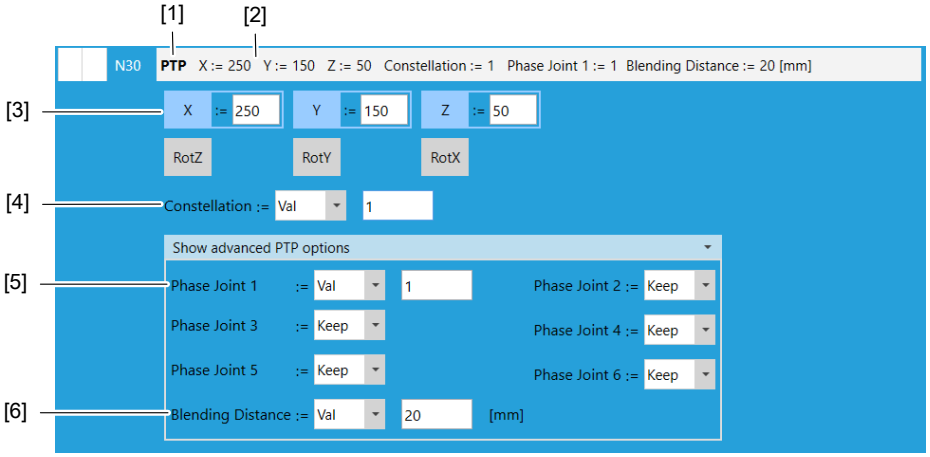
[6] Auswahl der Gelenkachsphasen (Var = SRL-Variable, Val = angegebener Wert, Keep = keine Zuweisung soll erfolgen und der bisherige Wert der Eigenschaft bleibt erhalten)
[7] Festlegung der Überschleifdistanz

26873346/DE – 07/2021

PTP-Segment mit expliziter Koordinatenvorgabe hinzufügen

PTP-Segment mit Vorgabe der Zielpose explizit im SRL-Programm.

"New Block"-Wert: "PTP_EXPLICIT



33225317387

[1] Kennung für PTP-Segment
[2] Kompakte Anzeige der selektierten Werte
[3] Eingabefelder der Koordinaten
[4] Auswahl der Konstellation über ...
- Zuweisungsart (Var = SRL-Variable, Val = angegebener Wert, Keep = keine Zuweisung soll erfolgen und der bisherige Wert der Eigenschaft bleibt erhalten)
- Zuweisungswert (Var = Auswahlfeld mit dem Index der verfügbaren SRL-Variablen, Val = Eingabefeld für absolute Werte mit dem Typ Gleitkommazahl)

[5] Auswahl der Gelenkachsphasen (Var = SRL-Variable, Val = angegebener Wert, Keep = keine Zuweisung soll erfolgen und der bisherige Wert der Eigenschaft bleibt erhalten)
[6] Festlegung der Überschleifdistanz

26873346/DE – 07/2021

10.11 Zuweisungen

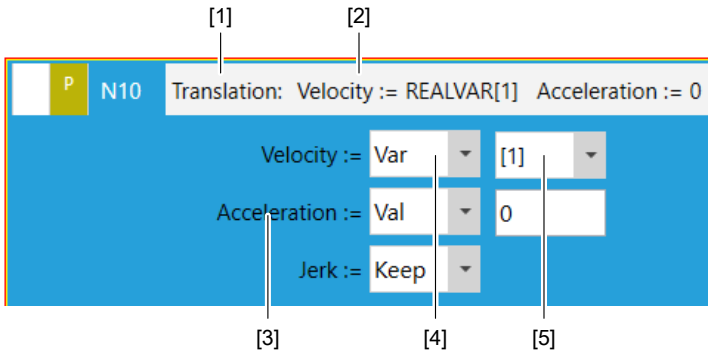
Zuweisungen sind Befehle, die solange wirksam sind, bis der Befehl erneut mit einer anderen Parametrierung aufgerufen wird.

10.11.1 Translatorische Bewegungseigenschaften setzen

Befehl, mit dem die Geschwindigkeit, Beschleunigung, Verzögerung und Ruck von nachfolgenden, translatorischen Bewegungsbefehlen festgelegt wird. Es dürfen nicht alle Zuweisungen auf "Keep" stehen. Mindestens eine Zuweisung ("Value" oder "Variable") muss erfolgen, ansonsten ist das Programm nicht gültig.

Die eingestellten Werte sind so lange gültig bis diese mit einem erneuten Aufruf von SET_MOTION_PARA erneut geändert werden oder alle Bewegungsparameter durch einen kompletten Bewegungsparametersatz mit dem Befehl SET_MOTIONSET überschrieben werden. Die Werte innerhalb der definierten Bewegungsparametersätze werden durch diesen Befehl nicht verändert.

"New Block"-Wert: "SET_MOTION_PARA"



31658896395

[1] Kennung für eine Zuweisung der translatorischen Bewegungsparameter
[2] Kompakte Anzeige der Zuweisungsauswahl
[3] Zu schreibende Eigenschaft der translatorischen Bewegung
[4] Auswahl der Zuweisungsart (Var = SRL-Variable, Val = Vorgegebener Wert, Keep = keine Zuweisung soll erfolgen der bisherige Wert der Eigenschaft bleibt erhalten)
[5] Wertigkeit der Zuweisung (Var = Auswahlfeld mit dem Index der verfügbaren SRL-Variablen, Val = Eingabefeld für absolute Werte mit dem Typ Gleitkommazahl

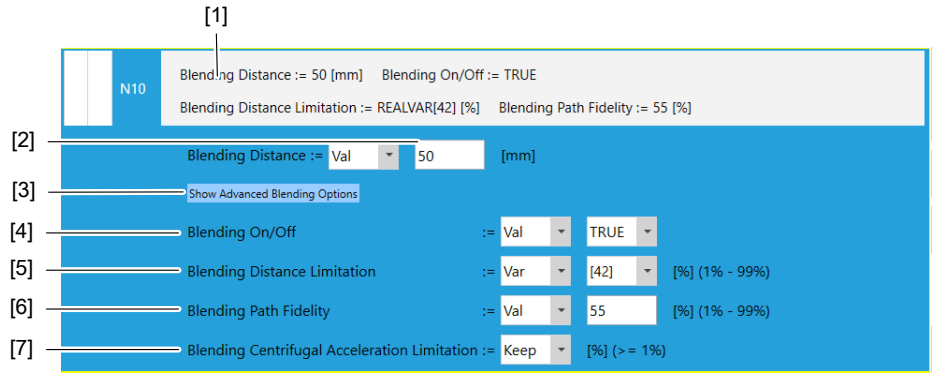
26873346/DE – 07/2021

10.11.2 Parameter für das Überschleifen setzen

HINWEIS

Bei Programmstart wird das Überschleifen ("Blending On/Off") automatisch eingeschaltet und "BlendingDistance" auf "0" gesetzt. Siehe auch "Herstellen des Grundzustands" (→ 🗎 38).

"New Block"-Wert: "SET_BLENDING_PARA"



18014423431719307

[1] Eingestellte Parameter zum Überschleifen
[2] Abstand zum Endpunkt des Segments, ab dem auf das neue Segment übergeschliffen werden soll. Die Überschleifdistanz 0 führt zu einem Genauhalt.
[3] Einblenden spezieller Parameter zum Überschleifen
[4] Ein-/Ausschalten des Überschleifens. Ausgeschaltet kommt es zu einem Genauhalt.
[5] Prozentsatz der Länge des Segments, auf das übergeschliffen werden soll, als Begrenzung der Blending Distance, Standardeinstellung 50 %
[6] Abweichung vom kreisförmigen Überschleifen, Standardeinstellung 1 % maximale Ruckbegrenzung beim Übergang, 99 % nahezu kreisförmig
[7] Begrenzung der Zentrifugalbeschleunigung, prozentual skaliert auf Minimum der translatorischen Beschleunigungen der angrenzenden Bahnsegmente, Standardeinstellung 100 %

Überschleifen zwischen Bewegungssegmenten

Folgendes Beispiel veranschaulicht das Überschleifen von einem Bewegungssegment auf ein anderes Bewegungssegment:

- Bewegungssegment 1: MotionSet Fast, LIN, Zielposition POSEVAR[1]
- Bewegungssegment 2: BlendingDistance Small, MotionSet Slow, LIN, Zielposition POSEVAR[2]

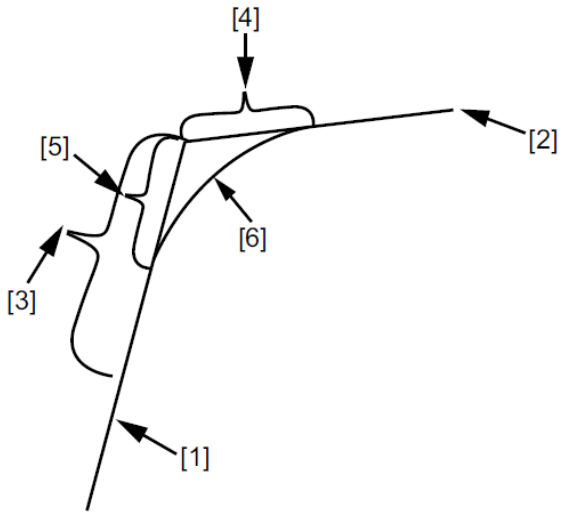
SRL-Programm

```
N10  MotionSet := Fast
N20  LIN POSEVAR[1]
N30  BlendingDistance := Small [mm]
N40  MotionSet := Slow
N50  LIN POSEVAR[2]
N60  END_PROG
```

31458160523

26873346/DE – 07/2021

Schaubild



31458163211

[1] Linearsegment zur Position POSEVAR[1]
[2] Zielposition POSEVAR[2]
[3] Eingestellte $Überschleifdistanz_{Soll}$ (BlendingDistance small) für das Überschleifen auf das Bahnsegment zur Zielposition POSEVAR[2])
[4] Segmentlänge * Begrenzungsprozentsatz
[5] $Überschleifdistanz_{effektiv}$
[6] Resultierender symmetrischer Überschleifbogen

Überschleifen mit Verzweigung

Folgendes Beispiel veranschaulicht das Überschleifen von einem Bewegungssegment auf ein anderes Bewegungssegment abhängig von der BOOL-Variablen *ToTheLeft*:

- Bewegungssegment 1: MotionSet Fast, LIN, Zielposition PoseApproach
- Bewegungssegment 2, abhängig von der BOOL-Variablen *ToTheLeft*:

  BlendingDistance Small, MOTIONSET Slow, LIN, Zielposition PoseLeft

  oder

  BlendingDistance Large, MOTIONSET Medium, LIN, Zielposition PoseRight

Abhängig vom anzufahrenden Punkt *PoseLeft* oder *PoseRight*, soll z. B. zur Kollisionsvermeidung mit der BlendingDistance *Small* oder *Large* auf das zweite Bewegungssegment übergeschliffen werden. Die Entscheidung, wohin verfahren werden soll, ergibt sich jedoch erst während der Bewegung zur Zwischenposition *PoseApproach*. Hierzu wird mittels WAIT *Ready* zunächst gewartet, bevor die BOOL Variable *ToTheLeft* ausgewertet wird.

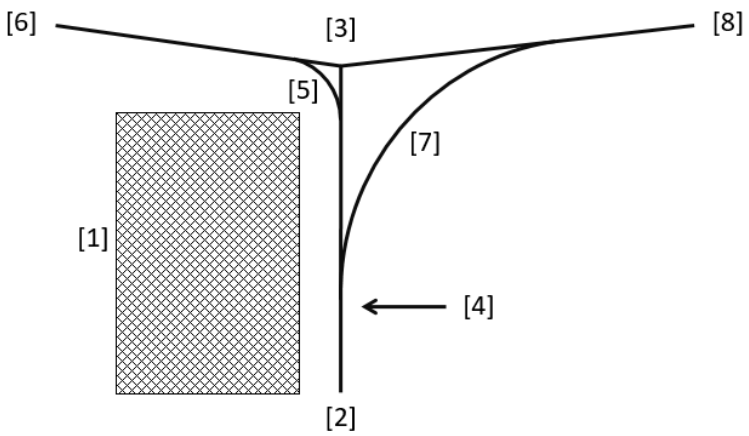
26873346/DE – 07/2021

SRL-Programm

```
N10  MotionSet := Fast
N20  LIN PoseApproach
N30  WAIT Ready
N40  IF ToTheLeft
N50      BlendingDistance := Small [mm]
N60      MotionSet := Slow
N70      LIN PoseLeft
N80  ELSE
N90      BlendingDistance := Large [mm]
N100     MotionSet := Medium
N110     LIN PoseRight
N120 END_IF
N130 END_PROG
```

31458170123

Schaubild



31458366859

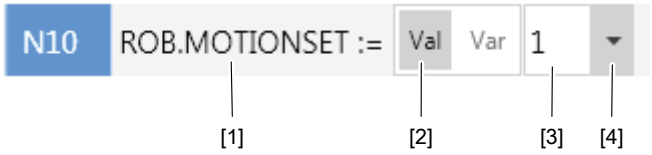
[1] Kollisionsobjekt
[2] Startposition
[3] Zwischenposition *PoseApproach*
[4] An der Stelle wird die BOOL Variable *Ready* TRUE
[5] Kleiner Überschleifbogen: BlendingDistance *Small*
[6] Zielposition *PoseLeft*
[7] Großer Überschleifbogen: BlendingDistance *Large*
[8] Zielposition *PoseRight*

26873346/DE – 07/2021

10.11.3 Bewegungsparametersatz-Zuweisung hinzufügen

Zuweisung, mit der die in einem Bewegungsparametersatz enthaltenen Parameter den folgenden Bewegungsbefehlen zugewiesen werden. Weitere Informationen finden Sie in den Kapiteln "Bewegungsparametersätze" (→ 🗎 28), "Bewegungsparametersätze einstellen" (→ 🗎 108) und "Bewegungsparameter einstellen" (→ 🗎 235).

"New Block"-Wert: "SET_MOTIONSET"



9007224177020939

[1] Kennung für die Bewegungsparametersatz-Zuweisung
[2] Einstellung, ob ein fester Wert (VAL) oder der Wert einer Variable (VAR) zugewiesen werden soll.
[3] Wenn [2] auf VAL steht: Wert, der zugewiesen wird.
Wenn [2] auf VAR steht: Index der Real-Variable, deren Wert zugewiesen wird.
[4] Auswahlliste zum Einstellen von [3]

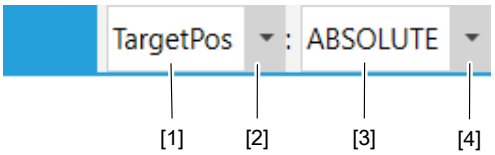
10.11.4 Absolute oder relative Koordinaten einstellen

HINWEIS

Bei Programmstart werden für die Parameter automatisch die Werte des Grundzustands gesetzt. Siehe auch "Herstellen des Grundzustands" (→ 🗎 38).

Zuweisung, mit der absolute oder relative Koordinaten für die nächsten Bewegungsbefehle eingestellt werden.

"New Block"-Wert: "SET_ABS_REL_POS"



33781154699

[1] Auswahl der einzustellenden Koordinaten
- TargetPos - Zielposition des Bewegungsbefehls (Default: ABSOLUTE)
- CircAuxPos - Hilfspunkt für Eingabe eines Kreisbefehls, je nach Kreis-Mode der Kreismittelpunkt oder ein Punkt auf dem Kreisbogen (Default: RELATIVE)

[2] Auswahlliste zum Austauschen von [1]
[3] Wert, der zugewiesen werden soll.
ABSOLUTE - Absolute Position im aktuellen Koordinatensystem
RELATIVE - Zur Anfangsposition des Bewegungssegments relative Position
[4] Auswahlliste zum Austauschen von [3]

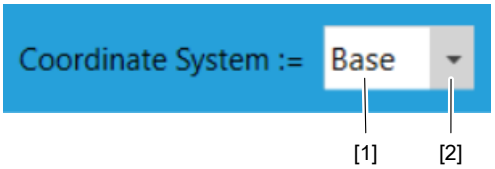
10.11.5 Koordinatensystem einstellen

HINWEIS

Bei Programmstart wird automatisch der Wert des Grundzustands gesetzt (Soordinate System ="Base"). Siehe auch "Herstellen des Grundzustands" (→ 🗎 38).

Zuweisung, mit der das Koordinatensystem eingestellt wird, auf das sich die nächsten Bewegungsbefehle beziehen.

"New Block"-Wert: "SET_COORDSYS"



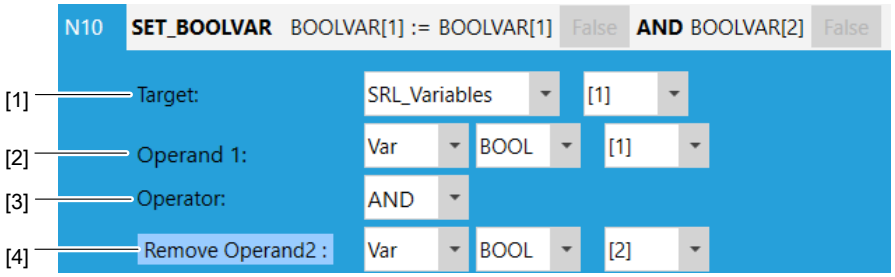
33781180939

[1] Ausgewähltes Koordinatensystem
[2] Auswahlliste zum Austauschen von [1]
- Base, User oder User_ControlledByRobot

10.11.6 Bool-Zuweisung hinzufügen

Zuweisung, in der einer boolschen SRL-Programmvariable ein konstanter Wert, der Wert einer anderen boolschen SRL-Programmvariablen oder das Ergebnis eines boolschen Ausdrucks zugewiesen wird.

"New Block"-Wert: "SET_BOOLVAR"



9007228303254155

[1] Konfiguration der Zielvariablen (Target)
- Speicherort der SRL-Programmvariablen
- Index der SRL-Programmvariablen

[2] Konfiguration von Operand 1:
- Art des Wertes ("Var" für Variablen oder "Val" für konstante Werte)
- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Index der SRL-Programmvariablen

[3] Auswahl des Operators. Verfügbare Operatoren: +, -, *, /
[4] Schaltfläche zum Hinzufügen/Entfernen eines weiteren Operanden
Konfiguration von Operand 2 siehe [1]

26873346/DE – 07/2021

10.11.7 Real-Zuweisung/-Berechnung hinzufügen

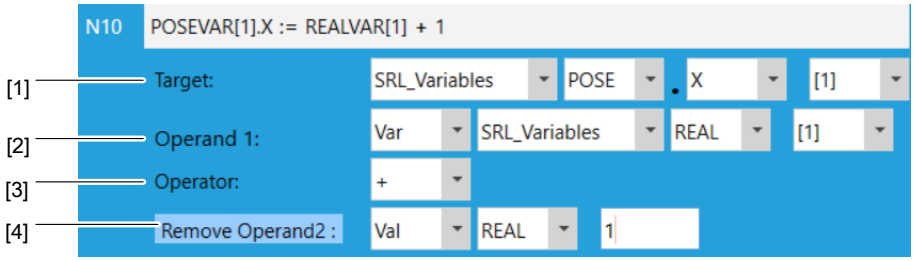
Befehl, um einen Wert des Datentyps REAL zu berechnen und zuzuweisen. Das Ergebnis der Berechnung kann entweder einer REAL-Variablen oder einer Koordinate einer POSE-Variablen zugewiesen werden. Für die beiden Operanden der Berechnung ist auswählbar, ob es sich bei diesen um einen Wert, eine REAL-Variablen oder die Koordinate einer POSE-Variablen handelt. Beispiele:

POSE[1].Z := REAL[1] + 10

REAL[1] := REAL[2] + REAL[3]

POSE[1].Z := POSE[2].Z + REAL[2]

“New Block”-Wert: "CALC_REALVAR"



9007228307417739

[1] Konfiguration der Zielvariablen (Target)

- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Koordinate der POSE-Variablen
- Index der SRL-Programmvariablen

[2] Konfiguration von Operand 1:

- Art des Wertes ("Var" für Variablen oder "Val" für konstante Werte)
- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Index der SRL-Programmvariablen

[3] Auswahl des Operators. Verfügbare Operatoren: +, -, *, /

[4] Schaltfläche zum Hinzufügen/Entfernen eines weiteren Operanden
Konfiguration von Operand 2 siehe [1]

26873346/DE – 07/2021

10.12 Kontrollstrukturen

10.12.1 Bedingte Anweisung

Wenn die Bedingung erfüllt ist, werden die Anweisungen des IF-Teils ausgeführt. Wenn die Bedingung nicht erfüllt ist, werden die Anweisungen des ELSE-Teils ausgeführt. Die Verschachtelungstiefe bedingter Anweisungen (IF-Kontrukt innerhalb eines anderen IF-Konstrukts) ist nicht begrenzt.

IF Bedingung

Anweisungen des IF-Teils

ELSE

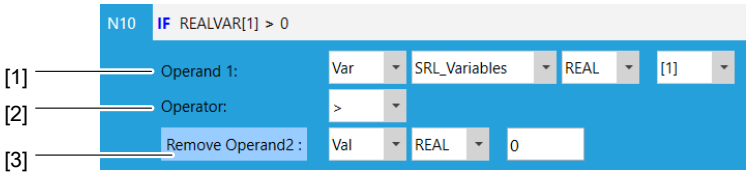
Anweisungen des ELSE-Teils

END_IF

IF-Bedingung hinzufügen

"New Block"-Wert: "IF"

Verschieben Sie den Anfang und das Ende der IF-Bedingung an die gewünschten Programmzeilen.



27021622687479947

[1] Konfiguration von Operand 1:
- Art des Wertes ("Var" für Variablen oder "Val" für konstante Werte)
- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Index der SRL-Programmvariablen

[2] Auswahl des Operators. Verfügbare Operatoren: +, -, *, /

[3] Schaltfläche zum Hinzufügen/Entfernen eines weiteren Operanden
Konfiguration von Operand 2 siehe [1]

ELSE-Bedingung hinzufügen

✓ Im Programm ist eine IF-Bedingung vorhanden.

1. Markieren Sie die Zeile zwischen IF und END_IF, unter die Sie die ELSE-Teil einfügen möchten.
2. Wählen Sie aus der Auswahlliste "New Block" den Wert "ELSE".
3. Klicken Sie auf die Schaltfläche [Add].
4. Parametrieren Sie die Anweisungen des ELSE-Teils.

10.12.2 Schleife

Führt solange die Anweisungen aus, bis die Bedingung nicht mehr erfüllt ist.

WHILE Bedingung

Anweisungen

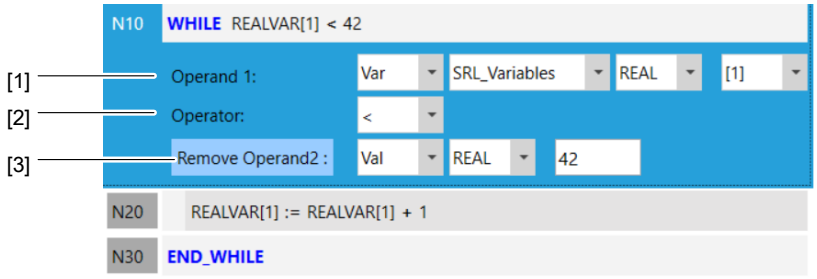
26873346/DE – 07/2021

END_WHILE

WHILE-Schleife hinzufügen

"New Block"-Wert: "WHILE".

Verschieben Sie den Anfang und das Ende der While-Schleife an die gewünschten Programmzeilen.



27021622687606795

[1] Konfiguration von Operand 1:
- Art des Wertes ("Var" für Variablen oder "Val" für konstante Werte)
- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Index der SRL-Programmvariablen

[2] Auswahl des Operators. Verfügbare Operatoren: +, -, *, /
[3] Schaltfläche zum Hinzufügen/Entfernen eines weiteren Operanden
Konfiguration von Operand 2 siehe [1]

10.12.3 Wartebefehl

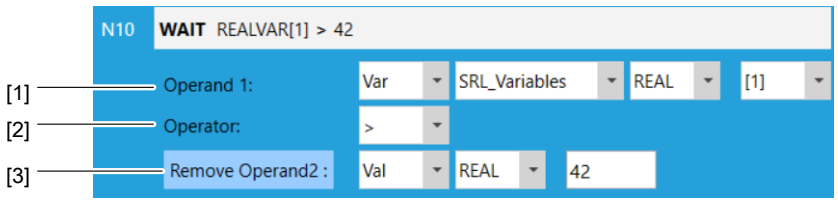
Es wird solange bei dem WAIT-Befehl verharrt, bis die Bedingung erfüllt ist.

WAIT Bedingung

Die Bedingung kann eine Bool-Variable, eine Zeitdauer oder ein Statussignal sein, z. B. "Bewegungsende" (MOTION_DONE).

Warten auf Bool-Variable oder erfüllte Bedingung einfügen

"New Block"-Wert: "WAIT_BOOLVAR"



27021622687627403

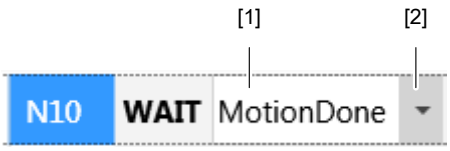
[1] Konfiguration von Operand 1:
- Art des Wertes ("Var" für Variablen oder "Val" für konstante Werte)
- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Index der SRL-Programmvariablen

[2] Auswahl des Operators. Verfügbare Operatoren: +, -, *, /
[3] Schaltfläche zum Hinzufügen/Entfernen eines weiteren Operanden
Konfiguration von Operand 2 siehe [1]

26873346/DE – 07/2021

Warten auf eine Bewegung einfügen

"New Block"-Wert: "WAIT_MOTION"



18014423432894987

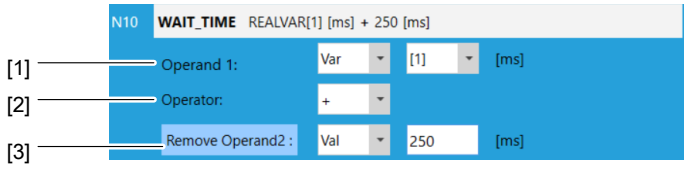
[1] Argument des Wartebefehls: Bewegungsende
- Motion_Done: Roboter hat alle Fahrbefehle ausgeführt und ist im Stillstand oder Bewegung ist in letztem Bewegungssegment.
- MotionInLastSeg: Roboter führt gerade den letzten Fahrbefehl aus.

[2] Auswahlliste zum Austauschen von [1]

Warten auf Ablauf einer Totzeit einfügen

Wartezeit beginnt, nachdem der Bewegungsbefehl vor WAIT_TIME beendet wurde. Auch wenn die eingestellte Wartezeit 0 ms ist, wird die vorherige Bewegung beendet, d. h. es kommt zu einem Rastpunkt.

"New Block"-Wert: WAIT_TIME



27021622687700747

[1] Konfiguration von Operand 1:
- Art des Wertes ("Var" für Variablen oder "Val" für konstante Werte)
- Speicherort der SRL-Programmvariablen
- Datentyp der SRL-Programmvariablen ("REAL", "BOOL" oder "POSE")
- Index der SRL-Programmvariablen

[2] Auswahl des Operators. Verfügbare Operatoren: +, -, *, /

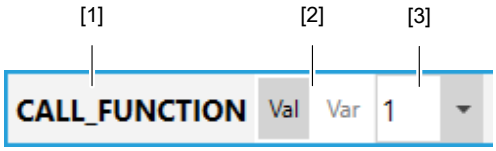
[3] Schaltfläche zum Hinzufügen/Entfernen eines weiteren Operanden
Konfiguration von Operand 2 siehe [1]

26873346/DE – 07/2021

10.12.4 IEC-Funktionsaufruf

Befehl zum Aufrufen einer IEC-Funktion, die vom Anwender im IEC-Editor programmiert werden kann. Dieser Befehl ruft die Funktion jeden Zyklus des hochpriorisierten zyklischen Tasks auf (TaskHighPrio), bis die IEC-Funktion den Rückgabewert "TRUE" zurückliefert. Erst anschließend wird der nächste Befehl ausgeführt.

Die CallFunction kann beispielsweise für die Erzeugung eines konsistenten Prozessabbilds an einer klar definierten Stelle im Programmablauf verwendet werden. Es können z. B. Positionen und MotionSets (mittels Methode im UserInterface) beschrieben werden, die im folgenden SRL-Code zu verwenden sind.



18014423433087499

[1] Kennung für den IEC-Funktionsaufruf
[2] Auswahl, ob die Nummer der aufzurufenden Funktion direkt eingegeben werden soll oder aus einer Variablen gelesen werden soll.
[3] Übergabeparameter für den Funktionsaufruf, damit mittels einer Fallunterscheidung in der IEC-Funktion verschiedene Funktionen aufgerufen werden können.

Die Erstellung der *CallFunction* im IEC-Programm und die Verknüpfung mit dem Roboter ist in Kapitel "IEC-Funktionsaufruf für das SRL-Programm" (→ 🗎 203) beschrieben.

10.12.5 Programmende

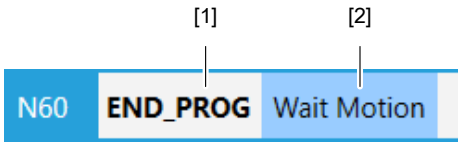
Befehl zum Beenden des Programms

Nachdem ein Programm mit diesem Befehl beendet wurde, kann das nächste Programm gestartet werden. Der Programmende-Befehle kann auch eingesetzt werden um das Programm vorzeitig zu beenden. Es muss mindestens nach dem letzten Befehl stehen, darf aber auch mehrfach verwendet werden.

Programmende hinzufügen

"New Block"-Wert: "END_PROG"

Es kann ausgewählt werden, ob der Programmende-Befehl warten soll bis die Roboterbewegung beendet ist, bevor das Programm beendet wird oder ob das Programm sofort beendet werden soll, wenn die Programmabarbeitung (Programmzeiger) am Programmende-Befehl angekommen ist. Ohne das Warten auf das Beenden der Bewegung ist es z. B. möglich, kontinuierlich Bewegungssegmente einzuspeisen.



31659655563

[1] Kennung für den Programmende-Befehl
[2] Schaltfläche um das Warten auf die Beendigung der Bewegung zu aktivieren / deaktivieren

10.12.6 Unterprogrammaufruf

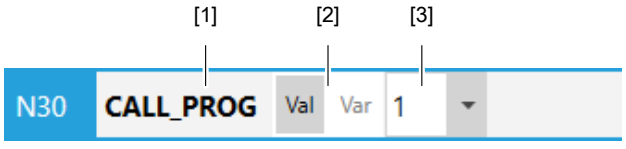
HINWEIS

Beim Unterprogrammaufruf erfolgt keine Herstellung des Grundzustands. Siehe auch "Herstellen des Grundzustands" (→ 🗎 38).

Durch diesen Befehl wird ein anderes SRL-Programm aufgerufen und ausgeführt. Nachdem das angerufene Programm beendet ist, wird das ursprüngliche Programm in der auf den CALL_PROG-Block folgenden Zeile fortgesetzt. Die maximale Verschachtelungstiefe für den Fall, dass durch ein anderes Programm aufgerufene Programme wiederrum Programmaufrufe enthalten, beträgt 10.

Aufruf eines anderen SRL-Programms hinzufügen

„New Block“-Wert: „CALL_PROG“.



31659767435

[1] Kennung für den Aufruf eines anderen Programms
[2] Auswahl, ob die Nummer des aufzurufenden Programms direkt eingegeben werden soll oder aus einer Variablen gelesen werden soll
[3] Eingabefeld für die Nummer des aufzurufenden Programms

26873346/DE – 07/2021

10.12.7 Bahnereignis

HINWEIS

Bei Programmstart werden für die Parameter automatisch die Werte des Grundzustands gesetzt (Reference = "BeginOfSegment", Distance = "0", Time = "0"). Siehe auch "Herstellen des Grundzustands" (→ 🗎 38).

Mit dem Befehl REG PATH_EVENT kann eine Anweisung registriert werden, die bei einem bestimmten Ereignis abhängig vom Bahnfortschritt ausgeführt werden soll. Der Anwender kann das Bahnereignis weg- und/oder zeitbasiert definieren.

Weiterführende Information finden Sie im Kapitel "Bahnereignisse" (→ 🗎 29) und "Funktionsbeschreibung" (→ 🗎 80).

Beim Eintreten des Bahnereignis kann eine der folgenden Anweisungen ausgeführt werden: SET_BOOLVAR, CALL_FUNCTION, REG_TP_POSITIONING, REG_TP_MEASURE, DEREG_TP. Die auszuführende Anweisung muss in der Zeile unter dem Befehl REG PATH_EVENT stehen und wird durch eine Einrückung markiert. Es kann genau eine Anweisung programmiert werden.

Die Parameter im Befehl REG PATH_EVENT beziehen sich auf den nächsten Bewegungsbefehl im SRL-Programm. Das SRL-Programm wird im RobotMonitor nur dann als fehlerfrei angezeigt, wenn auf REG PATH_EVENT ein Bewegungsbefehl folgt. Zwischen REG PATH_EVENT und dem Bewegungsbefehl können jedoch auch andere Anweisungen programmiert sein, z. B. weitere REG PATH_EVENT, die sich dann alle auf den gleichen, folgenden Bewegungsbefehl beziehen. So ist es möglich, mittels mehrerer REG PATH_EVENT Befehle mit gleicher Parametrierung mehrere Anweisungen beim gleichen Bahnereignis auszuführen.

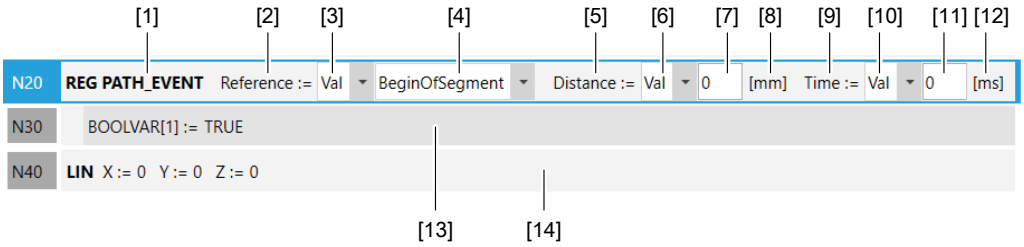
In der speziellen Situation, dass alle im gleichen SRL-Programm auf REG PATH_EVENT folgenden Bewegungsbefehle z. B. wegen einer IF-Verzweigung nicht ausgeführt werden und das SRL-Programm als Unterprogramm (CALL_PROG) aufgerufen wurde, bezieht sich REG PATH_EVENT auf den ersten Bewegungsbefehl im aufrufenden SRL-Programm. Im Gegensatz hierzu werden bei jedem Programm-Stopp und jedem Programm-Start (Grundzustand) alle registrierten und noch nicht ausgelösten Bahnereignisse abgelöscht.

Der IEC-Code in der Anweisung CALL_FUNCTION muss bei Auslösung durch ein Bahnereignis innerhalb eines Zyklus Done = TRUE zurückgeben. Hierbei kann z. B. ein nebenläufiger Prozess in der IEC angestoßen werden.

26873346/DE – 07/2021

Bahnereignis hinzufügen

"New Block"-Wert: "REG_PATH_EVENT"



31659788811

[1] Kennung für Registrierung eines Bahnereignisses
[2] Kennung für den Referenzpunkt
[3] Auswahl, ob der Referenzpunkt direkt eingegeben, aus einer Variablen gelesen oder von einem vorhergehenden Befehl beibehalten werden soll
[4] Eingabefeld für den Referenzpunkt
[5] Kennung des Distanzparameters
[6] Auswahl, ob der Wert für die Distanz direkt eingegeben, aus einer Variablen gelesen oder von einem vorhergehenden Befehl beibehalten werden soll
[7] Eingabefeld für den Distanzparameter
[8] Einheit des Distanzparameters
[9] Kennung des Zeitparameters
[10] Auswahl ob der Wert für den Zeitparameter direkt eingegeben werden soll, aus einer Variablen gelesen werden soll oder von einem vorhergehenden Befehl beibehalten werden soll
[11] Eingabefeld für den Zeitparameter
[12] Einheit des Zeitparameters
[13] Anweisung, die bei Eintritt des in der vorangegangenen Zeile registrieten Ereignisses ausgeführt werden soll, hier: Setzen einer BOOL-Variablen auf TRUE. Diese Zeile wird im normalen Programmablauf übersprungen.
[14] Bahnsegment, auf das sich REG PATH_EVENT bezieht

Ein Bahnereignis, das registriert wurde (Programminterpreter hat den darauffolgenden Bewegungsbefehl übernommen), ist mit einem drehenden Kreis gekennzeichnet. Sobald das Bahnereignis auslöst, wird es mit einem grünen Haken markiert. Bei Programm-Start und -Stopp werden die Kennzeichnungen abgelöscht. Wird ein Bahnereignis z. B. in einer Schleife oder in Unterprogrammen erneut registriert, wechselt die Anzeige vom Haken wieder zum drehenden Kreis. Wenn ein Bahnereignis bereits getriggert wurde und der Roboter aufgrund einer Vorlaufzeit noch nicht losfahren darf, werden der Haken und der drehende Kreis gleichzeitig angezeigt; der drehende Kreis verschwindet, sobald die Roboterbewegung beginnt.

Die Punkte, an denen die Bahnereignisse auslösen, werden in der 3D-Simulation bei eingeschaltetem Stift angezeigt. Aufgrund der niederprioren Kommunikation zwischen MOVI-C® CONTROLLER und PC kann es zu kleinen Ungenauigkeiten in der 3D-Anzeige kommen.

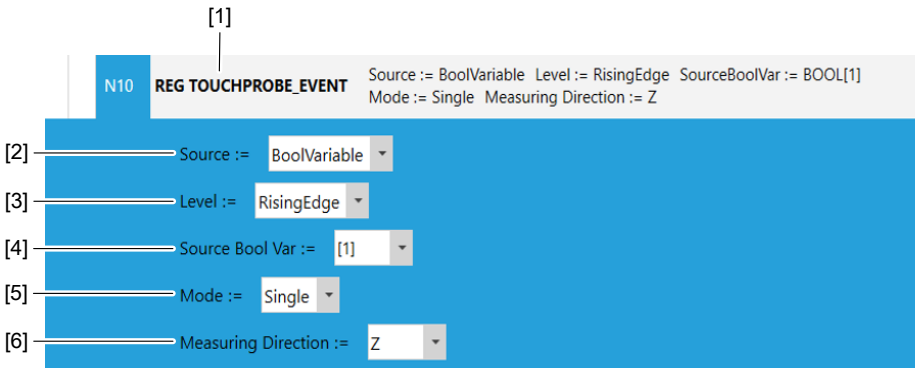
26873346/DE – 07/2021

10.12.8 Touchprobe

Touchprobe-Event registrieren

Über die Anweisung REG_TOUCHPROBE_EVENT wird die Touchprobe-Funktion aktiviert und parametriert.

"New Block"-Wert: "REG_TP_EVENT".



31690470539

[1] Kennung der Anweisung

[2] Verwendete Touchprobe-Triggerquelle: Auswahlmöglichkeit zwischen "BoolVariable" (BOOL-Touchprobe) und "InverterTouchprobe" (Umrichter-Touchprobe)

[3] Nur für BOOL-Touchprobe: Verwendete Flanke zum Triggern des Touchprobe-Events. Auswahlmöglichkeit zwischen "RisingEdge" (steigende Flanke), "FallingEdge" (fallende Flanke) und "RisingFallingEdge" (jeder Flankenwechsel).
Bei Umrichter-Touchprobe wird die Flanke über die Konfiguration der Einzelachsen in MOVISUITE® eingestellt.

[4] Nur für BOOL-Touchprobe: Verwendete SRL-BOOL-Variable zum Auslösen des Touchprobe-Events

[5] Ausführungsmodus des Touchprobes: Auswahlmöglichkeit "Single" (Touchprobe wird nach Triggern des Events deaktiviert) und "Multiple" (Touchprobe bleibt nach Triggern des Events aktiviert. Bei jedem Triggern wird die dem Event zugeordnete Anweisung ausgeführt)

[6] Richtung, in welcher der Touchprobe-Sensor misst bzw. die BOOL-Variable geschaltet wird, z. B. die Z-Richtung (Höhe / Hub) beim Palettieren.
Bei nachfolgender "POSITIONING"-Anweisung: Wirkrichtung der Restweglänge

Ein Touchprobe-Ereignis, das registriert wurde und dadurch aktiv ist, ist mit einem drehenden Kreis gekennzeichnet. Sobald das Touchprobe-Ereignis getriggert wurde, wird es mit einem grünen Haken markiert. Beim TouchProbe-Mode "Multiple" wird nach dem Triggern sowohl ein grüner Haken als auch ein drehender Kreis angezeigt, da das Ereignis getriggert wurde, aber weiterhin aktiv ist. Beim Programm-Start und -Stopp werden die Kennzeichnungen abgelöscht. Wurde ein Ereignis deregistriert (entweder durch einen Deregister-Befehl oder durch Hauptprogrammende) bevor es ausgelöst wurde, wird dies durch ein rotes Kreuz kenntlich gemacht.

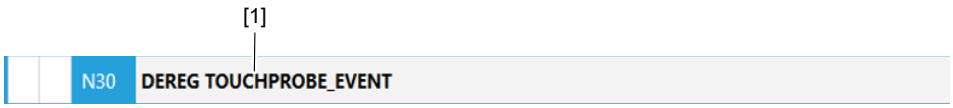
Die Punkte, an denen die Touchprobe-Ereignisse auslösen, werden in der 3D-Simulation bei eingeschaltetem Stift als pinke Quader angezeigt. Aufgrund der niederprioren Kommunikation zwischen MOVI-C® CONTROLLER und PC kann es zu kleinen Ungenauigkeiten in der 3D-Anzeige kommen.

Touchprobe-Event deregistrieren

Über die Anweisung DEREG_TOUCHPROBE_EVENT wird eine aktivierte Touchprobe-Funktion wieder deaktiviert. Dies kann z. B. genutzt werden, wenn nur in einem bestimmten Bereich der Bewegungsbahn Messungen vorgenommen werden sollen.

26873346/DE – 07/2021

"New Block"-Wert: "DEREG_TP_EVENT".



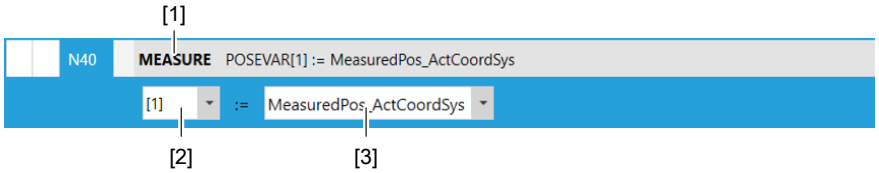
31690744843

[1] Kennung der Anweisung

Touchprobe-Messung durchführen

Über die Anweisung "REG_TP_MEASURE" wird nach dem Auslösen eines Touchprobe-Events die gemessene Position in eine SRL-Pose-Variable geschrieben. Eine "MEASURE"-Anweisung darf nur unter einer "REG_TOUCHPROBE_EVENT"-Anweisung verwendet werden.

"New Block"-Wert: "REG_TP_MEASURE".



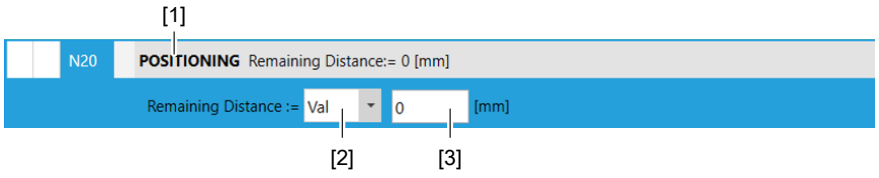
31690791307

[1] Kennung der Anweisung
[2] Auswahl der SRL-Pose-Variable, in welche die gemessene Position gespeichert werden soll
[3] Auswahl des Koordinatensystems, in welchem die gemessene Position abgespeichert werden soll. Auswahlmöglichkeiten: "MeasuredPos_ActCoordSys" und "MeasuredPos_BaseCoordSys"

Restwegpositionierung durchführen

Über die Anweisung "REG_TP_POSITIONING" wird nach dem Auslösen eines Touchprobe-Events eine Restwegpositionierung eingeleitet. Diese Positionierung wird entlang der Bahnsegmente um die Länge *Remaining Distance* in Richtung der *MeasuringDirection* durchgeführt. Gegebenenfalls wird das letzte Bewegungssegment verlängert. Eine "POSITIONING"-Anweisung darf nur unter einer "REG_TOUCHPROBE_EVENT"-Anweisung verwendet werden. Nach einer "POSITONING"-Anweisung ist zwingend eine "CONTINUE AFTER POSITIONING EVENT"-Anweisung erforderlich.

"New Block"-Wert: "REG_TP_POSITIONING".



31690803723

[1] Kennung der Anweisung
[2] Einstellmöglichkeit der *Remaining Distance*:
Auswahlmöglichkeiten: "Var", "Val", "Keep"
[3] Angabe der *Remaining Distance* in [mm]

26873346/DE – 07/2021

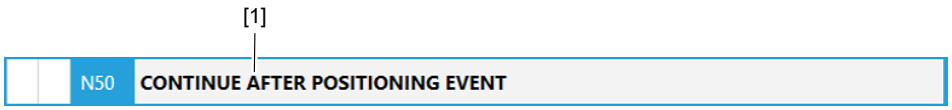
Nächste Programmposition festlegen

HINWEIS

Bei Programmstart werden für die Parameter der Touchprobe-Ereignisse und der Restwegpositionierung automatisch die Werte des Grundzustands gesetzt (Source = "Inverter", Source Bool Var = "[1]", Level = "RisingEdge", Mode = "Single", Measuring Direction = "Z", Remaining Distance = "0", Touchprobe-Zähler = "0"). Siehe auch "Herstellen des Grundzustands" (→ 🗎 38).

Über die Anweisung "CONTINUE AFTER POSITIONING EVENT" wird im Programmablauf die Stelle markiert, bis wohin Bahnsegmente zum Restwegfahrbereich gehören. Wenn die Restwegpositionierung gestartet wird, wird im Programmablauf zu der Zeile nach der "CONTINUE AFTER POSITIONING EVENT"-Anweisung gesprungen.

"New Block"-Wert: "TP_CONTINUE_AFTER_POS".



31690812171

[1] Kennung der Anweisung

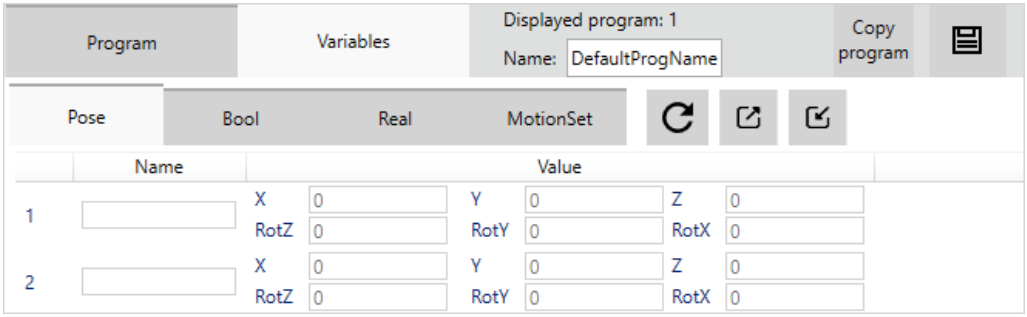
Erreicht die Bewegung das Bewegungssegment nach der Anweisung "CONTINUE AFTER POSITIONING EVENT" (gekennzeichnet durch den Bewegungszeiger "M") und es trat noch kein Touchprobe-Ereignis auf, wird die Touchprobe-Funktion deaktiviert.

26873346/DE – 07/2021

10.13 Variablen

Im Register "Variables" wird eine Liste an Variablen zur Verfügung gestellt. Diese Variablen können im SRL-Programm verwendet werden. Die Variablen können vom IEC-Programm und vom SRL-Programm aus geschrieben und gelesen werden.

Register "Variables"



33244052875

Variablen SRL-Programm	Beschreibung
Bool	Variablen mit Namen und Wert der Boolvariable
Real	Variablen mit Namen und Wert der Gleitkommavariable
Pose	Variablen mit Namen und Koordinaten der Posen
MotionSet	Variablen mit den Bewegungsparametersätzen Siehe auch Kapitel "Bewegungsparametersätze" (→ 🗎 28), "Bewegungsparametersätze einstellen" (→ 🗎 108) und "Bewegungsparameter einstellen" (→ 🗎 235). Es gelten dabei die festgelegten "Standardeinheiten" (→ 🗎 28).

10.13.1 SRL-Programmvariablen editieren

✓ Der Robotermonitor ist gestartet und verbunden.

1. Wechseln Sie zum Register "Variables".
2. Wechseln Sie zum Register der gewünschten SRL-Programmvariablen.
   ⇨ Es wird die Liste der gewünschten SRL-Programmvariablen angezeigt. Eine Zeile entspricht einer Variablen.
3. Tragen Sie in der Spalte "Name" für die Variable einen individuellen Namen ein. Dieser wird im SRL-Programm angezeigt. Ist das Feld leer, wird nur die Nummer im SRL-Programm angezeigt.
4. Tragen Sie in der Spalte "Value" für die Variable die gewünschten Werte ein.
5. Damit die Werte wirksam werden (solange wird die Zeile gelb markiert), drücken Sie während der Cursor im entsprechenden Eingabefeld steht die Eingabetaste oder markieren Sie die Zeile und klicken Sie auf die Schaltfläche [Send selected variable (POSE/BOOL/REAL)]. Damit ist sie im flüchtigen Speicher der Steuerung gespeichert.
6. Um die Variablen und das Programm dauerhaft auf der Speicherkarte zu speichern, klicken Sie auf das Speichersymbol.

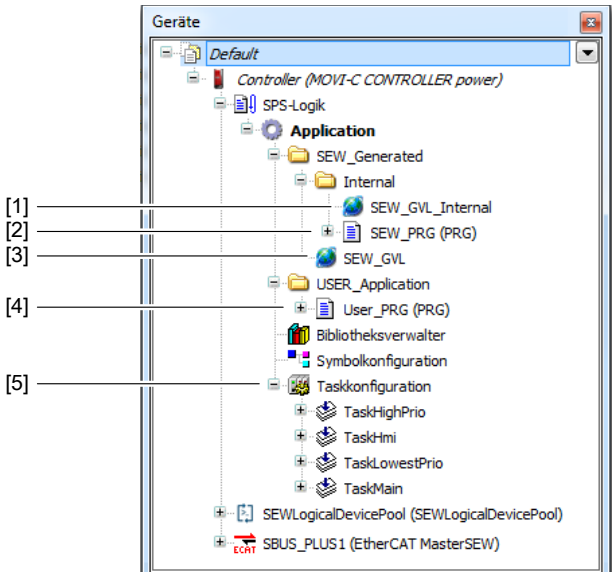
26873346/DE – 07/2021

11 IEC-Programmierung

In diesem Kapitel sind die IEC-Signale der Anwenderschnittstelle für die Programmierung beschrieben. Die Ansteuer-Reihenfolge der Signale ist im Kapitel "Ansteuerung durch die Prozess-Steuerung" (→ 🗎 130) beschrieben.

11.1 Aufbau des IEC-Projekts

Das IEC-Projekt weist folgende Grundstruktur auf:



Nr.	Name	Beschreibung
[1]	SEW_GVL_Internal	Die globale Variablenliste SEW_GVL_Internal beinhaltet die zum verwendeten Softwaremodul passenden Instanzen. Auf diese Variablen darf nicht aus dem Anwenderprogramm geschrieben werden. Des Weiteren enthält die Struktur eine Instanz als Kommunikationspuffer zum Steuern und Beobachten des Softwaremoduls mit dem MOVISUITE® Monitor.
[2]	SEW_PRG	Programm, in dem alle wichtigen Instanzaufrufe zusammengefasst sind. Die automatische Codegenerierung erzeugt dieses Programm bei jeder Generierung des IEC-Projekts entsprechend der Konfiguration in der MOVISUITE® neu und überschreibt die Vorgängerversion. Daher sollten in diesem Programm keine Änderungen vorgenommen werden.
[3]	SEW_GVL	Die globale Variablenliste SEW_GVL stellt die Schnittstelle für den Zugriff auf die Funktionalitäten des Softwaremoduls dar.
[4]	User_PRG	Programm, das von der automatischen Codegenerierung einmalig initial erzeugt wird. Da es nicht bei jeder weiteren Generierung überschrieben wird, ist dies die geeignete Stelle zum Einbinden von Anwenderprogrammen. Das Programm ist in fünf Aktionen gegliedert, die sich darin unterscheiden zu welchem Zeitpunkt des Programmablaufs sie aufgerufen werden.

26873346/DE – 07/2021

Nr.	Name	Beschreibung
[5]	Task-Konfiguration	Auflistung der im Projekt angelegten Tasks. Die automatische Codegenerierung fügt initial Tasks hinzu, die sich in ihrer Priorisierung unterscheiden. Der Anwender kann weitere Programme zu den bestehenden Tasks hinzufügen oder neue Tasks anlegen. Es liegt in der Verantwortung des Anwenders, die Auslastung der Tasks dabei so zu gestalten, dass diese in der geforderten Zykluszeit verarbeitet werden können. Das Überfahren insbesondere der zyklischen Tasks führt dazu, dass Sollwerte für interpolierende Achsen nicht rechtzeitig bereitgestellt und diese somit nicht mehr ordnungsgemäß betrieben werden können.

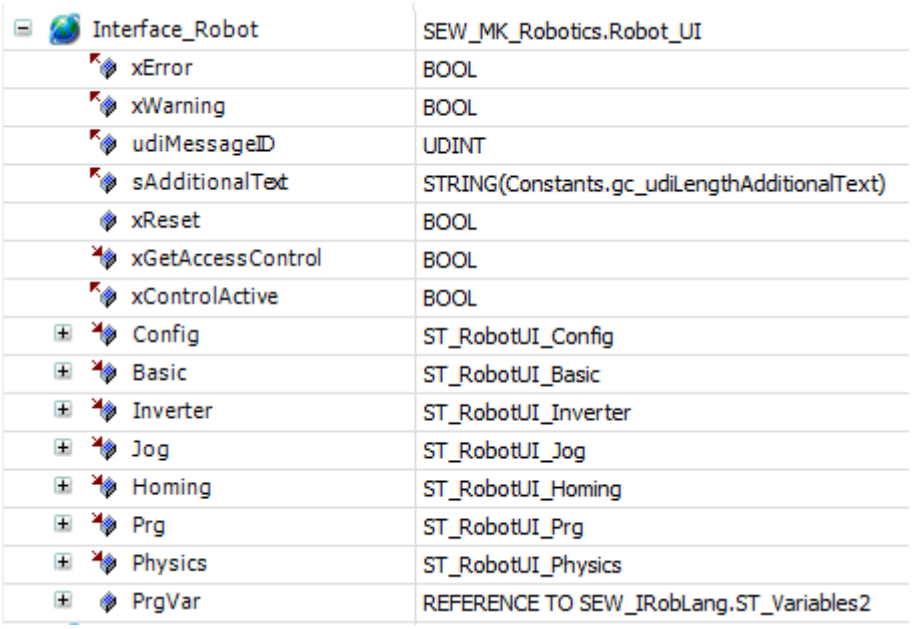
11.2 IEC-Projekt öffnen

- Wenn bereits ein IEC-Projekt generiert wurde, wählen Sie in MOVISUITE® im Kontextmenü des MOVI-C® CONTROLLER den Menübefehl [Tools] > [IEC-Editor].
- Wenn noch kein IEC-Projekt generiert wurde, befolgen Sie die im Kapitel "IEC-Projekt generieren" (→ 🗎 105) beschriebenen Schritte.

11.3 Anwenderschnittstelle

Die Anwenderschnittstelle ist im IEC-Programm durch eine Instanz in der globalen Variablenliste *SEW_GVL* realisiert.

Folgende Grafik zeigt die Schnittstelle im IEC-Editor:



27021622397307403

Dieser Variablenstruktur folgend, sind die einzelnen Variablen in den nächsten Kapiteln genauer beschrieben.

26873346/DE – 07/2021

11.4 Diagnose

Variablen zum Melden und Beschreiben von Fehlern und Warnungen.

Variablenname	Beschreibung
xError	Datentyp - BOOL
	• TRUE - Ein Fehler liegt vor • FALSE - Kein Fehler liegt vor
xWarning	Datentyp - BOOL
	• TRUE - Eine Warnung liegt vor • FALSE - Keine Warnung liegt vor
udiMessageID	Datentyp - UDINT
	Identifikationsnummer der Meldung
sAdditionalText	Datentyp - STRING
	Zusatztext der Meldung
xReset	Datentyp - BOOL
	• TRUE - Meldungen zurücksetzen • FALSE - Meldungen nicht zurücksetzen

11.5 Zugriffsverwaltung

Variablen zum Verwalten der Zugriffsberechtigung.

Variablenname	Beschreibung
xGetAccessControl	Datentyp - BOOL
	• TRUE - Zugriff anfordern • FALSE - Zugriff zurückgeben
xControlActive	Datentyp - BOOL
	• TRUE - Zugriff wurde gewährt • FALSE - Zugriff wurde nicht gewährt

Zugriff durch Anwenderschnittstelle (UserInterface):

Eine Instanz fordert den Zugriff durch das Setzen von *xGetAccessControl* auf "TRUE" an. Wenn *xControlActive* den Wert "TRUE" zurückmeldet, wurde der Zugriff gewährt.

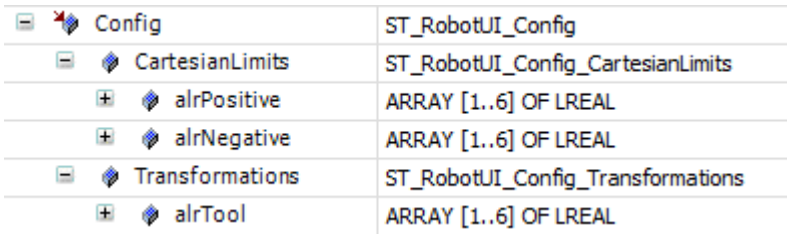
Zugriff durch andere Funktionsbausteine im Anwenderprogramm:

Wenn parallel zur Anwenderschnittstelle (UserInterface) ein anderer Funktionsbaustein im Steuer-Modus auf die Geräteschnittstelle zugreifen möchte, entscheidet die Geräteschnittstelle anhand der Priorität des Funktionsbausteins, wer das Schreibrecht erhält. Die Anwenderschnittstelle (UserInterface) hat dabei die höchste Priorität, d. h. wenn die Anwenderschnittstelle (UserInterface) steuernden Zugriff hat, wird allen weiteren Funktionsbausteinen über *xControlActive* "FALSE" zurückgemeldet.

26873346/DE – 07/2021

11.6 Konfiguration (Config)

Schnittstelle im IEC-Editor



33781265035

Die Variablen werden initial mit der in MOVISUITE® durchgeführten Konfiguration des Roboters beschrieben.

Die Variablen werden durch die Robotersteuerung übernommen, wenn der Roboter nicht freigegeben ist, also die Variable *Interface_Robot.Basic.OUT.eEnableState* einen der folgenden Zustände ausgibt (Namensraum: SEW_MK_Robotics.SEW_IRobHPub).

- EmergencyStoppedAxes
- EmergencyStoppingAxes
- WaitingForSetpointActive

Das ist u. a. der Fall, wenn *Interface_Robot.Basic.IN.xEnable_EmergencyStop* "FALSE" oder *Interface_Robot.xError* "TRUE" ist.

HINWEIS

Bei Anwendungen, die eine stetige Änderung der Werkzeugtransformation während der Roboterbewegung erfordern, wenden Sie sich an SEW-EURODRIVE, da die hierzu erforderliche Schnittstelle in der aktuellen Version der Software nicht in der Anwenderschnittstelle vorhanden ist.

11.6.1 CartesianLimits

Variablenname	Beschreibung
alrPositive	Datentyp - ARRAY [1..6] OF LREAL
	Positive kartesische Softwareendschalter
alrNegative	Datentyp - ARRAY [1..6] OF LREAL
	Negative kartesische Softwareendschalter

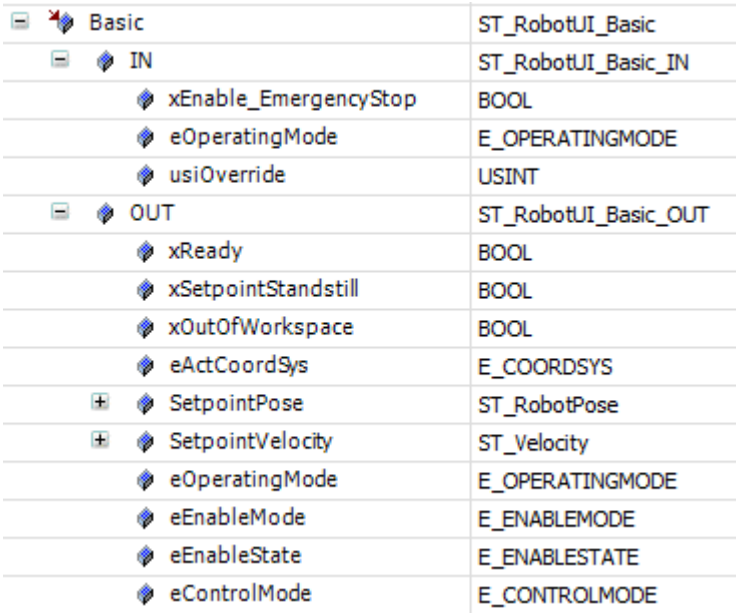
11.6.2 Transformations

Variablenname	Beschreibung
alrTool	Datentyp - ARRAY [1..6] OF LREAL
	Werkzeugtransformation

26873346/DE – 07/2021

11.7 Grundfunktionen (Basic)

Schnittstelle im IEC-Editor



18014423225395339

11.7.1 IN

Variablenname	Beschreibung
xEnable_EmergencyStop	Datentyp - BOOL
	• TRUE - Softwaremodul freigeben • FALSE - Das Softwaremodul führt einen Not-Halt mit der eingestellten Not-Halt-Rampe aus
eOperatingMode	Datentyp - E_OPERATINGMODE
	Betriebsart. "Betriebsart" (→ 🗎 31) • Manual_WithHighSpeed • Automatic Namensraum: *SEW_MK_Robotics.SEW_IRobHPub*
usiOverride	Datentyp - USINT
	Prozentuale Skalierung der Geschwindigkeit in allen Steuerungsarten

11.7.2 OUT

Variablenname	Beschreibung
xReady	Datentyp - BOOL
	• TRUE - Softwaremodul ist betriebsbereit (bereit für das freigeben) • FALSE - Softwaremodul ist nicht betriebsbereit. Nicht alle Voraussetzungen für ein erfolgreiches Freigeben sind erfüllt (Fehlermeldung mit der Ursache wird gemeldet, wenn dennoch freigegeben wird).

Variablenname	Beschreibung
xSetpointStandstill	Datentyp - BOOL
	• TRUE - Stillstand der Sollpose • FALSE - Sollpose ist nicht im Stillstand. Der Roboter bewegt sich.
xOutOfWorkspace	Datentyp - BOOL
	• TRUE - Roboter befindet sich außerhalb des erlaubten Arbeitsraums • FALSE - Roboter befindet sich innerhalb des erlaubten Arbeitsraumes.
eActCoordSys	Datentyp - E_COORDSYS
	Koordinatensystem, in dem aktuell die Bahn interpoliert wird (Base, Joint, World, User). Namensraum: *SEW_MK_Robotics.SEW_IRobMoCPub*
eOperatingMode	Datentyp - E_OPERATINGMODE
	Betriebsart. "Betriebsart" (→ 🗎 31) • Manual_WithHighSpeed • Automatic Namensraum: *SEW_MK_Robotics.SEW_IRobHPub*
eEnableMode	Datentyp - E_ENABLEMODE
	Aktuelle Freigabeart. "Freigabeart" (→ 🗎 32) • EmergencyStop_Axes • EmergencyStop • ApplicationStop • Enable Namensraum: *SEW_MK_Robotics.SEW_IRobHPub*
eEnableState	Datentyp - E_ENABLESTATE
	Freigabezustand. "Freigabezustand" (→ 🗎 34) • EmergencyStoppedAxes • EmergencyStoppingAxes • WaitingForSetpointActive • EmergencyStoppingOnPath • PositionHoldControl • ApplicationStoppingOnPath • WaitingForMotionCommand • PathMotion Namensraum: *SEW_MK_Robotics.SEW_IRobHPub*

26873346/DE – 07/2021

Variablenname	Beschreibung
eControlMode	Datentyp - E_CONTROLMODE
	Steuerungsart. "Steuerungsart" (→ 🗎 35) • Inactive • JogMode • ProgramMode Namensraum: *SEW_MK_Robotics.SEW_IRobHPub*

SetpointPose

Variablenname	Beschreibung
alrActCoordSys	Datentyp - ARRAY [1..6] OF LREAL
	Sollpose im aktuellen Koordinatensystem (Basic.Out.eAct-CoordSys)
alrBase	Datentyp - ARRAY [1..6] OF LREAL
	Sollpose im Basiskoordinatensystem
alrKinematicAxis	Datentyp - ARRAY [1..6] OF LREAL
	Sollposition der Einzelachsen des Kinematikmodells
alrKinematicJoint	Datentyp - ARRAY [1..6] OF LREAL
	Sollposition der Gelenkachsen des Kinematikmodells
alrAdditionalJoint	Datentyp - ARRAY [1..6] OF LREAL
	Sollposition der Zusatzgelenkachsen (Zusatzgelenkachsen sind im aktuellen Funktionsumfang nicht enthalten)
usiConstellation	Datentyp - USINT
	Konstellation der Kinematik in der Sollpose

SetpointVelocity

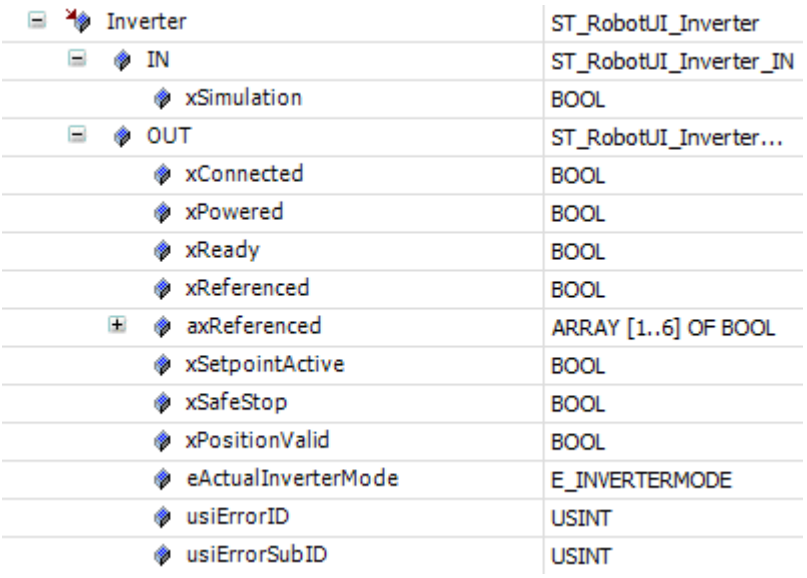
ActCoordSys

Variablenname	Beschreibung
lrTranslation	Datentyp - LREAL - Gleitkommazahl
	Translationsgeschwindigkeit des TCP im aktuellen Koordinatensystem

26873346/DE – 07/2021

11.8 Umrichterfunktionen (Inverter)

Schnittstelle im IEC-Editor



27021622481272971

11.8.1 IN

Weitere Informationen finden Sie im Kapitel "Simulation" (→ 🗎 42).

Variablenname	Beschreibung
xSimulation	Datentyp - BOOL
	• TRUE - Frequenzumrichter des Softwaremoduls simulieren (z. B. bei einem Test ohne Hardware) • FALSE - Frequenzumrichter nicht simulieren

11.8.2 OUT

Variablenname	Beschreibung
xConnected	Datentyp - BOOL
	• TRUE - Kommunikationsverbindung zum Controller existiert • FALSE - Keine Kommunikationsverbindung zum Controller
xPowered	Datentyp - BOOL
	• TRUE - Endstufen freigegeben (liefern Ausgangsspannung) • FALSE - Endstufen nicht freigegeben
xReady	Datentyp - BOOL
	• TRUE - Für Steuerung durch den Controller bereit • FALSE - Nicht bereit für Steuerung durch den Controller
xReferenced	Datentyp - BOOL
	• TRUE - Referenziert • FALSE - Nicht referenziert

26873346/DE – 07/2021

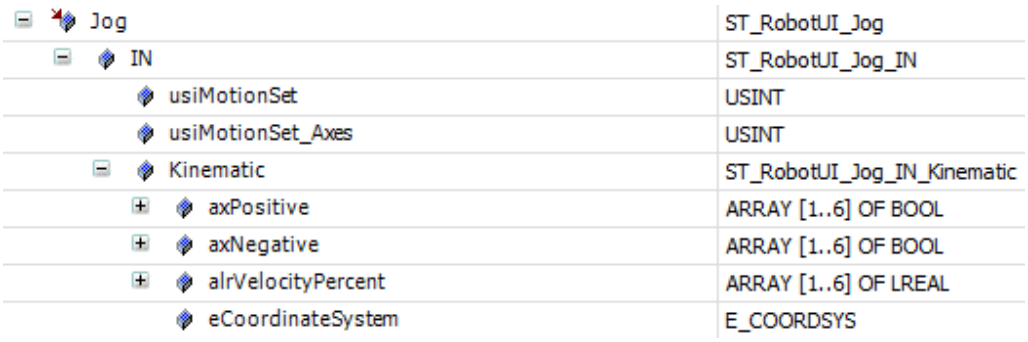
Variablenname	Beschreibung
axReferenced	Datentyp - ARRAY[1..6] OF BOOL
	• TRUE - Jeweilige Achse ist referenziert. • FALSE - Jeweilige Achse ist nicht referenziert.
xSetpointActive	Datentyp - BOOL
	• TRUE - Sollwerte werden verarbeitet. • FALSE - Sollwerte werden nicht verarbeitet.
xSafeStop	Datentyp - BOOL
	• TRUE - Die Achse steht (STO aktiv) • FALSE - Die Achse steht nicht (STO ist nicht aktiv)
xPositionValid	Datentyp - BOOL
	• TRUE - Die Position des Gebers ist gültig • FALSE - Für den Geber liegt ein Fehler vor. (z.B. bei einem Vogelschlag)
eActualInverterMode	Datentyp - E_INVERTERMODE
	Betriebsart des Umrichters (FCB des Umrichters): • Unknown • Standard • OuputDisabled (FCB01) • ManualMode (FCB04) • Stop (FCB02) • Homing (FCB12) • JogMode (FCB20) • BrakeTest (FCB21) • Positioning (FCB09) • PositioningInterpolated (FCB10) • Velocity (FCB05) • VelocityInterpolated (FCB06) • Torque (FCB07) • TorqueInterpolated (FCB08) • MotorParamMeasurement (FCB25) • PosHoldCtrl (FCB19) • RotorPosIdentification (FCB18) • ApplicationStop (FCB13) • EmergencyStop (FCB14) • UserStop (FCB26) *Bibliothek: SEW DeviceHandler Interfaces*
usiErrorID	Datentyp - USINT
	Fehler-ID

26873346/DE – 07/2021

Variablenname	Beschreibung
usiErrorSubID	Datentyp - USINT
	Sub-Fehler-ID

11.9 Software-Endschalter (SoftwareLimitSwitch)

Schnittstelle im IEC-Editor



18014423226473611

Weitere Informationen finden Sie im Kapitel "Software-Endschalter" (→ 🗎 43).

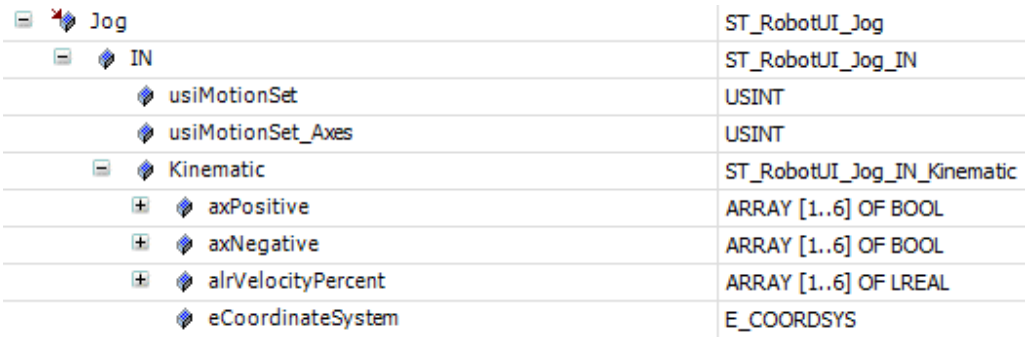
11.9.1 IN

Variablenname	Beschreibung
xDisable	Datentyp - BOOL
	• TRUE – Alle Software-Endschalter des Softwaremoduls (Gelenkachsen und kartesische Achsen) deaktivieren. Die Software-Endschalter der unterlagerten Softwaremodule und Geräte werden nicht deaktiviert • FALSE – Software-Endschalter des Softwaremoduls nicht deaktivieren.

26873346/DE – 07/2021

11.10 Tippen (Jog)

Schnittstelle im IEC-Editor



18014423226473611

Weitere Informationen finden Sie im Kapitel "Tippbetrieb" (→ 🗎 36).

11.10.1 IN

Variablenname	Beschreibung
usiMotionSet	Datentyp - USINT
	Bewegungsparametersatz für den Tippbetrieb beim Verwenden des Koordinatesystems "Joint" oder "Base"
usiMotionSet_Axes	Datentyp - USINT
	Bewegungsparametersatz für den Tippbetrieb bei Verwendung des Koorinatensystem "Axis"

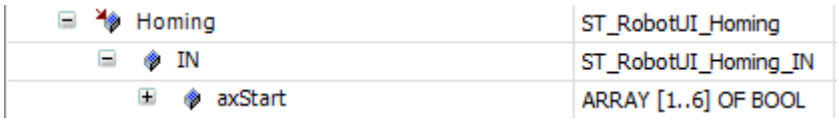
Kinematic

Variablenname	Beschreibung
axPositive[1..6]	Datentyp - ARRAY [1..6] OF BOOL
	Tippen in positive Richtung der jeweiligen Achse des Bezugskoordinatensystems (Jog.IN.Kinematic.eCoordinateSystem)
axNegative[1..6]	Datentyp - ARRAY [1..6] OF BOOL
	Tippen in negative Richtung der jeweiligen Achse des Bezugskoordinatensystems (Jog.IN.Kinematic.eCoordinateSystem)
alrVelocityPercent[1..6]	Datentyp - ARRAY [1..6] OF LREAL
	Prozentuale Skalierung der Tippgeschwindigkeit der jeweiligen Achse des Bezugskoordinatensystems (Jog.IN.Kinematic.eCoordinateSystem)
eCoordinateSystem	Datentyp - E_COORDSYS
	Bezugskoordinatensystem in dem getippt werden soll. Aktuell unterstützt: Gelenkachsen (Joint), kartesische Achsen des Basiskoordinatensystems (Base) und Einzelachsen (Axis). Namensraum: *SEW_MK_Robotics.SEW_IRobMoC-Pub*

26873346/DE – 07/2021

11.11 Referenzieren (Homing)

Schnittstelle im IEC-Editor



33225656075

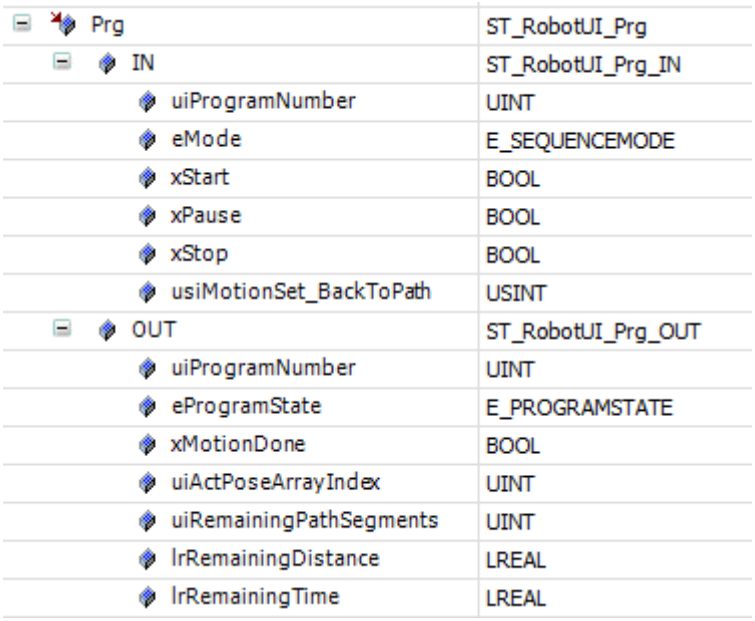
Weitere Informationen finden Sie im Kapitel "Referenzierbetrieb" (→ 🖹 40).

11.11.1 IN

Variablenname	Beschreibung
axStart	Datentyp - ARRAY[1..6] OF BOOL
	• TRUE - Referenzierbetrieb an der jeweiligen Achse starten.

11.12 Programmsteuerung (Prg)

Schnittstelle im IEC-Editor



18014423226476683

Weitere Informationen finden Sie im Kapitel "Programmbetrieb" (→ 🖹 37).

11.12.1 IN

Variablenname	Beschreibung
uiProgramNumber	Datentyp - UINT
	Nummer des Hauptprogramms, das sich in Ausführung befindet. Dies schließt nicht aus, dass gerade ein Unterprogramm ausgeführt wird.

26873346/DE – 07/2021

Variablenname	Beschreibung
eMode	Datentyp - E_SEQUENCEMODE
	Programmablaufart (AUTO, STEP_BLOCK) Kapitel "Programmbetrieb" (→ 🗎 37). Namensraum: *SEW_MK_Robotics.SEW_Rob.SEW_IRobLang*
xStart	Datentyp - BOOL
	• TRUE - Programmabarbeitung starten • FALSE - In der Betriebsart "Manuell mit hoher Geschwindigkeit": Programmabarbeitung pausieren.
xPause	Datentyp - BOOL
	• TRUE - Applikationshalt durchführen und die Programmabarbeitung pausieren. • FALSE - Programmabarbeitung nicht pausieren.
xStop	Datentyp - BOOL
	• TRUE - Applikationshalt durchführen und die Programmabarbeitung stoppen. • FALSE - Programmabarbeitung nicht stoppen.
usiMotionSet_BackToPath	Datentyp - USINT
	Bewegungsparametersatz für die "Rückpositionierung (BackToPath)" (→ 🗎 39) auf die Bahn

11.12.2 OUT

Variablenname	Beschreibung
uiProgramNumber	Datentyp - UINT
	Nummer des Hauptprogramms, das sich in Ausführung befindet. Dies schließt nicht aus, dass gerade ein Unterprogramm ausgeführt wird.
eProgramState	Datentyp - E_PROGRAMSTATE
	Status der Programmabarbeitung. "Programmbetrieb" (→ 🗎 37) • NotInitialized • Initialized • BeeingExecuted • Paused • Finished • BackToPathRequired • BackToPathActive Namensraum: *SEW_MK_Robotics.SEW_IRobHPub*

26873346/DE – 07/2021

Variablenname	Beschreibung
xMotionDone	Datentyp - BOOL
	• TRUE - Alle Bewegungsaufträge und Bahnereignisse mit Nachlaufzeit ausgeführt. • FALSE - Noch nicht alle Bewegungsaufträge ausgeführt.
uiActPoseArrayIndex	Datentyp - UINT
	Listenindex der Posenvariable (PrgVar.astPoseValues), die im aktuellen Bewegungsbefehl als Zielpose verwendet wird. Bei einem Bewegungsbefehl die keine Posenvariable verwendet, wird der Index 0 ausgegeben.
uiRemainingPathSegments	Datentyp - UINT
	Anzahl der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden.
uiRemainingDistance	Datentyp - UINT
	Verbliebene Reststrecke der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden.
uiRemainingTime	Datentyp - UINT
	Verbliebene Restzeit der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden.
uiNumberOfUserCoordSysChanges	Datentyp - UINT
	Anzahl der Wechsel in ein USER-Koordinatensystem und zwischen verschiedenen USER-Koordinatensystemen. Wird im Programmbetrieb bei Wechsel in das Koordinatensystem BASE auf 0 zurückgesetzt. Ein Überlauf erfolgt direkt auf den Wert 1.

Anmerkung:

Die Bewegungssteuerung berücksichtigt bei den *RemainingPathSegments*, der *RemainingDistance* und der *RemainingTime* nur die Bahnsegmente, die bereits vom SRL-Programm übergeben wurden (siehe Programmzeiger).

26873346/DE – 07/2021

11.13 Physiksimulation (Physics)

Schnittstelle im IEC-Editor

Physics	ST_RobotUI_Physics
OUT	ST_RobotUI_Physics_OUT
usiNumberOfDrives	USINT
Drives	ARRAY [1..6] OF SEW_IPhDr.ST_DriveOut
Drives[1]	SEW_IPhDr.ST_DriveOut
Drives[2]	SEW_IPhDr.ST_DriveOut
Drives[3]	SEW_IPhDr.ST_DriveOut
Drives[4]	SEW_IPhDr.ST_DriveOut
Drives[5]	SEW_IPhDr.ST_DriveOut
Drives[6]	SEW_IPhDr.ST_DriveOut

33965130763

Weitere Informationen finden Sie im Kapitel "Funktionsbeschreibung" (→ 🗎 44) und im Kapitel zum dazugehörigen "Konfigurationsmenü" (→ 🗎 126).

11.13.1 OUT

Variablenname	Beschreibung
usiNumberOfDrives	Datentyp - USINT
	Anzahl der physikalisch simulierten Antriebe
Drives	Datentyp - ARRAY OF SEW_IPhDr.ST_DriveOut
	Relevante Ausgangswerte der physikalisch simulierten Antriebe (Drehmoment, Leistung, ...)

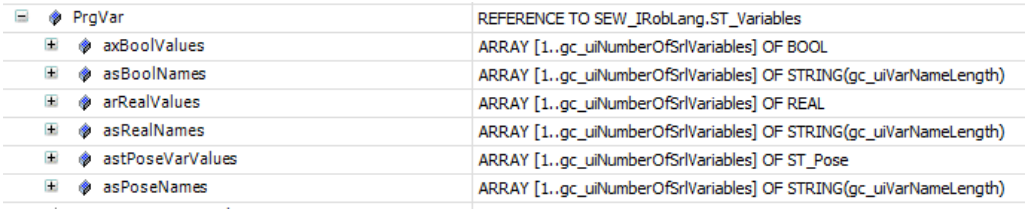
26873346/DE – 07/2021

11.14 SRL-Programmvariablen (PrgVar)

HINWEIS

Die SRL-Programmvariablen werden nicht in der Task *Main* gemappt (durch MapIn/ MapOut). Sie arbeiten direkt auf der Referenz der internen Variablen. Sobald die Zuweisung im Anwenderprogramm ausgeführt wird, wirkt der zugewiesene Wert für alle ab sofort ausgeführten Befehle im SRL-Programm. Die SRL-Programmvariablen sind von der Zugriffsverwaltung ausgenommen.

Schnittstelle im IEC-Editor



9007223971741835

Variablenname	Beschreibung
axBoolValues	Datentyp - ARRAY of BOOL
	Liste der Werte der Boolvariablen
asBoolNames	Datentyp - ARRAY OF STRING
	Liste der Namen der Boolvariablen
arRealValues	Datentyp - ARRAY OF REAL
	Liste der Werte der Gleitkommavariablen
asRealNames	Datentyp - ARRAY OF STRING
	Liste der Namen der Gleitkommavariablen
astPoseValues	Datentyp - ARRAY OF ST_Pose
	Liste der Werte der Posenvariablen
asPoseNames	Datentyp - ARRAY OF STRING
	Liste der Namen der Posenvariablen

26873346/DE – 07/2021

11.15 IEC-Funktionsaufruf für das SRL-Programm

Erstellen Sie zum Aufruf einer IEC-Funktion aus dem SRL-Programm im IEC-Programm folgende Funktion:

1. Legen Sie einen neuen Funktionsblock an (Kontextmenü / Objekt hinzufügen / POU… / Typ: Funktionsbaustein).
2. Implementieren Sie mittels des Funktionsbausteins folgendes Interface: *SEW_MK_Robotics.SEW_IRobLangPub.ICallF*
3. Implementieren Sie alle Schnittstellen (Kontextmenü des Funktionsbausteins / Schnittstellen implementieren…).
4. Instanzieren Sie den Funktionsbaustein im Programm *User_PRG*

```
1  PROGRAM User_PRG
2  VAR_OUTPUT
3      xInitDone : BOOL;
4  END_VAR
5  VAR
6      fbCallFunction: CallFunction;
7  END_VAR
```

9007225120208267

5. Verbinden Sie den Funktionsbaustein mit dem Roboter, in dem Sie in der *User_PRG.Init* folgende Codezeile hinzufügen ("INSTANCENAME" durch die Benennung des Roboters im MOVISUITE®-Projekt ersetzen): `INSTANCENAME.fbProgramInterpreter.LinkICallF(fbCallFunction);`
6. Programmieren Sie in der Methode *CallF* eine Fallunterscheidung abhängig von der Variable *uiCallIndex*.

CallFunction.CallF

```
1  METHOD CallF : BOOL
2  VAR_INPUT
3      uiCallIndex : UINT;
4  END_VAR
5
```

```
1  CASE uiCallIndex OF
2      1: // User function 1
3          CallF:=TRUE; // Success of user function
4      2: // User function 2
5          CallF:=TRUE; // Success of user function
6  ELSE
7      // Set error
8  END_CASE
```

25865677067

7. Programmieren Sie die gewünschten Funktion in den entsprechenden Zweig der Fallunterscheidung. Der nächste SRL-Befehl wird immer erst nach Setzen von *CallF* auf "TRUE" abgearbeitet.

Weitere Informationen zum Aufruf der IEC-Funktion im RobotMonitor finden Sie im Kapitel "IEC-Funktionsaufruf" (→ 🗎 179).

12 Prozessdatenbelegung

HINWEIS

Berücksichtigen Sie unbedingt das Hochlaufverhalten des MOVI-C® CONTROLLER am Feldbus. Weitere Informationen dazu finden Sie im Kapitel "Hochlaufverhalten" (→ 🗎 99).

In diesem Kapitel sind die Signale der Prozessdatenschnittstelle über Feldbus beschrieben. In welcher Reihenfolge die Signale angesteuert werden sollen, ist im Kapitel "Ansteuerung durch die Prozess-Steuerung" (→ 🗎 130) beschrieben.

12.1 Feldbus-Schnittstelle

Das Softwaremodul bietet zum Ansteuern von einer übergeordneten Steuerung aus eine standardisierte Feldbus-Schnittstelle an.

HINWEIS

Die Feldbus-Schnittstelle muss über das Konfigurationsmenü "Feldbus-Schnittstelle" (→ 🗎 122) nach abgeschlossener Konfiguration des Softwaremoduls aktiviert werden.

12.1.1 Feldbusprofile

HINWEIS

Der "MOVIKIT® Feldbusmonitor" (→ 🗎 218) unterstützt in der aktuellen Version nur das Feldbusprofil "Standardprofil für die Positionierung". Das Feldbusprofil "Flexible, parametrierbare Profil" wird nicht unterstützt.

Standardprofil für die Positionierung

HINWEIS

Für das "Standardprofil für die Positionierung" ist es erforderlich, dass ein Standardprogramm mit der entsprechenden Anzahl von Bahnsegmenten geladen wurde. Siehe dazu Kapitel "Standardprogramm laden" (→ 🗎 150).

Das Standardprofil für die Positionierung kann verwendet werden, um das Softwaremodul auf einfache Art und Weise von der übergeordneten Steuerung zu positionieren. Die Bahn wird dabei von der übergeordneten Steuerung verwaltet und an das Softwaremodul übermittelt. Die Schnittstelle und das SRL-Programm sind fest definiert.

Die Bahn wird aus Bahnsegmenten zusammengesetzt. Jedes Bahnsegment wird dabei durch eine Zielpose, eine Überschleifdistanz und Bewegungsparametersatz definiert. Siehe Kapitel "Standardprogramm laden" (→ 🗎 150). Beim Start wird die Bahn abgefahren. Anschließend können weitere Bahnen übermittelt und gestartet werden. Weitere Informationen zur Ansteuerung finden Sie im Kapitel "Prozessablauf" (→ 🗎 133).

26873346/DE – 07/2021

Die Bahn besteht aus Bewegungssequenzen (ohne Halt), die von der übergeordneten Steuerung zur Laufzeit durch Bahnpunkte und Überschleifbereiche vorgegeben werden. Durch diverse Freigaben wird gesteuert, wann der Roboter welche Bewegung durchführen soll. Mit welcher Geschwindigkeit und Beschleunigung der Roboter diese Bahnsegmente abfährt, wird durch vorkonfigurierbare Bewegungsparametersätze definiert. Die übergeordneten Steuerung gibt für jedes Bahnsegment die Nummer des zu verwendenden Bewegungsparametersatzes an. Beispielsweise können Datensätze für Eilgang, Schleichgang oder Greifbewegungen definiert werden.

Flexibles parametrierbares Profil

Mit dem flexiblen, parametrierbaren Profil kann das Softwaremodul von der übergeordneten Steuerung über Feldbus auf vielfältige Weise angesteuert werden. Das SRL-Programm kann dabei selbst im RobotMonitor programmiert und die verwendete SRL-Programmvariablen als Ein- oder Ausgangssignale auf den Feldbus gelegt werden. Damit ist mit diesem Profil auch eine Tabellenpositionierung des Roboters möglich.

Wie beim "Standardprofil für die Positionierung" sind dabei die ersten 4 Prozessdatenworte fest vorgegeben. Die optionalen Diagnosemodule, die Standard-Bahnsegmente sowie weitere SRL-Programmvariablen (Posen, Reals, Bools) können individuell hinzukonfiguriert werden. Siehe "Feldbus-Schnittstelle" (→ 🗎 122).

12.1.2 Konsistente Datenübertragung

HINWEIS

Für eine konsistente Datenübertragung zum MOVI-C® CONTROLLER müssen die Prozessdaten in der übergeordneten Steuerung entsprechend konfiguriert werden. Im Feldbus-Master müssen dazu Konsistenzblöcke verwendet werden, die zum ausgewählten Prozessdatenprofil passen. Dabei wird empfohlen, die Roboterinstanz und den konsistenten Block beim gleichen Wort beginnen zu lassen.

HINWEIS

Eine konsistente Datenübertragung über mehrere konsistente Blöcke wird von der aktuellen Version der Software nicht unterstützt. Daher ist die Ansteuerung des Handshake-Bits für eine konsistente Datenübertragung aktuell nicht erforderlich, wird jedoch empfohlen.

Für weitere Fragen und eine applikative Lösung für die konsistente Datenübertragung über mehrere konsistente Blöcke über den Feldbus wenden Sie sich bitte an SEW-EURODRIVE.

Der MOVI-C® CONTROLLER unterstützt konsistente Datenblöcke bis zu 64 Prozessdatenwörter (Siehe Handbuch des MOVI-C® CONTROLLER). Damit ist eine konsistente Datenübertragung für das "Standardprofil für die Positionierung" mit bis zu 10 Standard-Bahnsegmenten sichergestellt. Für das Feldbusprofil "Flexibles, parametrierbares Profil" ist dies individuell zu betrachten.

26873346/DE – 07/2021

12.1.3 Ansteuerung des Handshake-Bits

Die Ansteuerung des Handshake-Bits wird empfohlen, um zu überprüfen, ob die Daten der unterlagerten Steuerung noch korrekt empfangen, verarbeitet und hochgemeldet werden. Dafür muss auf der überlagerten Steuerung folgende Logik implementiert werden:

```
IF Handshake-Out = Handshake-In THEN
  Handshake-In invertieren
ELSE
  Zeitüberwachung, mit Fehler bei Zeitüberschreitung
END_IF
```

12.1.4 Steuerwort 1-4

Die Steuerwörter 1-4 sind in jedem Feldbusprofil vorhanden und können nicht angepasst werden. Jede Roboterinstanz auf dem Feldbus beginnt mit diesen 4 Prozessdatenwörtern.

Wort	Bit	Funktion	Beschreibung
PA 1	0	Freigabe/Not-Halt	Softwaremodul freigeben
	…	…	…
	8	Quittierung Fehler/Meldung	Fehler/Meldungen zurücksetzen
	9	Simulation	Frequenzumrichter des Softwaremoduls simulieren
	…	…	…
	12	Softwareendschalter deaktivieren	Deaktiviert alle Software-Endschalter des Softwaremoduls (Gelenkachsen und kartesische Achsen). Die Software-Endschalter der unterlagerten Softwaremodule und Geräte werden nicht deaktiviert. Siehe auch "Software-Endschalter" (→ 🗎 43).
	15	MOVIKIT® Handshake In	Dieses Signal wird intern auf das Statuswort Bit 15 (MOVIKIT® Handshake Out) kopiert. Sollte der Kopiervorgang fehlschlagen ("Handshake Out"-Signal bleibt konstant bei wechselndem "Handshake In"-Signal), ist die geräteinterne Bearbeitung des Softwaremoduls gestört.

26873346/DE – 07/2021

Wort	Bit	Funktion	Beschreibung
PA 2	0	Automatik/Manuell	Sollwert der "Betriebsart" (→ 🗎 31)
	…	…	
	2	Programmstart	Programmabarbeitung starten
	3	Programmpause	Applikationshalt durchführen und die Programmabarbeitung pausieren
	4	Programmstopp	Applikationshalt durchführen und die Programmabarbeitung stoppen
	…	…	
	10	Start Referenzieren Achse 1	Start des Referenzierens der jeweiligen Achse im "Referenzierbetrieb" (→ 🗎 40).
	11	Start Referenzieren Achse 2	
	12	Start Referenzieren Achse 3	
	13	Start Referenzieren Achse 4	
	14	Start Referenzieren Achse 5	
	15	Start Referenzieren Achse 6	

26873346/DE – 07/2021

Wort	Bit	Funktion	Beschreibung
PA 3	0	Tippen Gelenk-/ Einzel- /kartesische Achse 1 positiv	Tippen in positive Richtung der jeweiligen Achse des Bezugskoordinatensystems (Bit 12)
	1	Tippen Gelenk-/ Einzel- /kartesische Achse 2 positiv	
	2	Tippen Gelenk-/ Einzel- /kartesische Achse 3 positiv	
	3	Tippen Gelenk-/ Einzel- /kartesische Achse 4 positiv	
	4	Tippen Gelenk-/ Einzel- /kartesische Achse 5 positiv	
	5	Tippen Gelenk-/ Einzel- /kartesische Achse 6 positiv	
	6	Tippen Gelenk-/ Einzel- /kartesische Achse 1 negativ	Tippen in negative Richtung der jeweiligen Achse des Bezugskoordinatensystems (Bit 12)
	7	Tippen Gelenk-/ Einzel- /kartesische Achse 2 negativ	
	8	Tippen Gelenk-/ Einzel- /kartesische Achse 3 negativ	
	9	Tippen Gelenk-/ Einzel- /kartesische Achse 4 negativ	
	10	Tippen Gelenk-/ Einzel- /kartesische Achse 5 negativ	
	11	Tippen Gelenk-/ Einzel- /kartesische Achse 6 negativ	
	12	Tipp-Koordinatensystem: JCS/BCS	Bezugskoordinatensystem in dem getippt werden soll. FALSE - Basiskoordinatensystem (BCS) (wenn auch PA3.13 FALSE ist) TRUE - Gelenkkoordinatensystem (JCS)
	13	Tipp-Koordinatensystem ACS/BCS	Bezugskoordinatensystem in dem getippt werden soll. FALSE - Basiskoordinatensystem (BCS) (wenn auch PA3.12 FALSE ist) TRUE - Einzelachskoordinatensystem (ACS)
PA 4	0-7	Programmnummer	Nummer des Hauptprogramms, das ausgeführt werden soll.
	8-15	Override	Prozentuale Skalierung der programmierten Geschwindigkeit in allen Steuerungsarten

26873346/DE – 07/2021

12.1.5 Statuswort 1-4

Die Statuswörter 1-4 sind in jedem Feldbusprofil vorhanden und können nicht angepasst werden. Jede Roboterinstanz auf dem Feldbus beginnt mit diesen 4 Prozessdatenwörtern.

Wort	Bit	Funktion	Beschreibung
PE 1	0	Betriebsbereit (Ready)	Softwaremodul ist betriebsbereit
	1	STO-Freigabe (STO aktiv)	TRUE - Die Achse steht nicht (STO ist nicht aktiv) FALSE - Die Achse steht (STO aktiv)
	2	Endstufenfreigabe	Endstufen freigegeben
	…	…	
	4	Motoren drehen (Motorstillstand)	TRUE - Sollpose ist nicht im Stillstand. Der Roboter bewegt sich FALSE - Stillstand der Sollpose
	5	Referenziert	Umrichter referenziert
	6	Sollwert aktiv	Sollwerte werden verarbeitet
	…	…	
	8	Fehler Modul	Ein Fehler liegt vor
	9	Warnung Modul	Eine Warnung liegt vor
	…	…	
	15	MOVIKIT® Handshake Out	Dieses Signal wird intern von das Steuerwort Bit 15 (MOVIKIT® Handshake In) kopiert. Sollte der Kopiervorgang fehlschlagen ("Handshake Out"-Signal bleibt konstant bei wechselndem "Handshake In"-Signal), ist die geräteinterne Bearbeitung des Softwaremoduls gestört.

26873346/DE – 07/2021

Wort	Bit	Funktion	Beschreibung
PE 2	0	Automatik/Manuell	Aktuelle "Betriebsart" (→ 🗎 31). FALSE - Manuell TRUE - Automatik
	…	…	
	2	Programm initialisiert	Status der "Programmabarbeitung" (→ 🗎 37).
	3	Programm wird ausgeführt	
	4	Programm angehalten	
	5	Programm abgeschlossen	
	6	Rückpositionierung erforderlich	
	7	Rückpositionierung aktiv	
	…	…	
	10	Achse 1 referenziert	Status des "Referezierbetriebs" (→ 🗎 40): TRUE - Die jeweilige Achse ist referenziert.
	11	Achse 2 referenziert	
	12	Achse 3 referenziert	
	13	Achse 4 referenziert	
	14	Achse 5 referenziert	
	15	Achse 6 referenziert	
PE 3	0-15	Message ID	Identifikationsnummer der Meldung (Fehler, Warnung, Information). Anhand der Bits PE 1:8 und PE 1:9 kann die Schwere der Meldung ermittelt werden: Fehler: Nur PE 1:8 ist "TRUE" Warnung: Nur PE 1:9 ist "TRUE" Information: PE 1:8 und 1:9 sind "FALSE"
PE 4	0-7	Programmnummer	Nummer des Hauptprogramms, das sich in Ausführung befindet. **HINWEIS:** Dies schließt nicht aus, dass gerade ein Unterprogramm ausgeführt wird.
	8	Tippen	"Steuerungsart" (→ 🗎 35)
	9	Referenzieren	
	10	Programm	
	…	…	

26873346/DE – 07/2021

12.1.6 Standardprofil für die Positionierung

Die Länge des "Standardprofil für die Positionierung" ist abhängig von der maximalen Anzahl der Standard-Bahnsegmente:

- n ≤ 2 : Länge = 16
- n ≥ 2 : Länge = 4+6×n

Prozessausgangsdaten

HINWEIS

Für das "Standardprofil für die Positionierung" ist es erforderlich, dass ein Standardprogramm mit der entsprechenden Anzahl von Bahnsegmenten geladen wurde. Siehe dazu Kapitel "Standardprogramm laden" (→ 🗎 150).

Folgende Tabelle zeigt die Prozessausgangsdaten von der übergeordneten Steuerung zum Softwaremodul bei Ansteuerung über den Feldbus.

Wort	Funktion	Erklärung
PA 1-4	Steuerworte 1-4	Grundsignale zur Ansteuerung eines Roboters über Feldbus, siehe Kapitel "Steuerwort 1-4" (→ 🗎 206).
PA 5-10	Standard-Bahnsegment 1	Ab PA 5 liegen die Standard-Bahnsegmente, die jeweils 6 Prozessdatenworte lang sind. Die Anzahl der Bahnsegmente hängt von der konfigurierten "maximalen Anzahl der Standard-Bahnsegmente" (→ 🗎 122) ab.
PA 11-16	Standard-Bahnsegment 2	
usw.	usw.	

Standard-Bahnsegmente im Standardprofil

HINWEIS

Die Nummerierung der Prozessdaten bezieht sich nicht auf die absolute Nummer in der Feldbus-Schnittstelle, sondern ist nur die relative Nummer im Standard-Bahnsegment.

26873346/DE – 07/2021

Die Form des Standard-Bahnsegments ist eine Gerade (LIN).

Wort	Bit	In	Beschreibung
PA 1	0-3	Bewegungsparametersatz	Auswahl des Bewegungsparametersatzes für dieses Bahnsegment über die 4 Bits als NIBBLE.
	4	Wartepunkt	TRUE - Das Anhalten an der vorherigen Pose wird erzwungen. Es muss dazu seit Programmstart "TRUE" gesetzt sein. FALSE - Freigabe zum Abfahren dieses Bahnsegments wird erteilt, falls die Bahn nicht bereits durch ein Bahnende (Bit 5) beendet wurde.
	5	Bahnende	TRUE - Beenden des Programms an dieser Zielpose. Nur das erste Bahnende (über alle Bahnsegmente im gesamten Telegramm hinweg) ist wirksam. Das Signal muss zwischen Programmstart und Programmende auf "TRUE" gesetzt bleiben.
	…	…	
PA 2	0-15	Überschleifdistanz	Überschleifdistanz zwischen dem letzten und diesem Bahnsegment (also bei der Ziel-Pose des letzten Bahnsegments)
PA 3 - 6	0-15	Ziel-Pose	Vorgabe der Zielpose für dieses Bahnsegment (X, Y, Z- und A-Koordinate)

Prozesseingangsdaten

Folgende Tabelle zeigt die Prozesseingangsdaten vom Softwaremodul zur übergeordneten Steuerung bei Ansteuerung über den Feldbus.

Wort	Funktion	Erklärung
PE 1-4	Statusworte 1-4	Grundsignale zur Diagnose des Roboters über Feldbus. Siehe Kapitel "Statuswort 1-4" (→ 🗎 209).
PE 5-16	Modul zur Bewegungsdiagnose	Signale zur Diagnose der Bewegung des Roboters. Siehe Kapitel "Modul zur Bewegungsdiagnose im Standardprofil" (→ 🗎 213).

26873346/DE – 07/2021

Modul zur Bewegungsdiagnose im Standardprofil

Wort	Bit	In	Beschreibung
5	0	Arbeitsraum verlassen	Der Roboter befindet sich außerhalb des erlaubten Arbeitsraums.
	1	Aktueller Bewegungsauftrag ausgeführt	Alle Bewegungsaufträge und Bahnereignisse mit Nachlaufzeit ausgeführt.
	…	…	
	8-15	Konstellation	Sollwert der Konstellation der Kinematik
6	0-15	Translationsgeschwindigkeit	Translationsgeschwindigkeit des TCP im aktuellen Koordinatensystem
7	0-15	Sollwert-Pose im BCS (x-Koordinate)	Sollposition im Basiskoordinatensystem
8	0-15	Sollwert-Pose im BCS (y-Koordinate)	
9	0-15	Sollwert-Pose im BCS (z-Koordinate)	
10	0-15	Sollwert-Pose im BCS A-Koordinate)	
11	0-7	Aktuell angefahrene Pose	Nummer des Bahnsegmentes, das gerade abgefahren wird.
	8-15	Restliche Segmente	Anzahl der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden.
12	0-15	Zielentfernung	Verbliebene Reststrecke der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden
13	0-15	Sollwert-Pose im JCS (J1-Koordinate)	Sollpose der Gelenkachsen des Kinematikmodells
14	0-15	Sollwert-Pose im JCS (J2-Koordinate)	
15	0-15	Sollwert-Pose im JCS (J3-Koordinate)	
16	0-15	Sollwert-Pose im JCS (J4-Koordinate)	

26873346/DE – 07/2021

12.1.7 Flexibles parametrierbares Profil

Prozessausgangsdaten

Wort	Funktion	Erklärung
PA 1-4	Steuerworte 1-4	Grundsignale zur Ansteuerung des Roboters über Feldbus. Siehe "Steuerwort 1-4" (→ 🗎 206).
Ab PA 5	Optionale Prozessdaten-Module	Ab PA 5 können optionale Prozessdaten-Module hinzukonfiguriert "werden" (→ 🗎 122). Dabei gilt folgende Reihenfolge: - Standard-Bahnsegmente - Weitere Posen (SRL-Programmvariablen) - Weitere Reals (SRL-Programmvariablen) - Weitere Bools (SRL-Programmvariablen)

Prozesseingangsdaten

Wort	Funktion	Erklärung
PE 1-4	Statusworte 1-4	Grundsignale zur Diagnose des Roboters über Feldbus. Siehe "Statuswort 1-4" (→ 🗎 209).
Ab PA 5	Optionale Prozessdaten-Module	Ab PA 5 können optionale Prozessdaten-Module hinzukonfiguriert "werden" (→ 🗎 122). Dabei gilt folgende Reihenfolge: - Statuswort zur Bewegungsdiagnose - Weitere Posen (SRL-Programmvariablen) - Weitere Reals (SRL-Programmvariablen) - Weitere Bools (SRL-Programmvariablen)

Posen

Da die Posengröße auf dem Feldbus konfiguriert wird, variiert diese zwischen 2 und 12 Prozessdatenwörtern. Standardmäßig beträgt die Posengröße jedoch 4 Prozessdatenwörter (X, Y, Z, A). In den Standard-Bahnsegmenten, dem Modul zur Bewegungsdiagnose und den Posen der weiteren SRL-Programmvariablen werden alle Posen entsprechend der Konfiguration gemappt.

26873346/DE – 07/2021

Optionales Modul zur Bewegungsdiagnose mit variabler Größe

Wort	Bit	Funktion	Beschreibung
PE 5	1	Arbeitsraum verlassen	Roboter befindet sich außerhalb des erlaubten Arbeitsraums
	2	Aktueller Bewegungsauftrag ausgeführt	Alle Bewegungsaufträge und Bahnereignisse mit Nachlaufzeit ausgeführt
	…		
	8-15	Konstellation	Sollwert der Konstellation der Kinematik
PE 6	0-15	Translationsgeschwindigkeit	Translationsgeschwindigkeit des TCP im aktuellen Koordinatensystem.
PE 7 bis 6+p*	0-15	Sollwert-Pose im BCS	Sollposition im Basiskoordinatensystem
PE 7+p*	0-7	Aktuell angefahrene Pose	Listenindex der Posenvariable (*PrgVar.astPoseValues*), die im aktuellen Bewegungsbefehl als Zielpose verwendet wird. Bei einem Bewegungsbefehl der keine Posenvariable verwendet, wird der Index 0 ausgegeben.
	8-15	Restliche Segmente	Anzahl der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden.
PE 8+p*	0-15	Zielentfernung	Verbliebene Reststrecke der Bahnsegmente, die in der Bewegungsplanung berücksichtigt und noch nicht abgefahren wurden.
PE 9+p* bis 8+2p*	0-15	Sollwert-Pose im JCS	Sollpose der Gelenkachsen des Kinematikmodells

* p = Posengröße, siehe Kapitel "Posen" (→ 🗎 214). Standardwert: 4.

26873346/DE – 07/2021

Standard-Bahnsegmente mit variabler Größe

HINWEIS

Die Nummerierung der Prozessdaten bezieht sich nicht auf die absolute Nummer in der Feldbus-Schnittstelle, sondern ist nur die relative Nummer im Standard-Bahnsegment.

Wort	Bit	In	Beschreibung
PA 1	0-3	Bewegungsparameter-satz	Auswahl des Bewegungsparametersatzes für dieses Bahnsegment über die 4 Bits als NIBBLE.
	4	Wartepunkt	TRUE - Das Anhalten an der vorherigen Pose wird erzwungen FALSE - Freigabe zum Abfahren dieses Bahnsegments wird erteilt, falls die Bahn nicht bereits durch ein Bahnende (Bit 5) beendet wurde.
	5	Endpunkt	TRUE - Beenden des Programms an dieser Zielpose. Nur das erste Bahnende (über alle Bahnsegmente im gesamten Telegramm hinweg) ist wirksam.
	…	…	
PA 2	0-15	Überschleifdistanz	Überschleifdistanz zwischen dem letzten und diesem Bahnsegment (also bei der Ziel-Pose des letzten Bahnsegments)
PA 3 bis 2+p*	0-15	Ziel-Pose	Vorgabe der Zielpose für dieses Bahnsegment

* p=Posengröße, siehe Kapitel "Posen" (→ 🗎 214). Standardwert: 4.

26873346/DE – 07/2021

Weitere SRL-Programmvariablen

Die SRL-Programmvariablen (Posen, Reals, Bools) können sowohl als Prozesseingangs- als auch als Prozessausgangsdatenwort durch einfache Konfiguration auf den Feldbus gemappt werden. Siehe Kapitel "Weitere Posen" (→ 🗎 124). Diese werden am Ende der Feldbus-Schnittstelle angehängt. Die Signale werden dabei möglichst platzsparend gemappt.

Beispiel

Zuerst werden die Posen, dann die Reals und dann die Bools gemappt. Bei Datentypen kleiner WORD werden mehrere Real-Variablen in ein WORD gemappt (in folgendem Beispiel Real 2 und 3). Die Bool-Variablen füllen die durch die Real-Variablen angefangen WORDs auf (im Beispiel Bool 1-4) und werden dann im nächsten WORD fortgesetzt (im Beispiel Bool 1-4).

Konfiguration:

- 1 Pose mit X als DWORD, Y, Z und A als WORD
- 3 Reals als WORD, BYTE und NIBBLE
- 8 Bools

Feldbus-Schnittstelle:

Wort	Bit	Funktion
1-2	0-15	Pose: X-Koordinate als DWORD
3	0-15	Pose: Y-Koordinate als WORD
4	0-15	Pose: Z-Koordinate als WORD
5	0-15	Pose: A-Koordinate als WORD
6	0-15	Real 1 als WORD
7	0-7	Real 2 als BYTE
	8-11	Real 3 als NIBBLE
	12-15	Bool 1-4
8	0-3	Bool 5-8
	4-15	ungenutzt

26873346/DE – 07/2021

13 Diagnose

13.1 MOVIKIT® Feldbusmonitor

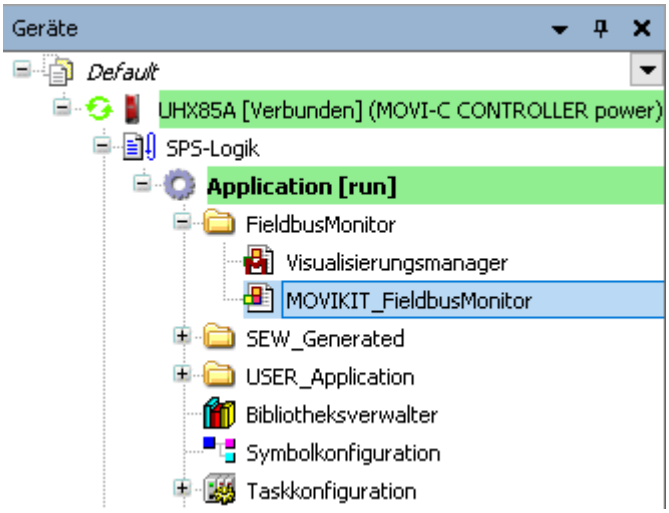
Der MOVIKIT® Feldbusmonitor ist ein Tool im IEC-Editor zum Beobachten und Steuern der Feldbus-Schnittstelle. Der MOVIKIT® Feldbusmonitor greift ausschließlich auf die Daten der Feldbus-Schnittstelle zu und stellt die zwischen übergeordneter Steuerung und dem Softwaremodul ausgetauschten Prozesseingangs- und -ausgangsdaten dar.

HINWEIS

Um den MOVIKIT® Feldbusmonitor zum Beobachten und Steuern der Feldbus-Schnittstelle zu nutzen, muss dieser importiert werden. Weitere Informationen dazu finden Sie in Kapitel "MOVIKIT® Feldbusmonitor importieren" (→ 🗎 106).

Führen Sie zum Öffnen des Tools folgende Schritte durch:

1. Öffnen Sie im MOVISUITE®-Projekt das Kontextmenü des MOVI-C® CONTROLLER und klicken Sie im Untermenü "Tools" auf den Menüeintrag [IEC-Editor].
   ⇨ Der IEC-Editor wird geöffnet.
2. Öffnen Sie das Menü [Online] und klicken Sie auf den Menüeintrag [Einloggen].
3. Doppelklicken Sie im Gerätebaum auf den Knoten "MOVIKIT_FieldbusMonitor". (Pfad: `Default > SPS-Logik > Application [run] > FieldbusMonitor`)



9007227578769547

⇨ Der MOVIKIT® Feldbusmonitor wird in einer neuen Registerkarte geöffnet.

⚠ WARNUNG

Unerwartetes Anlagenverhalten bei unterbrochener Kommunikation zwischen PC und MOVI-C® CONTROLLER durch das Weiterwirken der vorgegeben Sollwerte, bis die Verbindung zum IEC-Editor automatisch unterbrochen und der IEC-Editor ausgeloggt wird.

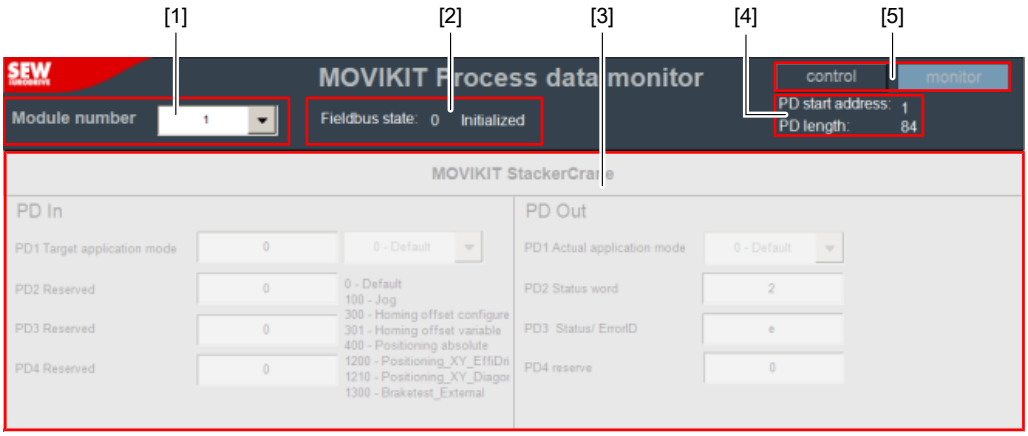
Tod, schwere Verletzungen oder Sachschaden

- Stellen Sie sicher, dass im Steuerbetrieb der Antrieb zu jeder Zeit über Not-Aus-Vorkehrungen gestoppt werden kann.

26873346/DE – 07/2021

13.1.1 Benutzeroberfläche

Die Benutzeroberfläche setzt sich aus folgenden Bereichen zusammen:



18014426835536651

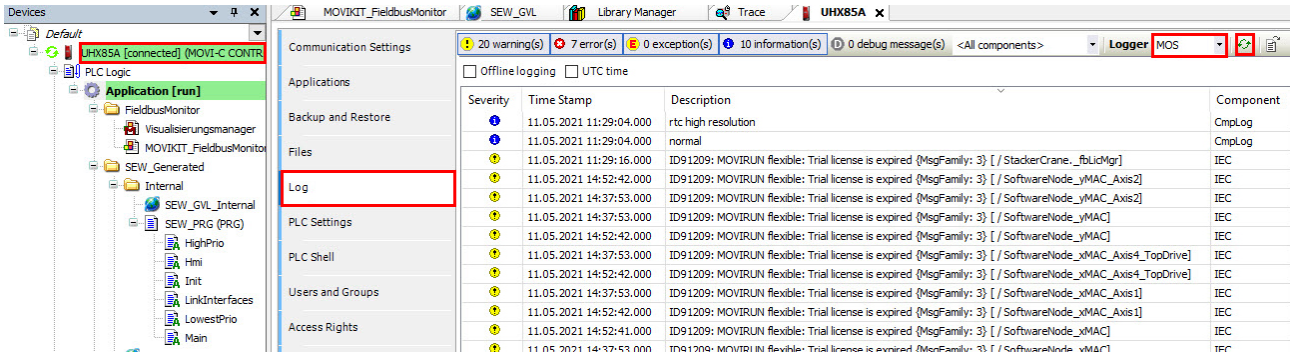
Nr.	Beschreibung
[1]	Nummer des Softwaremoduls, das beobachtet oder gesteuert werden soll. Wenn mehrere Softwaremodule vorhanden sind, richtet sich die Reihenfolge nach der in der Feldbus-Konfiguration des Softwaremoduls angegebenen Startadresse.
[2]	Statusinformationen des Feldbusses
[3]	Visualisierung der Prozessdaten und Bedienelemente zum Steuern der Bits
[4]	Startadresse und Prozessdatenlänge des unter [1] gewählten Softwaremoduls
[5]	Schaltflächen zum Wechseln zwischen "Beobachten" und "Steuern". Im Modus "Steuern" können die Funktionen des Softwaremoduls ohne Sollwerte der übergeordneten Steuerung getestet werden. Steuerbits und Prozessdatenwörter werden beim Drücken der Eingabetaste oder Klicken in ein anderes Eingabefeld direkt übernommen.

26873346/DE – 07/2021

13.2 Log-Funktion

Zur erweiterten Diagnose können alle Meldungen die durch Softwaremodule auf dem MOVI-C® CONTROLLER erzeugt werden über die Log-Funktion auch nach der Quittierung eingesehen werden. Gehen Sie zum Einsehen der Log-Funktion wie folgt vor:

1. Öffnen Sie über die MOVISUITE® den IEC-Editor.
2. Doppelklicken Sie im IEC-Editor im Gerätebaum auf das Objekt "Device (MOVI-C Controller)".
3. Öffnen Sie das Register "Log".
4. Wählen Sie als Logger den Eintrag "MOVIKIT" aus ("MOS" in älteren Versionen).
5. Klicken Sie auf die grüne Schaltfläche [Aktualisieren].
   ⇨ Sie können nun die Meldungen prüfen und ggf. die Ursache eines Problems beheben.



34972198027

14 Anwendungsbeispiele

14.1 Ansteuerung über das IEC-Programm

Die folgenden Anwendungsbeispiele zeigen, wie das Softwaremodul über das IEC-Anwenderprogramm angesteuert werden kann. Das Beispiel kann im IEC-Editor importiert werden. Öffnen Sie dazu im IEC-Editor das Menü [Tools] > [Skripting] > [Scripts] > [R] und klicken Sie auf den Menüeintrag [Robotics_Examples.py]. Anschließend muss das Programm z. B. wie folgt in der *USER_PRG.Main()* aufgerufen und dabei die Roboter-Instanz dem Beispielprogramm übergeben werden:

```
PRG_ExampleControlRobot_Recommended_Extended(
rInterface_Robot:=Interface_Robot);
//PRG_ExampleControlRobot_WithStateMachine(
Interface_Robot:=Interface_Robot);
```

Detaillierte Informationen zu den Signalen und der Ablaufsequenz finden Sie im Kapitel "Ansteuerung durch die Prozess-Steuerung" (→ 🗎 130).

14.1.1 Empfohlene Ansteuerung

```
PROGRAM PRG_ExampleControlRobot_Recommended_Extended
VAR_IN_OUT
    rInterface_Robot: SEW_MK_Robotics.Robot_UI;
END_VAR
VAR_INPUT
    usiRecipe : USINT:=1;
END_VAR
```

```
// Process start sequence
IF NOT SEW_PRG.xInitDone THEN
    RETURN;
ELSIF NOT rInterface_Robot.xControlActive THEN
    rInterface_Robot.xGetAccessControl:=TRUE;
ELSIF NOT (rInterface_Robot.Basic.OUT.xReady AND rInterface_Robot.Inverter.OUT.xReferenced) THEN
    rInterface_Robot.Basic.IN.eOperatingMode := SEW_MK_Robotics.SEW_IRobHPub.E_OperatingMode.Automatic;
    rInterface_Robot.Basic.IN.usiOverride := 100;
    rInterface_Robot.Basic.IN.xEnable_EmergencyStop := TRUE;
ELSIF NOT rInterface_Robot.Inverter.Out.xSetpointActive THEN
    rInterface_Robot.Prg.IN.xStart := FALSE; // Signal could still be TRUE from Line 25 and 36. Has to be set to FALSE to create a a rising edge there again
ELSE
    // Process sequence - Start with SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Initialized (Line 15 = line after next)
    CASE rInterface_Robot.Prg.OUT.eProgramState OF
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Initialized,
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Finished:
            CASE usiRecipe OF // Example for "recipe" - robot variables should be set before program execution, to assure that the correct values are respected
                1:  rInterface_Robot.Prg.IN.uiProgramNumber := 1;
                    rInterface_Robot.PrgVar.arRealValues[1] := 10.0;
                    rInterface_Robot.PrgVar.astPoseValues[1].arKinDim[2] := 0.0;
                2:  rInterface_Robot.Prg.IN.uiProgramNumber := 1;
                    rInterface_Robot.PrgVar.arRealValues[1] := 20.0;
                    rInterface_Robot.PrgVar.astPoseValues[1].arKinDim[2] := 10.0;
            END_CASE
            rInterface_Robot.Prg.IN.xStart := TRUE;
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BeingExecuted  (Line 27 = next line)
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BeingExecuted:
            CASE usiRecipe OF // Example for "recipe" - robot variables can be set during program execution to control program flow.
                // ATTENTION: other robot variables should be set before program execution, to assure that the correct values are respected
                1:  rInterface_Robot.PrgVar.axBoolValues[1] := TRUE;
                2:  rInterface_Robot.PrgVar.axBoolValues[1] := FALSE;
            END_CASE
            rInterface_Robot.Prg.IN.xStart := FALSE; // Signal is still TRUE from Line 25. Has to be set to FALSE to create a a rising edge in line 25 or 36 again.
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Finished (Line 16)
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BackToPathRequired:
            rInterface_Robot.Prg.IN.xStart := TRUE;
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BackToPathActive (Line 38 = next line)
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BackToPathActive:
            rInterface_Robot.Prg.IN.xStart := FALSE; // Signal is still TRUE from Line 36. Has to be set to FALSE to create a a rising edge in line 25 or 36 again.
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BeingExecuted (Line 27)
    END_CASE
END_IF
```

34950924171

14.1.2 Ansteuerung mittels State-Machine

```
PROGRAM PRG_ExampleControlRobot_Recommended_Extended
VAR_IN_OUT
    rInterface_Robot: SEW_MK_Robotics.Robot_UI;
END_VAR
VAR_INPUT
    usiRecipe : USINT:=1;
END_VAR
```

100 %

```
// Process start sequence
IF NOT SEW_PRG.xInitDone THEN
    RETURN;
ELSIF NOT rInterface_Robot.xControlActive THEN
    rInterface_Robot.xGetAccessControl:=TRUE;
ELSIF NOT (rInterface_Robot.Basic.OUT.xReady AND rInterface_Robot.Inverter.OUT.xReferenced) THEN
    rInterface_Robot.Basic.IN.eOperatingMode := SEW_MK_Robotics.SEW_IRobHPub.E_OperatingMode.Automatic;
    rInterface_Robot.Basic.IN.usiOverride := 100;
    rInterface_Robot.Basic.IN.xEnable_EmergencyStop := TRUE;
ELSIF NOT rInterface_Robot.Inverter.Out.xSetpointActive THEN
    rInterface_Robot.Prg.IN.xStart := FALSE; // Signal could still be TRUE from Line 25 and 36. Has to be set to FALSE to create a a rising edge there again
ELSE
    // Process sequence - Start with SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Initialized (Line 15 = line after next)
    CASE rInterface_Robot.Prg.OUT.eProgramState OF
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Initialized,
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Finished:
            CASE usiRecipe OF // Example for "recipe" - robot variables should be set before program execution, to assure that the correct values are respected
                1:  rInterface_Robot.Prg.IN.uiProgramNumber := 1;
                    rInterface_Robot.PrgVar.arRealValues[1] := 10.0;
                    rInterface_Robot.PrgVar.astPoseValues[1].arKinDim[2] := 0.0;
                2:  rInterface_Robot.Prg.IN.uiProgramNumber := 1;
                    rInterface_Robot.PrgVar.arRealValues[1] := 20.0;
                    rInterface_Robot.PrgVar.astPoseValues[1].arKinDim[2] := 10.0;
            END_CASE
            rInterface_Robot.Prg.IN.xStart := TRUE;
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BeingExecuted  (Line 27 = next line)
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BeingExecuted:
            CASE usiRecipe OF // Example for "recipe" - robot variables can be set during program execution to control program flow.
                // ATTENTION: other robot variables should be set before program execution, to assure that the correct values are respected
                1:  rInterface_Robot.PrgVar.axBoolValues[1] := TRUE;
                2:  rInterface_Robot.PrgVar.axBoolValues[1] := FALSE;
            END_CASE
            rInterface_Robot.Prg.IN.xStart := FALSE; // Signal is still TRUE from Line 25. Has to be set to FALSE to create a a rising edge in line 25 or 36 again.
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.Finished (Line 16)
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BackToPathRequired:
            rInterface_Robot.Prg.IN.xStart := TRUE;
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BackToPathActive (Line 38 = next line)
        SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BackToPathActive:
            rInterface_Robot.Prg.IN.xStart := FALSE; // Signal is still TRUE from Line 36. Has to be set to FALSE to create a a rising edge in line 25 or 36 again.
            // NEXT WILL BE: SEW_MK_Robotics.SEW_IRobHPub.E_ProgramState.BeingExecuted (Line 27)
    END_CASE
END_IF
```

34950926603

14.2 Statische USER-Koordinatensysteme

HINWEIS

Für statische Koordinatensysteme ist keine separate Lizenz erforderlich.

Gehen Sie zum Interpolieren von Bewegungen in einem statischen USER-Koordinatensystem wie folgt vor:

1. Legen Sie eine *StaticTransform*-Bausteininstanz und ein Array für die Transformation an. Verwenden Sie dafür z. B. den Deklarationsteil der im folgenden verwendeten *CallFunction*-Bausteininstanz.

```
fbStaticTransform:
SEW_MK_Robotics.SEW_RobUserCs.StaticTransform;
alrTransform:
ARRAY[1..6] OF LREAL;
```

2. Weisen Sie der Instanz die gewünschte Transformation zu. Verwenden Sie dafür z. B. einen CASE einer *CallFunction*. Siehe "IEC-Funktionsaufruf für das SRL-Programm" (→ 🗎 203).

```
CASE uiCallIndex OF
 1:alrTransform[1] := 100.0; //mm
   alrTransform[2] := 200.0; //mm
   alrTransform[3] := 300.0; //mm
   alrTransform[4] := 90.0; //Grad
   fbStaticTransform.SetTransform(alrTransform);
  Interface_Robot.LinkIUserCoordSys_Segment(fbStaticTransform);
   HMI_Robot.fbSimu3D.AddUserCS(fbStaticTransform, 1);
   CallF := TRUE;
```

3. Stellen Sie das gewünschte Koordinatensystem im SRL-Programm jeweils vor dem Bewegungsbefehl ein.

```
N10  MotionSet :=  1
N20  Blending Distance := 50  [mm]
N30  LIN  LeftUp
N40  CALL_FUNCTION  1
N50  Coordinate System := User
N60  LIN  RightUp
N70  LIN  RightDown
N80  WAIT MotionDone
N90  LIN  RightUp
N100 Coordinate System := Base
N110 LIN  LeftUp
N120 LIN  LeftDown
N130 END_PROG
```

33884952971

Anmerkungen:

HINWEIS

In der aktuellen Version der Software ist kein direkter Wechsel zwischen zwei verschiedenen USER-Koordinatensystemen möglich. Vor dem Wechsel in ein anderes USER-Koordinatensystem muss ein Bewegungsbefehl in BASE ausgeführt werden.

Das Koordinatensystem wird beim Herstellen des Grundzustands auf BASE eingestellt. Siehe "Herstellen des Grundzustands" (→ 🗎 38).

Eine neue Transformation wird nur genau beim Wechsel des Koordinatensystems von BASE nach USER übernommen. Wechselnde statische Transformationen können jeweils an die gleiche *StaticTransform*-Bausteininstanz angelegt werden. In diesem Fall müssen folgende Code-Zeilen nur einmal ausgeführt werden:

```
Interface_Robot.LinkIUserCoordSys_Segment(fbStaticTransform);
HMI_Robot.fbSimu3D.AddUserCS(fbStaticTransform, 1);
```

Alternativ können z. B. auch für verschiedene Transformationen verschiedene *StaticTransform*-Bausteininstanzen angelegt werden, die über die beiden Code-Zeilen jeweils bei ihrer Verwendung an den Roboter anzubinden sind.

14.3 Synchronisierte Bewegung mit einem Transportband

HINWEIS

Zum Durchführen dieses Anwendungsbeispiels ist das MOVIKIT® Robotics addon ConveyorTracking erforderlich.

Für die synchronisierte Bewegung mit einem Transportband (Conveyor Tracking), werden bewegbare USER-Koordinatensysteme verwendet. Gehen Sie zum Verwenden der Grundfunktionalität wie folgt vor:

1. Legen Sie eine *ConveyorOrRotaryTable*-Bausteininstanz und eine Konfiguration an. Verwenden Sie dafür z. B. den Deklarationsteil des Programms *User_PRG*.

```
fbConveyor :
SEW_MK_Robotics.SEW_RobUserCs.ConveyorOrRotaryTable;
stConfigConveyor :
SEW_MK_Robotics.SEW_RobUserCs.SEW_IRobUserCs.ST_Config;
```

2. Nehmen Sie die Konfiguration und die Anbindung an den Roboter vor. Verwenden Sie dafür z. B. die Aktion *User_PRG.Init*.

```
(* Typ *)
stConfigConveyor.eUserCoordSysType := SEW_MK_Robotics.
SEW_RobUserCs.SEW_IRobUserCs.E_UserCoordSysType.Linear;
(* Nullpunkt des Transportbands bzw. des USER-Koordinatensys-
tems *)
stConfigConveyor.alrOrigin[1] := 0.0; // mm
stConfigConveyor.alrOrigin[2] := 100.0;
(* 100 mm Verschiebung ggb. BASE in Y-Richtung *)
stConfigConveyor.alrOrigin[3] := 0.0; // mm
(* Richtung des Transportbands entlang der X-Achse *)
```

26873346/DE – 07/2021

```
stConfigConveyor.alrOrigin[4] := 30.0;
(* 30 Grad Verdrehung ggb. BASE um Z-Achse *)
stConfigConveyor.alrOrigin[5] := 0.0; // Grad
stConfigConveyor.alrOrigin[6] := 0.0;// Grad
(* Anbindung an den Roboter *)
fbConveyor.stConfig := stConfigConveyor;
Interface_Robot.LinkIUserCoordSys_Segment(fbConveyor);
HMI_Robot.fbSimu3D.AddUserCS(fbConveyor, 1);
xInitDone := (* xInitDone AND *) fbConveyor.Init();
```

3. Weisen Sie den Positionsverlauf des USER-Koordinatensystems stetig in der Task *HighPrio* zu. Verwenden Sie dafür z. B. die Task *User_PRG.HighPrio*. (Beispiel: USER-Koordinatensystem wird mittels Sinus bewegt).

```
lrSinArg := lrSinArg + 0.0001;
fbConveyor.lrSetpointPosition:=500.0(*mm*)*SIN(lrSinArg);
```

4. Stellen Sie das gewünschte Koordinatensystem im SRL-Programm jeweils vor dem Bewegungsbefehl ein.

```
N10  MotionSet := 1
N20  Blending Distance := 50 [mm]
N30  LIN LeftUp
N40  Coordinate System := User
N50  LIN RightUp
N60  LIN RightDown
N70  WAIT MotionDone
N80  LIN RightUp
N90  Coordinate System := Base
N100 LIN LeftUp
N110 LIN LeftDown
N120 END_PROG
```

33885095051

Anmerkungen:

Das Positionssignal muss bereits mindestens 5 HighPrio-Zyklen stetig der Variablen *fbConveyor.lrSetpointPosition* zugewiesen worden sein, bevor der Wechsel in das USER-Koordinatensystem erfolgt. Entsprechend muss das Signal solange stetig zugewiesen werden, wie die Interpolation in dem Koordinatensystem stattfindet.

Wenn das Positionssignal für den Eingang *fbConveyor.lrSetpointPosition* von einer MultiMotion-Achse verwendet wird, darf es nicht von den Ausgängen in *SEW_GVL.Interface_MultiMotionAxis* ausgelesen werden, die diese nur im Zyklus der *TaskMain* aktualisiert werden. Weisen Sie stattdessen z. B. die Ausgangssignale aus *SEW_GVL_Internal.MultiMotionAxis* zyklisch in der *TaskHighPrio* dem Eingang *fbConveyor.lrSetpointPosition* zu.

Wenn z. B. verrauschte Gebersignale verwendet werden sollen, sind diese applikativ zu glätten, bevor sie als stetiges Signal in der *TaskHighPrio* dem Eingang *fbConveyor.lrSetpointPosition* zugewiesen werden.

In der aktuellen Version der Software ist der direkte Wechsel zwischen zwei verschiedenen, bewegbaren linearen USER-Koordinatensystemen möglich, wenn beide die gleiche Ausrichtung haben, d. h. *alrOrigin[4..6]* beider Koordinatensysteme identisch sind. Hierzu ist eine weitere *ConveyorOrRotaryTable*-Bausteininstanz *fbConveyor2* erforderlich. Im SRL-Programm bleibt USER angewählt. Für den Wechsel wird vor dem gewünschten Bewegungsbefehl das neue USER-Koordinatensystem im entsprechenden CASE einer CallFunction an den Roboter angebunden. Siehe "IEC-Funktionsaufruf für das SRL-Programm" (→ 🗎 203).

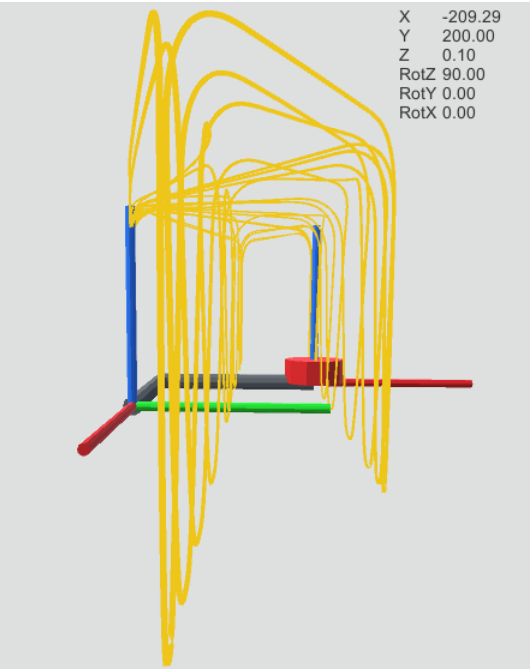
```
CASE uiCallIndex OF
  1:Interface_Robot.LinkIUserCoordSys_Segment(fbConveyor);
    HMI_Robot.fbSimu3D.AddUserCS(fbConveyor, 1);
    CallF := TRUE;
  2:Interface_Robot.LinkIUserCoordSys_Segment(fbConveyor2);
    HMI_Robot.fbSimu3D.AddUserCS(fbConveyor2, 1);
    CallF := TRUE;
```

Das Positionssignal des vorhergehenden USER-Koordinatensystems muss stetig zugewiesen werden, bis die Bewegung im neuen USER-Koordinatensystem erfolgt. Das ist der Fall, sobald sich der Wert in der Variablen *SEW_GVL.Interface_MyRobot.Prg.OUT.uiNumberOfUserCoordSysChanges* ändert.

Das Koordinatensystem wird beim Herstellen des Grundzustands auf BASE eingestellt. Siehe "Herstellen des Grundzustands" (→ 🗎 38).

In der aktuellen Version der Software sind die Verwaltung der auf dem Transportband bewegten Objekte sowie die Vermeidung eines Überlaufs eines Gebersignals applikativ zu realisieren.

Eine komplette, offene Automatisierungslösung für Conveyor Tracking inkl. Kameraanbindung und Lastaufteilung auf mehrere Roboter ist im "SEW Automation Framework" enthalten. Für weitere Informationen kontaktieren Sie SEW-EURODRIVE.



33885103243

26873346/DE – 07/2021

14.4 Synchronisierte Bewegung mit einem Drehtisch

HINWEIS

Zum Durchführen dieses Anwendungsbeispiels ist das MOVIKIT® Robotics addon ConveyorTracking erforderlich.

Für die synchronisierte Bewegung mit einem Drehtisch (Rotary Table Tracking), werden wie beim "Conveyor Tracking" bewegbare USER-Koordinatensysteme verwendet.

Das Verwenden der Funktion erfolgt analog zu der im Kapitel "Synchronisierte Bewegung mit einem Transportband" (→ 🗎 224) beschriebenen Vorgehensweise. Beachten Sie dabei jedoch folgende Unterschiede:

- In der Konfiguration muss folgender Typ ausgewählt werden:

```
stConfigRotaryTable.eUserCoordSysType :=
SEW_MK_Robotics.SEW_RobUserCs.SEW_IRobUserCs.
   E_UserCoordSysType.Rotary;
```

- Das Positionssignal *lrSetpointPosition* ist die stetige Folge der Drehwinkel um die Z-Achse des USER-Koordinatensystems des Drehtisches in Grad.
- Der direkte Wechsel vom/zum RotaryTable-USER-Koordinatensystem ist nur von / zu BASE möglich.

Daraus ergibt sich folgender Beispiel-Code im IEC-Programm:

1. Legen Sie eine *ConveyorOrRotaryTable*-Bausteininstanz und eine Konfigurationsvariable an. Verwenden Sie dafür z. B. den Deklarationsteil des Programms *User_PRG*.

```
fbRotaryTable :
SEW_MK_Robotics.SEW_RobUserCs.ConveyorOrRotaryTable;
stConfigRotaryTable :
SEW_MK_Robotics.SEW_RobUserCs.SEW_IRobUserCs.ST_Config;
```

2. Nehmen Sie die Konfiguration und die Anbindung an den Roboter vor. Verwenden Sie dafür z. B. die Aktion *User_PRG.Init*.

```
(* Typ *)
stConfigRotaryTable.eUserCoordSysType := SEW_MK_Robotics.
SEW_RobUserCs.SEW_IRobUserCs.E_UserCoordSysType.Rotary;
(* Nullpunkt des Drehtischs bzw. des USER-Koordinatensystems *)
stConfigRotaryTable.alrOrigin[1] := 200.0;
(* 200mm Verschiebung ggb. BASE in X-Richtung *)
stConfigRotaryTable.alrOrigin[2] := 0.0; // mm
stConfigRotaryTable.alrOrigin[3] := 0.0; // mm
(* Drehung um Z-Achse des Drehtischs *)
stConfigRotaryTable.alrOrigin[4] := 0.0; // Grad
stConfigRotaryTable.alrOrigin[5] := 0.0; // Grad
stConfigRotaryTable.alrOrigin[6] := 0.0; // Grad
(* Anbindung an den Roboter *)
fbRotaryTable.stConfig := stConfigRotaryTable;
Interface_Robot.LinkIUserCoordSys_Segment(fbRotaryTable);
HMI_Robot.fbSimu3D.AddUserCS(fbRotaryTable, 1);
```

26873346/DE – 07/2021

```
xInitDone := (* xInitDone AND *) fbRotaryTable.Init();
```

3. Weisen Sie den Positionsverlauf des USER-Koordinatensystems stetig in der Task *HighPrio* zu. Verwenden Sie dafür z. B. die Task *User_PRG.HighPrio*. (Beispiel: USER-Koordinatensystem wird mittels Sinus bewegt).

```
lrSinArg := lrSinArg + 0.0001;
fbRotaryTable.lrSetpointPosition :=
120.0(* Grad *) * SIN(lrSinArg);
```

Das SRL-Programm und die Anmerkungen hinsichtlich des Wechselns des Koordinatensystems sind die gleichen wie im Kapitel "Synchronisierte Bewegung mit einem Transportband" (→ 🗎 224).

33885258123

26873346/DE – 07/2021

14.5 Besonderheiten bei der Steuerung eines Drehtischs durch den Roboter

Ein Drehtisch kann optional durch den Roboter gesteuert werden, sodass der jeweils im USER-Koordinatensystem anzufahrende Bahnpunkt in der ZX-Ebene des BASE-Koordinatensystems bei Y = 0 liegt. So lassen sich die Freiheitsgrade des Roboters um die Bewegung in Y-Richtung erweitern. Siehe dazu auch Kapitel "Steuerung einer Transporteinrichtung durch den Roboter" (→ 🗎 94). Im SRL-Programm muss hierzu das Koordinatensystem "USER_ControlledByRobot" eingestellt werden.

Der Beispiel IEC-Code im Kapitel "Synchronisierte Bewegung mit einem Drehtisch" (→ 🗎 227) ist gültig mit einer Ausnahme (im Punkt 3). Der Positionsverlauf wird nicht dem *fbRotaryTable* übergeben. Stattdessen werden die Setpoints aus dem *fbRotaryTable* ausgelesen und für die Ansteuerung einer MultiMotion-Tracking-Achse verwendet.

```
(* Setpoints to MuMo tracking axis, [-180°, 180°[ *)
lrSetpointForMuMoTrackingAxis :=
fbRotaryTable.lrSetpointPosition;
```

Für eine Vorpositionierung der Drehtischs kann die Methode *AdjustToPoseWithProfile()* des Funktionsbausteins *ConveyorOrRotaryTable* verwendet werden. Während der Positionierung darf im SRL-Programm für diesen Drehtisch **nicht** das Koordinatensystem "USER_ControlledByRobot" eingestellt sein. Der Roboter muss sich während der Positionierung z. B. in "BASE" befinden. Führen Sie den folgenden Beispielcode z. B. in der Task *User_PRG.HighPrio* aus.

Die Positionierung des Drehtischs erfolgt derart, dass der Punkt mit den Koordinaten X = *alrPose[1]*, Y = *alrPose[2]* im USER-Koordinatensystem nach erfolgter Positionierung in BASE-Koordinaten den Koordinatenwert Y = 0 hat und zwischen Roboterbasis und dem Mittelpunkt des Drehtischs liegt. Die Positionierung erfolgt auf dem kürzesten Weg. Die Positionierung startet bei steigender Flanke von *xStart*. Die Eingangsvariable muss bis zum Ende der Positionierung "TRUE" bleiben und muss danach mit "FALSE" aufgerufen werden, sobald wieder *USER_ControlledByRobot* eingestellt wird. Der Baustein gibt "TRUE" zurück, wenn die Positionierung komplett ausgeführt ist.

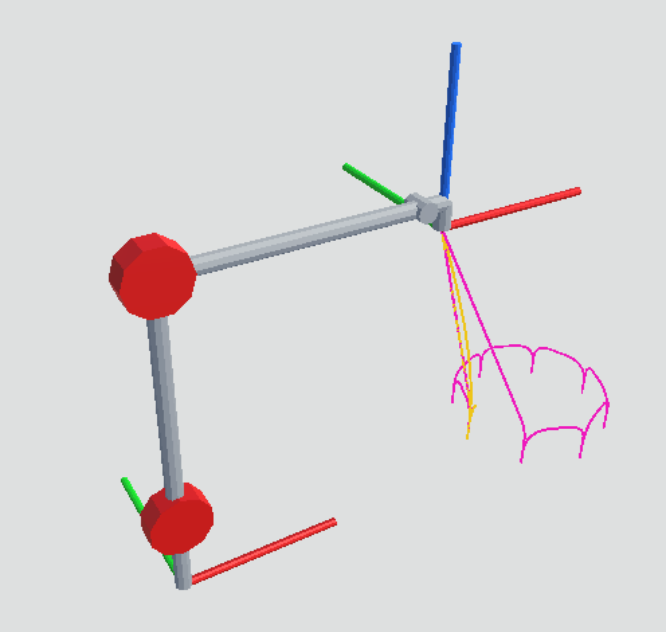
In dem Beispiel werden Roboter-Posevariablen für *alrPose* verwendet sowie Roboter-Boolvariablen für *xStart* und den Rückgabewert. So ist ein direkter Handshake mit dem SRL-Programm möglich.

```
alrPose[1] :=
SEW_GVL.Interface_Robot.PrgVar.astPoseValues[21].arKinDim[1];
alrPose[2] :=
SEW_GVL.Interface_Robot.PrgVar.astPoseValues[21].arKinDim[2];
SEW_GVL.Interface_Robot.PrgVar.axBoolValues[2] :=
fbRotaryTable.AdjustToPoseWithProfile(
   xFeedEnable := TRUE,
   xRapidStop := xRapidStop_RotaryTable,
   (* Schnellstopp, wenn TRUE *)
   lrVelocity := 100, // [°/s], > 0
   lrAcceleration := 100, // [°/s²], > 0
   lrJerk := 1000, // [°/s³], > 0
   lrRapidDeceleration := 1000, // >= lrAcceleration
   lrRapidJerk := 10000, // >= lrJerk
   xStart := SEW_GVL.Interface_Robot.PrgVar.axBoolValues[1],
   alrPose := alrPose (* ARRAY[1..6] OF LREAL *));
```

26873346/DE – 07/2021

Bei der Erstellung der Applikation muss außerdem auf folgende Eigenschaften geachtet werden:

- Ein Wechsel von/zu dem USER-Koordinatensystem eines Drehtischs ist nur über "BASE" möglich.
- Der Drehtisch wird automatisch immer so ausgerichtet, dass sich die anzufahrende Position auf der dem Roboter zugewandten Seite des Drehtischs befindet.
- Das Durchfahren des Mittelpunkts des Drehtischs verursacht einen Fehler, da dies im Allgemeinen zu sehr schnellen Bewegungen des Drehtischs führen würde.
- Für den Wechsel in das oder aus dem USER-Koordinatensystem muss sich der Drehtisch im Stillstand befinden, wenn der Roboter in Y-Richtung keine Ausgleichbewegung ausführen kann.
- Die Funktion "BackToPath" funktioniert mit der aktuellen Version der Software nicht, da der Drehtisch nicht automatisch in die passende Stellung zurückpositioniert wird, um die Bahn stetig fortsetzen zu können.



34927673483

26873346/DE – 07/2021

14.6 Nicht sicherheitsbewertete Freigabesteuerung

Zum Implementieren einer nicht sicherheitsbewerteten Freigabesteuerung, weisen Sie die geforderte Freigabeart den im Folgenden aufgeführten Variablen in *SEW_GVL_internal* zu.

Anwendungsbeispiele:

- Verschaltung des Signals einer Sicherheitssteuerung an den MOVI-C® CONTROLLER zum bahntreuen Anhalten des Roboters vor Aktivierung von STO, durch die Sicherheitssteuerung am MOVIDRIVE®-Umrichter der Generation C.
- Applikative Hardware-Endschalter bei einem ROLLER GANTRY
- Eine applikative Arbeitsraum- oder Kollisionsüberwachung

Der Vorteil dieser Variablen gegenüber den Freigabesignalen in den Steuerquellen (UserInterface *SEW_GVL*, RobotMonitor, Feldbusschnittstelle) ist, dass sie unabhänigig von der zugriffsberechtigten Steuerquelle immer wirken. Wenn also mindestens eines der drei folgenden Signale nicht "TRUE" ist, wird unabhängig von den Freigabesignalen in den Steuerquellen ein Halt ausgelöst.

```
MyRobot.xEnable_EmergencyStopAxis_ConnectToSafetyController
   _NOTSAFE := xApplicativeEnable_EmergencyStopAxis;
MyRobot.xEnable_EmergencyStop_ConnectToSafetyController
   _NOTSAFE := xApplicativeEnable_EmergencyStop;
MyRobot.xEnable_ApplicationStop_ConnectToSafetyController
   _NOTSAFE := xApplicativeEnable_ApplicationStop;
```

Die höchste Priorität hat das Rücksetzen auf "FALSE" des Signals *xEnable_EmergencyStopAxis_ConnectToSafetyController_NOTSAFE*, gefolgt von *xEnable_EmergencyStop_ConnectToSafetyController_NOTSAFE*.

Die Wegnahme der Freigabe in den Steuerquellen wirkt zusätzlich.

PRG_User_CollisionProtection

```
1   PROGRAM PRG_User_CollisionProtection

1   // Forbidden area is at X<=-1000 AND Y<=-1000
2       IF Interface_SCARA_Lab.Basic.OUT.SetpointPose.alrBase[1]<=-1000
3       AND Interface_SCARA_Lab.Basic.OUT.SetpointPose.alrBase[2]<=-1000 THEN
4           // Emergency stop
5               SCARA_Lab.xEnable_EmergencyStopAxis_ConnectToSafetyController_NOTSAFE   :=TRUE;
6               SCARA_Lab.xEnable_EmergencyStop_ConnectToSafetyController_NOTSAFE       :=FALSE
7               SCARA_Lab.xEnable_ApplicationStop_ConnectToSafetyController_NOTSAFE     :=TRUE;
8       ELSE
9           // Enable
10              SCARA_Lab.xEnable_EmergencyStopAxis_ConnectToSafetyController_NOTSAFE   :=TRUE;
11              SCARA_Lab.xEnable_EmergencyStop_ConnectToSafetyController_NOTSAFE       :=TRUE;
12              SCARA_Lab.xEnable_ApplicationStop_ConnectToSafetyController_NOTSAFE     :=TRUE;
13      END_IF
```

9007222992193547

14.7 Externen Anwahlschalter für die Betriebsarten verwenden

Wenn die Betriebsart über einen externen Anwahlschalter vorgegeben werden soll (z. B. über den Schlüsselschalter eines Bedienpanels) und nicht über eine Steuerquelle des Softwaremoduls (UserInterface, RobotMonitor, Feldbusschnittstelle), ist dies folgendermaßen möglich:

1. Legen Sie eine Interface-Variable an, über die die Betriebsart des Roboters direkt vorgegeben werden kann. Dies kann im Anwenderprogramm (User_PRG) im IEC-Programm erfolgen:
   - ⇨ `itfOperatingMode:SEW_MK_Robotics.SEW_IRobHPub.IOperating-Mode_ConnectToSafetyController_NOTSAFE:= MyRobot.fbOperatingModeHandler;`
   - ⇨ "MyRobot" entspricht dem in MOVISUITE® vergebenen Namen des Roboterknotens.
2. Weisen Sie der Variable die von außen vorgegebenen Betriebsart zu:
   - ⇨ `itfOperatingMode.eOperatingMode_ConnectToSafetyController_NOTSAFE:= SEW_MK_Robotics.SEW_IRobHPub.E_OperatingMode.Automatic;`
   - ⇨ `itfOperatingMode.eOperatingMode_ConnectToSafetyController_NOTSAFE:= SEW_MK_Robotics.SEW_IRobHPub.E_OperatingMode.Manual_WithHighSpeed;`

HINWEIS

Sobald die Betriebsart einmalig über diesen Mechanismus ausgewählt wurde, wird die Betriebsart, die in den anderen Steuerquellen (UserInterface, RobotMonitor, Feldbusschnittstelle) angewählt ist, ignoriert und hat keine Wirkung mehr.

26873346/DE – 07/2021

14.8 Umschaltung auf Steuerung der Einzelachsen

Für das Nutzen einiger Einzelachsfunktionalitäten muss die Steuerung der Achsen vom Roboter auf die Einzelachsen umgeschaltet werden. Der Roboter muss sich für das Umschalten in einem sicheren Zustand befinden.

Nehmen Sie vor dem Umschalten der Steuerung die Freigabe zurück. Wenn der Roboter freigegeben ist und der Zugriff auf eine untergeordnete Einzelachse wird angefordert, wird der Roboter in den Fehlerzustand versetzt. Der Roboter bremst mit den verbliebenen Achsen mit einem achsweisen Not-Halt ab.

14.8.1 Umschaltung von Roboter auf Einzelachse

1. Nehmen Sie die Freigabe am Roboter zurück (*xEnable_EmergencyStop* = "FALSE").
2. Prüfen Sie, dass der Sollwert nicht mehr aktiv ist (*xSetpointActive*="FALSE")
3. Fordern Sie den Zugriff auf die untergeordnete Achse an (*xGetAccessControl*="TRUE").
   ⇨ Der Zugriff auf die Einzelachse wird gewährt (*xControlActive*="TRUE").

⇨ Die Einzelachse kann gesteuert werden.

14.8.2 Umschaltung von Einzelachse auf Roboter

1. Geben Sie die Zugriffsberechtigung auf die untergeordnete Achse zurück (*xGetAccessControl*="FALSE").
2. Der Zugriff auf die Einzelachse wird zurückgegeben (*xControlActive*="FALSE").

⇨ Roboter kann gesteuert werden.

14.9 Freie Funktionstasten des Bedienpanels verwenden

Das Bedienpanel verfügt über Funktionstaster, die vom Anwender beliebig verwendet werden können z. B. um digitale Ausgänge zu schreiben. Die Funktionstaster werden nur an den MOVI-C® CONTROLLER übertragen, wenn der RobotMonitor auf dem Bedienpanel ausgeführt wird, mit der Steuerung verbunden ist und Zugriff auf den Roboter hat.

Die Zustände der Funktionstaster können in folgenden Variablen ausgelesen werden:

- `HMI_MyRobot.fbMonitor.InSignals.xUserFunctionKey1`
- `HMI_MyRobot.fbMonitor.InSignals.xUserFunctionKey2`

"MyRobot" entspricht dem vom Anwender in MOVISUITE® vergebenen Namen des Roboterknotens.

Ist die BOOL-Variable TRUE, bedeutet dies, dass der Taster gerade gedrückt ist. Ist die BOOL-Variable FALSE, bedeutet dies, dass der Taster gerade nicht gedrückt ist.

26873346/DE – 07/2021

14.10 Kombination des RobotMonitors mit einer Visualisierung

Der RobotMonitor kann einerseits eigenständig auf einem Gerät mit Windows-Betriebssystem ausgeführt und andererseits in Kombination mit einer Maschinen- oder Anlagenvisualisierung verwendet werden. Dazu kann der RobotMonitor z. B. durch einen Tastendruck in der Visualisierung gestartet werden. Falls der RobotMonitor bereits gestartet ist, wird er bei erneutem Tastendruck nicht noch einmal gestartet, sondern in den Vordergrund gerückt. Damit der RobotMonitor nicht die komplette Visualisierung überdeckt und man wieder in die Visualisierung zurückwechseln kann, kann man die Größe des RobotMonitors (in px) beim Starten über folgende Argumente angeben:

-WindowWidth 600 -WindowHeight 920

Wenn der RobotMonitor nicht im Vordergrund ist, verwendet dieser nicht die Hardware-Taster des Bedienpanels und diese können von der Visualisierung verwendet werden.

In folgendem Beispiel wurde der RobotMonitor mit einer PackML-Visualisierung kombiniert. Im unteren Bereich sind die Steuerelemente der Visualisierung zu sehen. Darüber liegt der RobotMonitor. Durch Drücken eines Steuerelements in der Visualisierung wird der RobotMonitor automatisch in den Hintergrund gerückt und die Visualisierung wird komplett angezeigt. Über einen Button in der Visualisierung kann der RobotMonitor wieder gestartet bzw. in den Vordergrund gerückt werden.

32292574603

14.11 Bewegungsparameter einstellen

14.11.1 Geschwindigkeit

Faustformel:

- Strecke/Sollzeitbedarf (Von diesem Wert aus nach oben tasten)

Korrekter Wert - Minimum aus ...

- Maximale Geschwindigkeit, die die Mechanik, das Werkzeug und das Produkt erfahren darf.
- Geschwindigkeit am Werkzeug ermitteln, die mit den maximalen Motordrehzahlen möglich ist (abhängig von der Getriebe-Übersetzung, der Stellung der Kinematik und damit den Hebelarmen)

14.11.2 Beschleunigung

Faustformel:

- Geschwindigkeit × 10 (d. h. die Geschwindigkeit soll in 100 ms aufgebaut werden)

Korrekter Wert - Minimum aus ...

- Maximale Beschleunigung, die die Mechanik, das Werkzeug und das Produkt erfahren darf.
- Mittels der maximalen Motormomente und Getriebe-Eintriebsmomente ermitteln, welche Beschleunigung am Werkzeug möglich ist (Dies ist abhängig von der Übersetzung, der Stellung der Kinematik und damit den Hebelarmen; sowie den Trägheiten).

14.11.3 Ruck

Faustformel:

- Beschleunigung × 10 bzw. bei steifen Mechaniken Geschwindigkeit × 100 (d. h. die Beschleunigung soll in 100 ms bzw. in 10 ms aufgebaut werden)

Korrekter Wert - Minimum aus ...

- Maximaler Ruck, den die Mechanik, das Werkzeug und das Produkt erfahren darf.
- Maximaler Ruck, mit dem das Werkzeug und die Robotermechanik nicht zum Schwingen angeregt wird.

26873346/DE – 07/2021

14.12 Optimierung der Taktzeit

Wenn die gewünschte Taktzeit nicht erreicht wird, führen Sie folgende Schritte durch:

1. Stellen Sie sicher, dass identische Posen nicht doppelt angefahren werden.
2. Erhöhen Sie den Ruck oder die Beschleunigung in der Konfiguration. Führen Sie dies auch für vermeintlich in der aktuellen Betriebsart nicht genutzte Parameter aus, da sich Ihre Einstellung durch die Synchronisierung auf die resultierenden Bewegungsparameter auswirken kann. Siehe auch "Bewegungsparametersätze" (→ 🗎 28) und "Bewegungsparameter einstellen" (→ 🗎 235).
3. Prüfen Sie mittels des Signals *SEW_GVL_Internal.INSTANZNAME.fbMotionControl.Out.stProg.lrLimitedSpeedDueToCentrifugalAcc*, ob die Bahngeschwindigkeit aufgrund der Zentrifugalkraftbeschleunigung begrenzt ist (0 = nicht limitiert).
4. Vergrößern Sie die Überschleifdistanz, wenn es die Störkonturen zulassen. Siehe Kapitel "Überschleifen" (→ 🗎 26).
   - ⇨ Die Taktzeit wird erreicht.
   - ⇨ Wenn die Taktzeit immer noch nicht erreicht wird, wenden Sie sich an den Service von SEW-EURODRIVE.

26873346/DE – 07/2021

15 Fehlermanagement

15.1 Problembehebung

15.1.1 Roboter führt keine Bewegung aus

Problem

Der Roboter führt keine Bewegung aus.

Abhilfe

- Allgemeine Ursachen prüfen und beseitigen:
  - Kein Zugriff angefordert
  - Keine Zugriffsberechtigung erhalten
  - Falsche Roboterinstanz angewählt
  - Keine Freigabe gesetzt
  - *Override* auf den Wert "0" gesetzt
  - Es liegt ein Fehler vor
  - Sollwerte nicht aktiv
  - Umrichter nicht bestromt
  - Umrichter nicht verbunden
- Wenn der Roboter im Tippbetrieb ist, folgende Ursachen prüfen und beseitigen:
  - Betriebsart ist nicht "Manuell mit hoher Geschwindigkeit"
  - Kein Tippsignal aktiv
- Wenn der Roboter im Programmbetrieb ist, folgende Ursachen prüfen und beseitigen:
  - Programmpause aktiviert
  - Programmstopp aktiviert
  - Keine steigende Flanke von Programmstart erkannt
  - In der Betriebsart "Manuell mit hoher Geschwindigkeit" Programmstart nicht aktiviert
  - Zielposition(en) identisch mit aktueller Position
- Weitere Ursachen prüfen und beseitigen:
  - Vorgegebene Geschwindigkeit zu klein
  - Vorgegebene Beschleunigung zu klein
  - Vorgegebener Ruck zu klein

15.1.2 Antriebe brummen

Problem

Die Antriebe brummen während der Bewegung.

Abhilfe

Überprüfen Sie, ob die im MOVIKIT® MultiMotion vorgeschlagenen Nachkommastellen übernommen wurden.

26873346/DE – 07/2021

15.1.3 Kommunikation RobotMonitor/MOVI-C® CONTROLLER nicht möglich

Problem

Die Kommunikation zwischen dem RobotMonitor und dem MOVI-C® CONTROLLER ist nicht möglich. Die Meldung "NO Communication" wird angezeigt.

Abhilfe

1. Prüfen Sie, ob "Start Communication" gedrückt wurde.
2. Prüfen Sie, ob die eingestellte IP des MOVI-C® CONTROLLER korrekt ist. Die eingestellte IP können Sie der MOVISUITE® oder dem IEC-Editor entnehmen.
3. Öffnen Sie im IEC Editor die globale Variablenliste *SEW_GVL_Internal*.
4. Navigieren Sie zu der Variable *HMI_Instanzname*.
5. Prüfen Sie diese und die unterlagerten Bausteine *fbSimu3D*, *fbMonitor* und *fbProgramEditor* auf Fehlernachrichten.
6. Doppelklicken Sie im IEC-Editor im Gerätebaum auf das Objekt *Device*.
7. Öffnen Sie auf das Register "Log".
8. Wählen Sie als Logger den Eintrag "MOS" aus.
9. Klicken Sie auf die Schaltfläche [Aktualisieren].
10. Prüfen Sie die Nachrichten auf Fehler aus dem Komminikationsbaustein des Softwaremoduls und beheben Sie die Ursache.
    ⇨ Wenn der RobotMonitor immer noch nicht verbunden ist, wenden Sie sich an den Service von SEW-EURODRIVE.

15.1.4 Kommunikation RobotMonitor/MOVI-C® CONTROLLER bricht immer wieder ab (Timeout)

Problem

Wenn der RobotMonitor auf Geräten mit schwacher Leistung ausgeführt wird, z. B. ein alter Engineering-PC oder ein mobiles Panel, dessen Fokus auf der Robustheit und nicht auf Performance liegt, kann es zu Timeouts des RobotMonitors kommen. Dies äußert sich darin, dass während der RobotMonitor den Zugriff auf den Roboter hat, der Roboter einen Notstopp durchführt und die Bremsen der Roboterachsen einfallen.

Abhilfe

Um den RobotMonitor auch auf Geräten mit schwacher Leistung nutzen zu können, kann die Timeoutzeit erhöht werden. Über die INPUT-Variable *tTimeOut* des HMI-Bausteins kann die Zeit vorgegeben werden, nach welcher ein Timeout erkannt und der Roboter gestoppt wird. Der Standardwert für die Timeout-Zeit ist 1.5 s. Auf den beschriebenen Geräten hat sich eine Timeout-Zeit von 4 s als praktikabel erwiesen.

Die Zuweisung der Timeout-Zeit kann wie folgt in der Aktion *User_PRG.Init* erfolgen:

```
HMI_SoftwareNode.tTimeOut := T#4S;
```

Wenn die Kommunikation auch nach dem Erhöhen der Timeout-Zeit nicht stabil läuft, wenden Sie sich an den Service von SEW-EURODRIVE.

26873346/DE – 07/2021

15.1.5 SRL-Programm wird nicht gefunden

Problem

Das SRL-Programm wird nicht gefunden. Die Fehlermeldung "File not Found: srlfiles/.../Prog1.crc. Please click on "Save" in robot monitor and restart PLC!" wird angezeigt.

Abhilfe

✓ Das Signal "GetControl" wurde aktiviert und die Rückmeldung "Control Active" wird angezeigt.

1. Klicken Sie im RobotMonitor auf den Disketten-Button [Save programs and variables on memory card].
2. Setzen Sie den Fehler mit dem Signal "Reset" zurück.

26873346/DE – 07/2021

15.2 Fehlercodes

15.2.1 Robotics Fehler

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32000	0x7d00	InterfaceNotValid	Robot_ltfs
32001	0x7d01	NoValidRobotKinematicModel_NrOfJoints	Robot_ltfs
32002	0x7d02	KinematicModelNotLinked	Robot_ltfs
32003	0x7d03	ActualRobotPoseInvalid	Robot_ltfs
32004	0x7d04	InternalInitializingStateWrong	Robot_ltfs
32005	0x7d05	ReplacementOfKinematicModelNotSupported	Robot_ltfs
32064	0x7d40	InterfaceNotValid	RobotHandler_ltfs
32065	0x7d41	EnableOutOfBoundaries_Safety	RobotHandler_ltfs
32066	0x7d42	InternalWrongEnableMode	RobotHandler_ltfs
32067	0x7d43	ProgramAndJogModeAtSameTime	RobotHandler_ltfs
32068	0x7d44	Just_SeqModeAuto_in_OpModeAuto	RobotHandler_ltfs
32069	0x7d45	ProgramChangeNotPossible	RobotHandler_ltfs
32070	0x7d46	InternalProgramInitNotPossible	RobotHandler_ltfs
32071	0x7d47	ProgramNumberChangedWhileExecution	RobotHandler_ltfs
32072	0x7d48	Inverter_NotConnected	RobotHandler_ltfs
32073	0x7d49	Inverter_SafeStop	RobotHandler_ltfs
32074	0x7d4a	Inverter_NotReady	RobotHandler_ltfs
32075	0x7d4b	Inverter_NotReady_TooLong	RobotHandler_ltfs
32076	0x7d4c	Inverter_NotReferenced	RobotHandler_ltfs
32077	0x7d4d	JogOnlyInManuelReducedSpeed	RobotHandler_ltfs
32078	0x7d4e	AxisGroupInverterModeWrong	RobotHandler_ltfs
32079	0x7d4f	AxisGroupInverterModeWrongWhileInterp	RobotHandler_ltfs
32080	0x7d50	AxisGroupInverterModeWrongTooLong	RobotHandler_ltfs
32081	0x7d51	InternalRefToMotionSetInvalid	RobotHandler_ltfs
32082	0x7d52	InternalResetProgOfMotionControlNotPossible	RobotHandler_ltfs
32083	0x7d53	ModuleNotNeeded	RobotHandler_ltfs
32084	0x7d54	ManualWithReducedSpeedNotSupported	RobotHandler_ltfs
32085	0x7d55	InternalSetUpNotPossible	RobotHandler_ltfs
32086	0x7d56	ChangeOfSimulationNotAllowed	RobotHandler_ltfs
32087	0x7d57	NoAxesLinkedAndNoSimulation	RobotHandler_ltfs
32088	0x7d58	ProgramInterpreterStateNotKnown	RobotHandler_ltfs
32089	0x7d59	ControlModeNotKnown	RobotHandler_ltfs
32090	0x7d5a	NoAccessToSubordinatedAxes	RobotHandler_ltfs
32091	0x7d5b	AxisGroupInverterModeWrongWhileHoming	RobotHandler_ltfs
32092	0x7d5c	AxisGroupInverterModeWrongTooLongWhileHoming	RobotHandler_ltfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32093	0x7d5d	Inverter_LostReference	RobotHandler_Itfs
32094	0x7d5e	HomingOnlyInManuelReducedSpeed	RobotHandler_Itfs
32096	0x7d60	RobotHMI_LinkRobotError_TargetIndexOutOfRange	RobotHMI_Itfs
32097	0x7d61	RobotHMI_LinkRobotError_RobotInterfaceInvalid	RobotHMI_Itfs
32098	0x7d62	RobotHMI_SwitchRobotError_AtLeastOneInterfaceInvalid	RobotHMI_Itfs
32099	0x7d63	RobotHMI_SwitchRobotError_InvalidRobotInterface	RobotHMI_Itfs
32100	0x7d64	RobotHMI_SwitchRobotError_RequestedIndexOutOfRange	RobotHMI_Itfs
32101	0x7d65	RobotHMI_InvalidComState	RobotHMI_Itfs
32102	0x7d66	RobotHMI_ErrorInGuiCom	RobotHMI_Itfs
32103	0x7d67	RobotHMI_ReceiveToolVersionTimeOut	RobotHMI_Itfs
32104	0x7d68	RobotHMI_ConnectSubModulesTimeOut	RobotHMI_Itfs
32128	0x7d80	InvalidArrayIndex	RobotLanguage_Itfs
32129	0x7d81	InvalidInterface	RobotLanguage_Itfs
32130	0x7d82	InvalidReference	RobotLanguage_Itfs
32131	0x7d83	InvalidOperation	RobotLanguage_Itfs
32132	0x7d84	InvalidVarType	RobotLanguage_Itfs
32133	0x7d85	InvalidValueType	RobotLanguage_Itfs
32134	0x7d86	InvalidVarIndex	RobotLanguage_Itfs
32135	0x7d87	InvalidModuleType	RobotLanguage_Itfs
32136	0x7d88	InvalidBlock	RobotLanguage_Itfs
32137	0x7d89	InvalidProgramNumber	RobotLanguage_Itfs
32138	0x7d8a	NotImplementedInThisVersion	RobotLanguage_Itfs
32139	0x7d8b	ExecutionNotPossible	RobotLanguage_Itfs
32140	0x7d8c	InvalidOperand	RobotLanguage_Itfs
32141	0x7d8d	ReadDataError	RobotLanguage_Itfs
32142	0x7d8e	WriteDataError	RobotLanguage_Itfs
32143	0x7d8f	InvalidCircParameter	RobotLanguage_Itfs
32144	0x7d90	InvalidVariableValue	RobotLanguage_Itfs
32145	0x7d91	InvalidCommand	RobotLanguage_Itfs
32146	0x7d92	InvalidWordNumber	RobotLanguage_Itfs
32147	0x7d93	InvalidState	RobotLanguage_Itfs
32148	0x7d94	InvalidVarLength	RobotLanguage_Itfs
32149	0x7d95	InvalidInput	RobotLanguage_Itfs
32150	0x7d96	EventNumberOverflow	RobotLanguage_Itfs
32151	0x7d97	EventBuffer	RobotLanguage_Itfs
32152	0x7d98	EventExecution	RobotLanguage_Itfs
32153	0x7d99	InvalidEventCommand	RobotLanguage_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32154	0x7d9a	ContinueMissing	RobotLanguage_Itfs
32160	0x7da0	RobLangEditor_ISRL_Data_NotConnected	RobotLanguage_Editor_Itfs
32161	0x7da1	RobLangEditor_IRobMoCOutForEditor_NotConnected	RobotLanguage_Editor_Itfs
32162	0x7da2	RobLangEditor_IProgramInterpreter_NotConnected	RobotLanguage_Editor_Itfs
32163	0x7da3	RobLangEditor_GuiCommError	RobotLanguage_Editor_Itfs
32164	0x7da4	RobLangEditor_InvalidRef_SRL_Data	RobotLanguage_Editor_Itfs
32165	0x7da5	RobLangEditor_InvalidRef_ProgInterpreterOut	RobotLanguage_Editor_Itfs
32169	0x7da9	RobLangEditor_ReceivedData_ProgIndexOutOfRange	RobotLanguage_Editor_Itfs
32170	0x7daa	RobLangEditor_ReceivedData_ProgTooLong	RobotLanguage_Editor_Itfs
32171	0x7dab	RobLangEditor_ReceivedData_SetBoolVarValue_IndexOutOfRange	RobotLanguage_Editor_Itfs
32172	0x7dac	RobLangEditor_ReceivedData_SetBoolVarName_IndexOutOfRange	RobotLanguage_Editor_Itfs
32173	0x7dad	RobLangEditor_ReceivedData_SetRealVarValue_IndexOutOfRange	RobotLanguage_Editor_Itfs
32174	0x7dae	RobLangEditor_ReceivedData_SetRealVarName_IndexOutOfRange	RobotLanguage_Editor_Itfs
32175	0x7daf	RobLangEditor_ReceivedData_SetPoseVarValue_IndexOutOfRange	RobotLanguage_Editor_Itfs
32177	0x7db1	RobLangEditor_ReceivedData_SetPoseVarName_IndexOutOfRange	RobotLanguage_Editor_Itfs
32178	0x7db2	RobLangEditor_ReceivedTelegramTypeNotSupported	RobotLanguage_Editor_Itfs
32179	0x7db3	RobLangEditor_Send_NameRequestNotSupported	RobotLanguage_Editor_Itfs
32180	0x7db4	RobLangEditor_Send_ValueRequestNotSupported	RobotLanguage_Editor_Itfs
32181	0x7db5	RobLangEditor_Send_RequestNotSupported	RobotLanguage_Editor_Itfs
32182	0x7db6	InterfaceError	RobotLanguage_Editor_Itfs
32183	0x7db7	RobLangEditor_InvalidRef_MotionSet	RobotLanguage_Editor_Itfs
32184	0x7db8	RobLangEditor_IndexOutOfRange	RobotLanguage_Editor_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32185	0x7db9	RobLangEditor_ResetComNotSuccessfull	RobotLanguage_Editor_Itfs
32186	0x7dba	RobLangEditor_ReceivedData_RequestedProgIndexOutOfRange	RobotLanguage_Editor_Itfs
32187	0x7dbb	RobLangEditor_InvalidRef_ActivePathEvents	RobotLanguage_Editor_Itfs
32188	0x7dbc	RobLangEditor_InvalidRef_TriggeredPathEvents	RobotLanguage_Editor_Itfs
32189	0x7dbd	RobLangEditor_InvalidRef_ActiveTpEvent	RobotLanguage_Editor_Itfs
32190	0x7dbe	RobLangEditor_InvalidRef_DeregTpEvents	RobotLanguage_Editor_Itfs
32191	0x7dbf	RobLangEditor_WriteCommentsError	RobotLanguage_Editor_Itfs
32192	0x7dc0	RobMon_InvalidInterface	RobotMonitor_Itfs
32193	0x7dc1	RobMon_GuiCommError	RobotMonitor_Itfs
32194	0x7dc2	RobMon_InvalidRef	RobotMonitor_Itfs
32195	0x7dc3	RobMon_InvalidOperatingModeReceived	RobotMonitor_Itfs
32196	0x7dc4	RobMon_InvalidJogCoordSysReceived	RobotMonitor_Itfs
32197	0x7dc5	RobMon_LinkRobotNotSuccessFull	RobotMonitor_Itfs
32224	0x7de0	InterfaceNotValid	RobotUI_Itfs
32225	0x7de1	SetMotionParamSet_OnlyIfInactive	RobotUI_Itfs
32256	0x7e00	Internal_SeeDetailsInLogbook	RobotMotionControl_Itfs
32257	0x7e01	ConfigurationReadFileFailed	RobotMotionControl_Itfs
32258	0x7e02	ConfigurationReadParameterFailed	RobotMotionControl_Itfs
32259	0x7e03	ConfigurationVersionNotSupported	RobotMotionControl_Itfs
32260	0x7e04	ConfigurationNotValid	RobotMotionControl_Itfs
32261	0x7e05	ErrorResetOnlyPossibleAfterProgramStop	RobotMotionControl_Itfs
32262	0x7e06	ModuleNotLinkable	RobotMotionControl_Itfs
32263	0x7e07	MaxMotionQueueSizeTooSmall	RobotMotionControl_Itfs
32272	0x7e10	LicenseNotAvailable	RobotMotionControl_Itfs
32273	0x7e11	LicenseNotRequested	RobotMotionControl_Itfs
32274	0x7e12	LicenseCorrupted	RobotMotionControl_Itfs
32275	0x7e13	JointEStopDecelerationOutOfRange	RobotMotionControl_Itfs
32276	0x7e14	JointEstopJerkOutOfRange	RobotMotionControl_Itfs
32277	0x7e15	JointLimitSwitchOutOfRange	RobotMotionControl_Itfs
32278	0x7e16	RotationEStopDecelerationOutOfRange	RobotMotionControl_Itfs
32279	0x7e17	RotationEStopJerkOutOfRange	RobotMotionControl_Itfs
32280	0x7e18	TranslationEStopDecelerationOutOfRange	RobotMotionControl_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32281	0x7e19	TranslationEStopJerkOutOfRange	RobotMotionControl_Itfs
32282	0x7e1a	CartesianLimitSwitchOutOfRange	RobotMotionControl_Itfs
32288	0x7e20	AxisJointTransformationNotLinked	RobotMotionControl_Itfs
32289	0x7e21	AxisJointTransformationInterfaceZero	RobotMotionControl_Itfs
32290	0x7e22	AxisJointTransformationSetupFailed	RobotMotionControl_Itfs
32291	0x7e23	AxisJointTransformationFailed	RobotMotionControl_Itfs
32292	0x7e24	KinematicModelNotLinked	RobotMotionControl_Itfs
32293	0x7e25	KinematicModelInterfaceZero	RobotMotionControl_Itfs
32294	0x7e26	ForwardKinematicsFailed	RobotMotionControl_Itfs
32295	0x7e27	ForwardKinematicsInterfaceCorrupted	RobotMotionControl_Itfs
32296	0x7e28	InverseKinematicsFailed	RobotMotionControl_Itfs
32297	0x7e29	InverseKinematicsInterfaceCorrupted	RobotMotionControl_Itfs
32298	0x7e2a	ReadAxesFailed	RobotMotionControl_Itfs
32299	0x7e2b	WriteAxesFailed	RobotMotionControl_Itfs
32300	0x7e2c	AxesConfigNotLinked	RobotMotionControl_Itfs
32301	0x7e2d	AxesConfigInterfaceZero	RobotMotionControl_Itfs
32304	0x7e30	MotionControlNotSetUp	RobotMotionControl_Itfs
32305	0x7e31	MotionSetNumberInvalid	RobotMotionControl_Itfs
32306	0x7e32	JointAccelerationOutOfRange	RobotMotionControl_Itfs
32307	0x7e33	JointDecelerationOutOfRange	RobotMotionControl_Itfs
32308	0x7e34	JointJerkOutOfRange	RobotMotionControl_Itfs
32309	0x7e35	JointVelocityOutOfRange	RobotMotionControl_Itfs
32310	0x7e36	RotationAccelerationOutOfRange	RobotMotionControl_Itfs
32311	0x7e37	RotationDecelerationOutOfRange	RobotMotionControl_Itfs
32312	0x7e38	RotationJerkOutOfRange	RobotMotionControl_Itfs
32313	0x7e39	RotationVelocityOutOfRange	RobotMotionControl_Itfs
32314	0x7e3a	TranslationAccelerationOutOfRange	RobotMotionControl_Itfs
32315	0x7e3b	TranslationDecelerationOutOfRange	RobotMotionControl_Itfs
32316	0x7e3c	TranslationJerkOutOfRange	RobotMotionControl_Itfs
32317	0x7e3d	TranslationVelocityOutOfRange	RobotMotionControl_Itfs
32318	0x7e3e	OverrideOutOfRange	RobotMotionControl_Itfs
32319	0x7e3f	TimeOverrideOutOfRange	RobotMotionControl_Itfs
32320	0x7e40	MotionTypeInvalid	RobotMotionControl_Itfs
32321	0x7e41	BlendingDistanceNegative	RobotMotionControl_Itfs
32322	0x7e42	CircleCenterCoordinatesLeadingToRadiusZero	RobotMotionControl_Itfs
32323	0x7e43	CircleCenterCorrectionLargerThanThreshold	RobotMotionControl_Itfs
32324	0x7e44	CircleCenterCorrectionThresholdNotPositive	RobotMotionControl_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32325	0x7e45	CircleCenterEqualToStartOrEndPoint	RobotMotionControl_Itfs
32326	0x7e46	CircleCoordinatesUnclear	RobotMotionControl_Itfs
32327	0x7e47	CircleDirectionInvalid	RobotMotionControl_Itfs
32328	0x7e48	CircleRadiusZero	RobotMotionControl_Itfs
32329	0x7e49	CircleTypeInvalid	RobotMotionControl_Itfs
32330	0x7e4a	CircleTypeRadiusAngleRequiresPositiveRadius	RobotMotionControl_Itfs
32331	0x7e4b	PlaneInvalid	RobotMotionControl_Itfs
32332	0x7e4c	TangentPerpendicularToPlane	RobotMotionControl_Itfs
32336	0x7e50	CoordinateSystemNotSupported	RobotMotionControl_Itfs
32337	0x7e51	UserCoordinateSystemInterfaceZero	RobotMotionControl_Itfs
32338	0x7e52	UserCoordinateSystemInterfaceCorrupted	RobotMotionControl_Itfs
32339	0x7e53	UserCoordinateSystemTypeNotSupported	RobotMotionControl_Itfs
32340	0x7e54	UserCoordinateSystemVelocityOutOfRange	RobotMotionControl_Itfs
32341	0x7e55	UserCoordinateSystemAccelerationOutOfRange	RobotMotionControl_Itfs
32352	0x7e60	OutOfWorkspace	RobotMotionControl_Itfs
32353	0x7e61	JogAxisSinceNotReferenced	RobotMotionControl_Itfs
32354	0x7e62	JogAxisSinceJointsNotValid	RobotMotionControl_Itfs
32355	0x7e63	JogAxisOrJointSinceCartPosNotValid	RobotMotionControl_Itfs
32368	0x7e70	NoSolutionFound	RobotMotionControl_Itfs
32384	0x7e80	PathEventInterfaceOfInterpreterZero	RobotMotionControl_Itfs
32385	0x7e81	PathEventQueryInterfaceNotSuccessful	RobotMotionControl_Itfs
32386	0x7e82	PathEventPositionBeforePathBegin	RobotMotionControl_Itfs
32387	0x7e83	PathEventTimeShiftExceedsRemainingTimeInMotion	RobotMotionControl_Itfs
32400	0x7E90	TouchProbeInterfaceZero	RobotMotionControl_Itfs
32401	0x7E91	TouchProbeQueryInterfaceNotSuccessful	RobotMotionControl_Itfs
32402	0x7E92	TouchProbePositioningTriggerBeforeContinue	RobotMotionControl_Itfs
32403	0x7E93	TouchProbeForLinearInterpolation	RobotMotionControl_Itfs
32404	0x7E94	TouchProbePosRequiresTransInMeasuringDirection	RobotMotionControl_Itfs
32405	0x7E95	TouchProbePositioningDistanceTooShort	RobotMotionControl_Itfs
32406	0x7E96	TouchProbePositioningEndInNonTangentialBlending	RobotMotionControl_Itfs
32407	0x7E97	TouchProbePositioningEndBeforeActualSetpoint	RobotMotionControl_Itfs
32408	0x7E98	TouchProbeMultipleSimultaneousContinue	RobotMotionControl_Itfs
32416	0x7ea0	JogVelocityPercentOutOfRange	RobotMotionControl_Itfs
32512	0x7f00	Internal_SeeDetailsInLogbook	RobotMathematics_Itfs
32544	0x7f20	Internal_SeeDetailsInLogbook	RobotProfileGeneration
32545	0x7f21	VelocityZeroOrNegative	RobotProfileGeneration
32546	0x7f22	AccelerationZeroOrNegative	RobotProfileGeneration

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32547	0x7f23	AccelerationGreaterThanRapidDeceleration	RobotProfileGeneration
32548	0x7f24	JerkZeroOrNegative	RobotProfileGeneration
32549	0x7f25	JerkGreaterThanRapidJerk	RobotProfileGeneration
32550	0x7f26	SetPositionNotReachableWithSettings	RobotProfileGeneration
32551	0x7f27	DecelerationZeroOrNegative	RobotProfileGeneration
32552	0x7f28	DecelerationGreaterThanRapidDeceleration	RobotProfileGeneration
32553	0x7f29	SetPositionZero	RobotProfileGeneration
32576	0x7f40	Internal_SeeDetailsInLogbook	RobotAxisJointTransformation_ltfs
32577	0x7f41	ConfigurationReadFileFailed	RobotAxisJointTransformation_ltfs
32578	0x7f42	ConfigurationReadParameterFailed	RobotAxisJointTransformation_ltfs
32579	0x7f43	ConfigurationVersionNotSupported	RobotAxisJointTransformation_ltfs
32580	0x7f44	ConfigurationNotValid	RobotAxisJointTransformation_ltfs
32581	0x7f45	AxisGroupPositioningInterpolatedNotLinked	RobotAxisJointTransformation_ltfs
32582	0x7f46	AxisGroupPositioningInterpolatedInterfaceZero	RobotAxisJointTransformation_ltfs
32583	0x7f47	ParameterValueInvalid	RobotAxisJointTransformation_ltfs
32584	0x7f48	TouchProbeValueMultidimensionalNotLinked	RobotAxisJointTransformation_ltfs
32585	0x7f49	TouchProbeValueMultidimensionalInterfaceZero	RobotAxisJointTransformation_ltfs
32586	0x7f4a	UnknownCrankType	RobotAxisJointTransformation_ltfs
32587	0x7f4B	NegativeEffectiveWindingRadius	RobotAxisJointTransformation_ltfs
32608	0x7f60	Internal_SeeDetailsInLogbook	RobotKinematicModel_ltfs
32609	0x7f61	ConfigurationReadFileFailed	RobotKinematicModel_ltfs
32610	0x7f62	ConfigurationReadParameterFailed	RobotKinematicModel_ltfs
32611	0x7f63	ConfigurationVersionNotSupported	RobotKinematicModel_ltfs
32612	0x7f64	ConfigurationNotValid	RobotKinematicModel_ltfs
32613	0x7f65	ConfiguredModelDiffersFromLinked	RobotKinematicModel_ltfs
32614	0x7f66	ParameterValueOutOfRange	RobotKinematicModel_ltfs
32624	0x7f70	PositionNotPossible	RobotKinematicModel_ltfs
32625	0x7f71	OrientationNotPossible	RobotKinematicModel_ltfs
32626	0x7f72	PoseNotReachable	RobotKinematicModel_ltfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
32627	0x7f73	CartesianAssignmentNotAllowed	RobotKinematicModel_Itfs
32628	0x7f74	ConstellationNotSupported	RobotKinematicModel_Itfs
32629	0x7f75	ConstellationNotAllowedInPose	RobotKinematicModel_Itfs
32768	0x8000	Internal_SeeDetailsInLogbook	RobotUserCoordinateSys-tems_Itfs
32800	0x8020	InvalidInterface	RobotTouchprobe_Itfs
32801	0x8021	SEW_IRobLangNotLinked	RobotTouchprobe_Itfs
32802	0x8022	SEW_ITouchProbeProfileNotLinked	RobotTouchprobe_Itfs
32803	0x8023	SEW_ITransformNotLinked	RobotTouchprobe_Itfs
32804	0x8024	SEW_IExecuteEventNotLinked	RobotTouchprobe_Itfs
32805	0x8025	LinkableModuleNotUsed	RobotTouchprobe_Itfs
32806	0x8026	InvalidSource	RobotTouchprobe_Itfs
32807	0x8027	InvalidMode	RobotTouchprobe_Itfs
32808	0x8028	InvalidLevel	RobotTouchprobe_Itfs
32809	0x8029	InvalidMeasuringDirection	RobotTouchprobe_Itfs
32810	0x802A	InvalidSourceVarIndex	RobotTouchprobe_Itfs
32811	0x802B	SeveralTouchProbesAtSameTime	RobotTouchprobe_Itfs
32812	0x802C	CounterOverflowTouchProbeCounter	RobotTouchprobe_Itfs
32813	0x802D	CounterOverflowActiveTPEvent_ChangeCounter	RobotTouchprobe_Itfs
32814	0x802E	ErrorResetOnlyPossibleAfterProgramStop	RobotTouchprobe_Itfs
32815	0x802F	TransformationNotSuccessful	RobotTouchprobe_Itfs
32816	0x8030	ExecuteEventNotSuccessful	RobotTouchprobe_Itfs
32817	0x8031	ContinueBeforeRegister	RobotTouchprobe_Itfs
32818	0x8032	Internal_SeeDetailsInLogbook	RobotTouchprobe_Itfs
33024	0x8100	InterfaceNotValid	RobotPD
33025	0x8101	ErrorReadingConfigFile	RobotPD
33026	0x8102	ErrorReadingParameter	RobotPD
33027	0x8103	RequiredLengthOfProfileToLarge	RobotPD
33028	0x8104	uiProcessDataProfile_NoConsistency	RobotPD
33029	0x8105	InternalProfileLengthCheckDefect	RobotPD
33030	0x8106	InvalidReferenceOnProgramVariables	RobotPD
33031	0x8107	InternalConfigCheckInvalid	RobotPD
33032	0x8108	ValueOutOfBoundsOfDatatype	RobotPD
33033	0x8109	WrongConfiguredSizeOnBusOfReal	RobotPD
33041	0x8111	JogCoordinateSystemInvalid	RobotPD
33184	0x81a0	RobLangEditor_ReadCommentsFileError	RobotLanguage_Edi-tor_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
33185	0x81a1	RobLangEditor_SaveCommentsFileError	RobotLanguage_Editor_Itfs
33186	0x81a2	RobLangEditor_LoadCommentsFileError	RobotLanguage_Editor_Itfs
33280	0x8200	Simu3D_InvalidInterface	Simu3D_Itfs
33281	0x8201	Simu3D_InvalidRef	Simu3D_Itfs
33282	0x8202	Simu3D_GuiCommError	Simu3D_Itfs
33283	0x8203	Simu3D_AddUserCS_IndexOutOfRange	Simu3D_Itfs
33284	0x8204	Simu3D_UserCS_GetTransformNotSuccessful	Simu3D_Itfs
33285	0x8205	Simu3D_NumberOfTransmittedKinParsToSmall	Simu3D_Itfs
33286	0x8206	Simu3D_TimeoutModelAcknowledge	Simu3D_Itfs
33287	0x8207	Simu3D_TimeoutConfigAcknowledge	Simu3D_Itfs
34624	0x8740	IUSERUNITSTOSI_NOT_LINKED	PhysicDrive
34625	0x8741	IUSERUNITSTOSI_EMPTY	PhysicDrive
34631	0x8747	ICOMPONENTDYNAMIC_HAS_NO_ITaskHighPrioAndInit	PhysicDrive
34632	0x8748	ICOMPONENTDYNAMIC_IS_EMPTY	PhysicDrive
34633	0x8749	ICOMPONENTDYNAMIC_HAS_NO_IERROR	PhysicDrive
34634	0x874a	ICOMPONENTDYNAMIC_HAS_NO_IUSERUNITSTOSI	PhysicDrive
34636	0x874c	LINKIUSERUNITSTOSI_NOT_SUCCESSFUL	PhysicDrive
34637	0x874d	LinkISuperordinatedFB_NOT_SUCCESSFUL	PhysicDrive
34638	0x874e	IDRIVEINTERFACE_EMPTY	PhysicDrive
34639	0x874f	IDRIVEINTERFACE_NOT_LINKED	PhysicDrive
34640	0x8750	IDRIVEINTERFACE_HAS_NO_ERRORHANDLING	PhysicDrive
34641	0x8751	ITaskHighPrioAndInit_NOT_LINKED	PhysicDrive
34688	0x8780	FileCouldNotBeWriten	PhysicExport
34689	0x8781	CouldNotGetDateAndTime	PhysicExport
34690	0x8782	ExportArrayToSmallForRecord	PhysicExport
34691	0x8783	AddingStringToBufferFailed	PhysicExport
34692	0x8784	CouldNotLinkDrive	PhysicExport
34693	0x8785	MaximumNumberOfDrivesReached	PhysicExport
34694	0x8786	NoDrivesLinked	PhysicExport
34695	0x8787	DynamicModelNotLinked	PhysicExport
34696	0x8788	UserUnitsNotLinked	PhysicExport
34697	0x8789	InterfaceNotValid	PhysicExport
34698	0x878a	CouldNotLinkGear	PhysicExport
34699	0x878b	MaximumNumberOfGearsReached	PhysicExport
34700	0x878c	LinkInterfacesNotCompleted	PhysicExport
34720	0x87a0	IUSERUNITSTOSI_NOT_LINKED	PhysicGear

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
34721	0x87a1	IUSERUNITSTOSI_EMPTY	PhysicGear
34722	0x87a2	DRIVE_LINEAR_NOGEAR_ALLOWED	PhysicGear
34723	0x87a3	GEAR_RATIO_WRONG	PhysicGear
34724	0x87a4	GEAR_INERTIA_NEG	PhysicGear
34726	0x87a6	FRICTION_OFFSET_NEG	PhysicGear
34727	0x87a7	FRICTION_THRESHOLD_NEG	PhysicGear
34728	0x87a8	FRICTION_THRESHOLD_ZERO	PhysicGear
34729	0x87a9	ICOMPONENTDYNAMIC_IS_EMPTY	PhysicGear
34730	0x87aa	ICOMPONENTDYNAMIC_HAS_NO_ITASKSIMPLE	PhysicGear
34731	0x87ab	ICOMPONENTDYNAMIC_HAS_NO_IERROR	PhysicGear
34732	0x87ac	ICOMPONENTDYNAMIC_HAS_NO_IUSERUNITSTOSI	PhysicGear
34733	0x87ad	LinkISuperordinatedFB_NOT_SUCCESSFUL	PhysicGear
34734	0x87ae	LINKIUSERUNITSTOSI_NOT_SUCCESSFUL	PhysicGear
34735	0x87af	ITASKSIMPLE_NOT_LINKED	PhysicGear
34736	0x87b0	RADIUS_ZERO_OR_NEGATIVE	PhysicGear
34737	0x87b1	UNKNOWN_FRICTION_MODEL	PhysicGear
34738	0x87b2	FRICTION_HYS_OFFSET_NEG	PhysicGear
34739	0x87b3	FRICTION_HYS_OFFSET2_NEG	PhysicGear
34740	0x87b4	FRICTION_LUGRE_OFFSET_NEG	PhysicGear
34784	0x87e0	SmoothPufferSizeOutOfBounds	PhysicMathematics
34785	0x87e1	SmoothFilterRangeOutOfBounds	PhysicMathematics
34786	0x87e2	DeadTimeToLargeForPufferSize	PhysicMathematics
34816	0x8800	InterfaceNotValid	PhysicMeasuring
34817	0x8801	ModuleNotNeeded	PhysicMeasuring
34818	0x8802	eUseNotValid	PhysicMeasuring
34848	0x8820	IUSERUNITSTOSI_NOT_LINKED	PhysicMotor
34849	0x8821	IUSERUNITSTOSI_EMPTY	PhysicMotor
34850	0x8822	IUSERUNITSTOSI_NOT_INITIALIZED	PhysicMotor
34851	0x8823	MOTOR_INERTIA_NEG	PhysicMotor
34852	0x8824	MOTOR_MAG_FORCE_NEG	PhysicMotor
34853	0x8825	IPLC_NOT_LINKED	PhysicMotor
34854	0x8826	IMOTORINTERFACE_EMPTY	PhysicMotor
34855	0x8827	IMOTORINTERFACE_HAS_NO_ERRORHANDLING	PhysicMotor
34856	0x8828	IPositioningInterpolatedEmpty	PhysicMotor
34857	0x8829	MotorNominalTorqueOrForceNotPositive	PhysicMotor
34858	0x882a	NotConfigured	PhysicMotor
34944	0x8880	USERUNIT_FACTOR_WRONG	PhysicUserUnits

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
34945	0x8881	USERUNIT_UNKNOWN	PhysicUserUnits
34946	0x8882	USERUNIT_ANGLE_WRONG	PhysicUserUnits
34976	0x88a0	INTERFACE_NOT_LINKED	PhysicCollisionDetection
34977	0x88a1	COLLISION_REACTION_FINISHED	PhysicCollisionDetection
34978	0x88a2	TORQUE_THRESHOLD_NEGATIVE	PhysicCollisionDetection
34979	0x88a3	VELOCITY_THRESHOLD_NEGATIVE	PhysicCollisionDetection
34980	0x88a4	NOMINAL_TORQUE_FORCE_ZERO_OR_NEGATIVE	PhysicCollisionDetection
34981	0x88a5	DEADTIME_CYCLES_TOO_HIGH	PhysicCollisionDetection
34982	0x88a6	REF_NOT_VALID	PhysicCollisionDetection
35072	0x8900	InterfaceNotValid	DynamicModel
35073	0x8901	InternalNewDriveIndexInvalid	DynamicModel
35074	0x8902	InternalNewGearIndexInvalid	DynamicModel
35075	0x8903	InvalidReferenceOfMotionControl	DynamicModel
35076	0x8904	ErrorReadingConfigFile	DynamicModel
35077	0x8905	ErrorReadingParameter	DynamicModel
35078	0x8906	IndexOfGetInputForceOutOfBounds	DynamicModel
35079	0x8907	IndexOfGetInputGravityOutOfBounds	DynamicModel
35080	0x8908	TimeLimitForLicenselessUseReached	DynamicModel
35081	0x8909	ReferenceNotValid	DynamicModel
35082	0x890a	InternalError	DynamicModel
35083	0x890b	MassNegative	DynamicModel
35084	0x890c	InertiaNegative	DynamicModel
35085	0x890d	UnableToDetermineCycleTime	DynamicModel
35104	0x8920	InterfaceNotValid	DynamicModelDelta
35105	0x8921	MassNegative	DynamicModelDelta
35106	0x8922	InertiaMatriceAssymetric	DynamicModelDelta
35107	0x8923	InertiaMatriceNegativeComponents	DynamicModelDelta
35108	0x8924	CrankLengthZero	DynamicModelDelta
35115	0x892b	KinematicConfigurationIsNotSymetric	DynamicModelDelta
35116	0x892c	KinematicConfigurationIsNegative	DynamicModelDelta
35117	0x892d	KinematicConfigurationNotSupported	DynamicModelDelta
35118	0x892e	TelescopicShaftNotRespectedInModel	DynamicModelDelta
35136	0x8940	InterfaceNotValid	DynamicModelOKC
35136	0x8940	InterfaceNotValid	DynamicModelRigidBody
35137	0x8941	InternalError	DynamicModelOKC
35138	0x8942	InternalGlobalConstantOfLinksWrong	DynamicModelOKC
35139	0x8943	ExtraBearingRequired	DynamicModelOKC

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
35140	0x8944	ExtraBearingNeedless	DynamicModelOKC
35141	0x8945	LinkMassNegative	DynamicModelOKC
35142	0x8946	LinkInertiaNegative	DynamicModelOKC
35143	0x8947	LinkNumberOfPartsOutOfBounds	DynamicModelOKC
35143	0x8947	LinkNumberOfPartsOutOfBounds	DynamicModelRigidBody
35144	0x8948	LinkEffectDirectionZero	DynamicModelOKC
35144	0x8948	LinkEffectDirectionZero	DynamicModelRigidBody
35145	0x8949	GeometryUnknown	DynamicModelOKC
35145	0x8949	GeometryUnknown	DynamicModelRigidBody
35146	0x894a	PrimitiveMassNegative	DynamicModelOKC
35146	0x894a	PrimitiveMassNegative	DynamicModelRigidBody
35147	0x894b	PrimitiveDensitiyNegative	DynamicModelOKC
35147	0x894b	PrimitiveDensitiyNegative	DynamicModelRigidBody
35148	0x894c	PrimitiveLengthXNegative	DynamicModelOKC
35148	0x894c	PrimitiveLengthXNegative	DynamicModelRigidBody
35149	0x894d	PrimitiveLengthYNegative	DynamicModelOKC
35149	0x894d	PrimitiveLengthYNegative	DynamicModelRigidBody
35150	0x894e	PrimitiveLengthZNegative	DynamicModelOKC
35150	0x894e	PrimitiveLengthZNegative	DynamicModelRigidBody
35151	0x894f	PrimitiveCylinderDiameterNegative	DynamicModelOKC
35151	0x894f	PrimitiveCylinderDiameterNegative	DynamicModelRigidBody
35152	0x8950	MountedDriveArrayFull	DynamicModelOKC
35152	0x8950	MountedDriveArrayFull	DynamicModelRigidBody
35153	0x8951	InternalNumberOfLinksZero	DynamicModelOKC
35154	0x8952	InternalWrongConfig	DynamicModelOKC
35155	0x8953	InternalNewLinkIndexInvalid	DynamicModelOKC
35156	0x8954	PrimitiveArrayFull	DynamicModelOKC
35156	0x8954	PrimitiveArrayFull	DynamicModelRigidBody
35157	0x8955	ModuleNotNeeded	DynamicModelOKC
35157	0x8955	ModuleNotNeeded	DynamicModelRigidBody
35158	0x8956	CouldNotGetLCS	DynamicModelOKC
35159	0x8957	ReferenceNotValid	DynamicModelOKC
35168	0x8960	InterfaceNotValid	DynamicModelRollerGantry
35169	0x8961	WrongConfig_MassNegative_CartesianJoint	DynamicModelRollerGantry
35172	0x8964	WrongConfig_InertiaNegative_BeltAndPulley	DynamicModelRollerGantry

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
35173	0x8965	WrongConfig_PretensionNegative	DynamicModelRollerGantry
35174	0x8966	WrongConfig_FrictionOffsetNegative	DynamicModelRollerGantry
35175	0x8967	WrongConfig_FrictionThresholdNegative	DynamicModelRollerGantry
35176	0x8968	WrongConfig_FrictionThresholdZero	DynamicModelRollerGantry
35177	0x8969	WrongConfig_TypeNotKnown	DynamicModelRollerGantry
35178	0x896a	WrongConfig_BeltWidthNegative	DynamicModelRollerGantry
35200	0x8980	AlgorithmOutOfBounds	DynamicModelTripod
35201	0x8981	UpperarmMassNeg	DynamicModelTripod
35202	0x8982	LowerarmMassNeg	DynamicModelTripod
35203	0x8983	UpperarmInertiaNeg	DynamicModelTripod
35204	0x8984	LowerarmInertiaNeg	DynamicModelTripod
35206	0x8986	CouldNotGetLCS	DynamicModelTripod
35207	0x8987	TripodHasTelescopicshaft	DynamicModelTripod
35208	0x8988	UpperArmRadiusCoGZero	DynamicModelTripod
35209	0x8989	LowerArmCoGZeroOrNegative	DynamicModelTripod
35211	0x898b	DENOMINATOR_ZERO	DynamicModelTripod
35212	0x898c	INTERNAL_ERROR	DynamicModelTripod

26873346/DE – 07/2021

15.2.2 Robotics Meldungen

Dez	Hex	Meldung	Bibliothek SEW_MOS_...
34848	0x8820	MOTOR_INERTIA_ZERO	PhysicMotor
34849	0x8821	MOTOR_MAG_FORCE_ZERO	PhysicMotor
34850	0x8822	NO_SPECIFIC_MOTOR_DEFINED	PhysicMotor
97600	0x17d40	WhishedEnableModeNotPossible	RobotHandler_Itfs
97601	0x17d41	WaitingForActiveControlMode	RobotHandler_Itfs
97602	0x17d42	WaitingForEnable	RobotHandler_Itfs
97603	0x17d43	SafetyControllerRequestsEmergencyStopAxes	RobotHandler_Itfs
97604	0x17d44	SafetyControllerRequestsEmergencyStop	RobotHandler_Itfs
97605	0x17d45	SafetyControllerRequestsApplicationStop	RobotHandler_Itfs
97609	0x17d49	Inverter_SafeStop	RobotHandler_Itfs
97610	0x17d4a	RisingEdgeOfStartIgnoredBecauseProgramPause	RobotHandler_Itfs
97611	0x17d4b	RisingEdgeOfStartIgnoredBecauseProgramStop	RobotHandler_Itfs
97612	0x17d4c	RisingEdgeOfStartIgnoredBecauseProgramNotYetInitialized	RobotHandler_Itfs
97613	0x17d4d	RisingEdgeOfStartIgnoredBecauseRobotNotReadyToMove	RobotHandler_Itfs
97614	0x17d4e	RisingEdgeOfStartIgnoredBecauseProgramIsBeeingExecuted	RobotHandler_Itfs
97615	0x17d4f	Inverter_NotReferenced	RobotHandler_Itfs
97632	0x17d60	RobotHMI_MsgHmiConnected	RobotHMI_Itfs
97635	0x17d63	RobotHMI_ReceiveToolVersionTimeOut	RobotHMI_Itfs
97636	0x17d64	RobotHMI_ConnectionLossDuringSubmodulesConnect	RobotHMI_Itfs
97696	0x17da0	RobLangEditor_FailedFBCallsInGuiCom	RobotLanguage_Editor_Itfs
97697	0x17da1	RobLangEditor_LoadingCommentsNotReadyYet	RobotLanguage_Editor_Itfs
97760	0x17de0	UseNewFunctionInsteadOfTemporary	RobotUI_Itfs
97792	0x17e00	Internal_SeeDetailsInLogbook	RobotMotionControl_Itfs
97793	0x17e01	NotReferenced	RobotMotionControl_Itfs
97794	0x17e02	JointsNotValid	RobotMotionControl_Itfs
97795	0x17e03	CartPosNotValid	RobotMotionControl_Itfs
97808	0x17e10	JogKinematicJointDOFNotExisting	RobotMotionControl_Itfs
97809	0x17e11	JogAdditionalJointDOFNotExisting	RobotMotionControl_Itfs
97810	0x17e12	JogCartesianDOFNotExisting	RobotMotionControl_Itfs
97811	0x17e13	JogKinematicJointStoppedDueToSWLS	RobotMotionControl_Itfs
97812	0x17e14	JogAdditionalJointStoppedDueToSWLS	RobotMotionControl_Itfs
97813	0x17e15	JogCartesianStoppedDueToSWLS	RobotMotionControl_Itfs
97814	0x17e16	JogKinematicAxisDOFNotExisting	RobotMotionControl_Itfs
97824	0x17e20	LimitedSpeedDueToCentrifugalAcc	RobotMotionControl_Itfs

26873346/DE – 07/2021

Dez	Hex	Meldung	Bibliothek SEW_MOS_...
97827	0x17e23	AxisJointTransformationFailed	RobotMotionControl_Itfs
97830	0x17e26	ForwardKinematicsFailed	RobotMotionControl_Itfs
97888	0x17e60	OutOfWorkspace	RobotMotionControl_Itfs
97889	0x17e61	SWLSIgnored	RobotMotionControl_Itfs
97920	0x17e80	PathEventRegistrationDelayedMaximumNumberReached	RobotMotionControl_Itfs
97921	0x17e81	PathEventTimeShiftDelaysMotion	RobotMotionControl_Itfs
97922	0x17e82	PathEventTimeShiftElapsedNotInProgramMode	RobotMotionControl_Itfs
97923	0x17e83	PathEventStillActiveAfterMotionEnd	RobotMotionControl_Itfs
97936	0x17e90	TouchProbePositioningMotionParametersIncreased	RobotMotionControl_Itfs
98080	0x17f20	UseOfIncreasedDeceleration	RobotProfileGeneration
98081	0x17f21	UseOfIncreasedJerk	RobotProfileGeneration
98082	0x17f22	UseOfIncreasedDecelerationAndIncreasedJerk	RobotProfileGeneration
98112	0x17f40	Internal_SeeDetailsInLogbook	RobotAxisJointTransformation_Itfs
98144	0x17f60	Internal_SeeDetailsInLogbook	RobotKinematicModel_Itfs
98336	0x18020	PositioningWhileMultipleMode	RobotTouchprobe_Iffs
98337	0x18021	TouchProbeCounterReachingMaxValueRestartAt0	RobotTouchprobe_Itfs
98816	0x18200	Simu3D_TimeoutModelAcknowledge	Simu3D_Itfs
98817	0x18201	Simu3D_TimeoutConfigAcknowledge	Simu3D_Itfs
98818	0x18202	Simu3D_FailedFBCallsInGuiCom	Simu3D_Itfs
98819	0x18203	Simu3D_InterfaceMissing	Simu3D_Itfs
100224	0x18780	NoRecordedCyclesToExport	PhysicExport
100225	0x18781	InternalWarning	PhysicExport
100226	0x18782	RecordingSequenceCanceled	PhysicExport
100256	0x187a0	GEAR_RATIO_ONE	PhysicGear
100257	0x187a1	GEAR_INERTIA_ZERO	PhysicGear
100258	0x187a2	NO_SPECIFIC_GEAR_DEFINED	PhysicGear
100259	0x187a3	FRICTION_OFFSET_ZERO	PhysicGear
100260	0x187a4	FRICTION_GRADIENT_UNNORMAL	PhysicGear
100261	0x187a5	FRICTION_HYS_GRADIENT_UNNORMAL	PhysicGear
100262	0x187a6	FRICTION_HYS_OFFSET_ZERO	PhysicGear
100263	0x187a7	FRICTION_HYS_OFFSET2_ZERO	PhysicGear
100264	0x187a8	FRICTION_HYS_VP_NEG	PhysicGear
100265	0x187a9	FRICTION_LUGRE_OFFSET_ZERO	PhysicGear
100266	0x187aa	FRICTION_LUGRE_FS_UNNORMAL	PhysicGear
100267	0x187ab	FRICTION_LUGRE_SIGMA0_UNNORMAL	PhysicGear
100268	0x187ac	FRICTION_LUGRE_SIGMA1_UNNORMAL	PhysicGear

26873346/DE – 07/2021

Dez	Hex	Meldung	Bibliothek SEW_MOS_...
100269	0x187ad	FRICTION_LUGRE_SIGMA2_UNNORMAL	PhysicGear
100270	0x187ae	FRICTION_LUGRE_VS_UNNORMAL	PhysicGear
100512	0x188a0	COLLISION_DETECTED	PhysicCollisionDetection
100513	0x188a1	EXECUTING_COLLISION_REACTION	PhysicCollisionDetection
100608	0x18900	LicenseMissing	DynamicModel
100672	0x18940	MaxLinksGreater8	DynamicModelOKC
100673	0x18941	LinkMassZero	DynamicModelOKC
100704	0x18960	WrongConfig_MassZero_CartesianJoint	DynamicModelRollerGantry
100706	0x18962	WrongConfig_InertiaZero_BeltAndPulley	DynamicModelRollerGantry
100707	0x18963	WrongConfig_PretensionZero	DynamicModelRollerGantry
100708	0x18964	WrongConfig_FrictionGradientUnnormal	DynamicModelRollerGantry
100712	0x18968	WrongConfig_FrictionOffsetZero	DynamicModelRollerGantry
100713	0x18969	WrongConfig_BeltWidthZero	DynamicModelRollerGantry
100714	0x1896a	CartesianAssignmentNotRespected	DynamicModelRollerGantry
100736	0x18980	UpperarmInertiaZero	DynamicModelTripod
100737	0x18981	LowerarmInertiaZero	DynamicModelTripod
100738	0x18982	UpperarmMassZero	DynamicModelTripod
100739	0x18983	LowerarmMassZero	DynamicModelTripod

15.2.3 Untergeordnete Softwaremodule Fehler

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
25664	0x6440	eSEW_LicMgr_GetInfo	IECLicenseManager
25665	0x6441	eSEW_LicMgr_GetInfo_PerfClass	IECLicenseManager
25666	0x6442	eSEW_LicMgr_CheckAndReportRuntime	IECLicenseManager
25667	0x6443	eSEW_LicMgr_SecretChallenge	IECLicenseManager
25668	0x6444	eSEW_LicMgr_NoRuntime	IECLicenseManager
25669	0x6445	eSEW_LicMgr_NoValidRuntime	IECLicenseManager
25670	0x6446	eSEW_LicMgr_CheckLicense	IECLicenseManager
25671	0x6447	eSEW_LicMgr_ConsumeLicense	IECLicenseManager
25672	0x6448	eSEW_LicMgr_ReportMissingLicense	IECLicenseManager
25673	0x6449	eSEW_LicMgr_FileReloadWatcher	IECLicenseManager
25674	0x644a	eSEW_LicMgr_ConfirmToken	IECLicenseManager

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
25696	0x6460	InterfaceNotValid	ErrorHandling_Itfs
25697	0x6461	SubordinatedFBArrayFull	ErrorHandling_Itfs
25698	0x6462	ErrorIDZero	ErrorHandling_Itfs
25699	0x6463	MessageIDZero	ErrorHandling_Itfs
25700	0x6464	FBHasAlreadyAnSuperordinatedFB	ErrorHandling_Itfs
25701	0x6465	SubordinatedFBAlreadyAdded	ErrorHandling_Itfs
25702	0x6466	MessageIDisEqualToErrorID	ErrorHandling_Itfs
25703	0x6467	MessageIDEqualvocal	ErrorHandling_Itfs
25704	0x6468	CompletionOfAdditionalTextFailed	ErrorHandling_Itfs
25728	0x6480	MessageBufferFull	LoggingAdapter_Itfs
26112	0x6600	ConfigFileNotFound	AxisConfig_Itfs
26113	0x6601	ConfigFileNotOpened	AxisConfig_Itfs
26114	0x6602	ConfigFileNotClosed	AxisConfig_Itfs
26115	0x6603	ConfigDataNotRead	AxisConfig_Itfs
26116	0x6604	ConfigParameterNotFound	AxisConfig_Itfs
26117	0x6605	ConfigParameterNotValid	AxisConfig_Itfs
26208	0x6660	IDeviceHandlerBase_NotLinked	SoftwareLimitSwitch_Itfs
26209	0x6661	IUnitCalculationsUI_Basic_NotLinked	SoftwareLimitSwitch_Itfs
26210	0x6662	IConfigData_NotLinked	SoftwareLimitSwitch_Itfs
26211	0x6663	IConfigDataHandler_NotLinked	SoftwareLimitSwitch_Itfs
26212	0x6664	ReadConfigDataFailed	SoftwareLimitSwitch_Itfs
27136	0x6a00	DeviceError	DeviceAdapter_Itfs
27137	0x6a01	DeviceHandlerError	DeviceAdapter_Itfs
27168	0x6a20	eSEW_OSCHandler_InterfaceNotValid	DeviceAdapter_OSC71B
27169	0x6a21	eSEW_OSCHandler_TimeoutSendSync	DeviceAdapter_OSC71B
27170	0x6a22	eSEW_OSCHandler_InvalidCanId	DeviceAdapter_OSC71B
27171	0x6a23	eSEW_OSCHandler_InvalidInteface	DeviceAdapter_OSC71B
27172	0x6a24	eSEW_OSCHandler_Error	DeviceAdapter_OSC71B
27173	0x6a25	eSEW_OSCHandler_InvalidDataAddress	DeviceAdapter_OSC71B
27174	0x6a26	eSEW_OSCHandler_CanNotReady	DeviceAdapter_OSC71B
27175	0x6a27	eSEW_OSCHandler_DeviceNotReady	DeviceAdapter_OSC71B
27176	0x6a28	eSEW_OSCHandler_NotInitialized	DeviceAdapter_OSC71B
27177	0x6a29	eSEW_OSCHandler_CanIdAlreadyRegistered	DeviceAdapter_OSC71B
27178	0x6a2a	eSEW_OSCHandler_InvalidCanAddress	DeviceAdapter_OSC71B
27179	0x6a2b	eSEW_OSCHandler_NumberCanPDOToLarge	DeviceAdapter_OSC71B
27180	0x6a2c	eSEW_OSCHandler_TimeoutSendPDO	DeviceAdapter_OSC71B
27181	0x6a2d	eSEW_OSCHandler_TimeoutRequestSDO	DeviceAdapter_OSC71B

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
27182	0x6a2e	eSEW_OSCHandler_TimeoutHeartbeat	DeviceAdapter_OSC71B
27183	0x6a2f	eSEW_OSCHandler_EmergencyTelegram	DeviceAdapter_OSC71B
27184	0x6a30	eSEW_OSCHandler_SDOAbortCode	DeviceAdapter_OSC71B
27185	0x6a31	eSEW_OSCHandler_TimeoutRequestSBUSParam	DeviceAdapter_OSC71B
27186	0x6a32	eSEW_OSCHandler_MVLAbortCode	DeviceAdapter_OSC71B
27187	0x6a33	eSEW_OSCHandler_TimeoutMvlPD	DeviceAdapter_OSC71B
27188	0x6a34	eSEW_OSCHandler_InvalidPDLenght	DeviceAdapter_OSC71B
27189	0x6a35	eSEW_OSCHandler_InvalidPDSegment	DeviceAdapter_OSC71B
27712	0x6c40	eSEW_ExSourc_GetConfig	SyncExtSource_Itfs
27713	0x6c41	eSEW_ExSourc_NotLinked_SendObject	SyncExtSource_Itfs
27714	0x6c42	eSEW_ExSourc_ValueOutOfLimits	SyncExtSource_Itfs
27715	0x6c43	eSEW_ExSourc_VZ1Filter	SyncExtSource_Itfs
27716	0x6c44	eSEW_ExSourc_AverageFilter	SyncExtSource_Itfs
27717	0x6c45	eSEW_ExSourc_DeltaValueToLarge	SyncExtSource_Itfs
27718	0x6c46	eSEW_ExSourc_NotLinked_Persistent	SyncExtSource_Itfs
27719	0x6c47	eSEW_ExSourc_NotLinked_SyncExtSourceVa-lues	SyncExtSource_Itfs
27720	0x6c48	eSEW_ExSourc_NotLinked_ConfigData	SyncExtSource_Itfs
28224	0x6e40	eSEW_ParamHandler_Request	ParameterHandler
28225	0x6e41	eSEW_ParamHandler_Response	ParameterHandler
28226	0x6e42	eSEW_ParamHandler_NoDeviceLink	ParameterHandler
28672	0x7000	eSEW_FH_ASM_Result	FileHandler
28673	0x7001	eSEW_FH_TimeOut	FileHandler
28674	0x7002	eSEW_FH_FileNotHere	FileHandler
28675	0x7003	eSEW_FH_RTS_Result	FileHandler
28704	0x7020	eSEW_PLCGetInfo	Util
28705	0x7021	eSEW_DeltaValueToLarge	Util
28706	0x7022	eSEW_VZ1Filter	Util
28707	0x7023	eSEW_AverageFilter	Util
28708	0x7024	eSEW_ModuloMax_ModuloMin	Util
28709	0x7025	eSEW_ValueOutOfLimits	Util
28710	0x7026	eSEW_NotInitialized	Util
30208	0x7600	SoftwareLimitSwitchNotValid	InterpolationModes_Itfs
30209	0x7601	ModuloLimitsNotValid	InterpolationModes_Itfs
30210	0x7602	PresetPositionNotValid	InterpolationModes_Itfs
30211	0x7603	ReferenceOffsetNotValid	InterpolationModes_Itfs
30212	0x7604	ReferenceOffsetOutOfModuloLimit	InterpolationModes_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
30213	0x7605	HomingStartPositionNotValid	InterpolationModes_Itfs
30214	0x7606	ModuloModeNotSupported	InterpolationModes_Itfs
30215	0x7607	AxisNotReferenced	InterpolationModes_Itfs
30216	0x7608	TargetPositionNotValid	InterpolationModes_Itfs
30217	0x7609	TravelDistanceNotValid	InterpolationModes_Itfs
30218	0x760a	TargetPositionOutOfSoftwareLimitSwitch	InterpolationModes_Itfs
30219	0x760b	VelocityStopPositionNotValid	InterpolationModes_Itfs
30220	0x760c	MasterResolutionOutsideLimits	InterpolationModes_Itfs
30221	0x760d	MasterModuloOutsideLimits	InterpolationModes_Itfs
30222	0x760e	SlaveModuloOutsideLimits	InterpolationModes_Itfs
30223	0x760f	NumeratorDenominatorOutsideLimits	InterpolationModes_Itfs
30224	0x7610	MasterPositionOutsideLimits	InterpolationModes_Itfs
30225	0x7611	MasterTimeBaseOutsideLimits	InterpolationModes_Itfs
30226	0x7612	SlaveTimeBaseOutsideLimits	InterpolationModes_Itfs
30227	0x7613	SoftwareLimitPositive_Reached	InterpolationModes_Itfs
30228	0x7614	SoftwareLimitNegative_Reached	InterpolationModes_Itfs
30229	0x7615	ApplicationLimitDeceleration	InterpolationModes_Itfs
30230	0x7616	ApplicationLimitAcceleration	InterpolationModes_Itfs
30231	0x7617	ApplicationLimitVelocityPositive	InterpolationModes_Itfs
30232	0x7618	ApplicationLimitVelocityNegative	InterpolationModes_Itfs
30233	0x7619	InterfaceNotLinked	InterpolationModes_Itfs
30234	0x761a	ProfileGeneratorInternalError	InterpolationModes_Itfs
30235	0x761b	ReadConfigDataFailed	InterpolationModes_Itfs
30236	0x761c	InvalidLicence	InterpolationModes_Itfs
30237	0x761d	ContinueRelativeNotPossible	InterpolationModes_Itfs
30464	0x7700	InterfaceNotLinked	AntiSloshInterpolation
30465	0x7701	WrongInputParameter	AntiSloshInterpolation
30466	0x7702	ReadConfigDataFailed	AntiSloshInterpolation
30467	0x7703	InvalidConfigParameter	AntiSloshInterpolation
30495	0x771f	UndefinedError	AntiSloshInterpolation
30721	0x7801	OutOfLagErrorWindow	Controller_Itfs
30722	0x7802	EC_EncoderIsNotConnected	Controller_Itfs
30723	0x7803	PositionNotValid	Controller_Itfs
30724	0x7804	InterfaceError	Controller_Itfs
30725	0x7805	QueryFailed	Controller_Itfs
30726	0x7806	GearRatioIsZero	Controller_Itfs
30727	0x7807	IndexOutOfBounds	Controller_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
30728	0x7808	InvalidValueForControlloop	Controller_Itfs
30729	0x7809	InvalidValueForActValueEvaluation	Controller_Itfs
30730	0x780a	OutOfSkewErrorWindow	Controller_Itfs
30731	0x780b	WrongReferencedBitOnStatusWord	Controller_Itfs
30732	0x780c	WrongActualPositionSource	Controller_Itfs
30733	0x780d	ExternalEncoderActivatedOnMDD	Controller_Itfs
30734	0x780e	NoExternalEncoderSelected	Controller_Itfs
30735	0x780f	NoCombinedEncoderEvaluationSelected	Controller_Itfs
30736	0x7810	TorqueLevelingPGainMaxIsZero	Controller_Itfs
30737	0x7811	TooManyAssociatedAGMembers	Controller_Itfs
30738	0x7812	ConfirmTokenFailed	Controller_Itfs
30739	0x7813	InsufficientExternalEncoder	Controller_Itfs
30849	0x7881	InterfaceError	MultiAxisController_Itfs
30850	0x7882	NoAccessToAxes	MultiAxisController_Itfs
30851	0x7883	NoAccessReturnPossible	MultiAxisController_Itfs
30852	0x7884	IndexOutOfBounds	MultiAxisController_Itfs
30853	0x7885	WrongTravelTypeForReadjustment	MultiAxisController_Itfs
30854	0x7886	MoreThanTwoAxesForReadjustment	MultiAxisController_Itfs
30855	0x7887	WrongLSOperationForReadjustment	MultiAxisController_Itfs
30856	0x7888	QueryFailed	MultiAxisController_Itfs
30857	0x7889	ConfirmTokenFailed	MultiAxisController_Itfs
30858	0x788a	MissingCascadingLicense	MultiAxisController_Itfs
30859	0x788b	MissingFourAxesLicense	MultiAxisController_Itfs
30860	0x788c	MissingTorqueSkewingLicense	MultiAxisController_Itfs
30861	0x788d	WrongPriorityLicenseActivated	MultiAxisController_Itfs
30862	0x788e	ParameterChannelTimeOut	MultiAxisController_Itfs
30863	0x788f	NotAllAxesAreActivated	MultiAxisController_Itfs
30865	0x7891	HomingLSReversed	MultiAxisController_Itfs
30866	0x7892	HomingSafetyTimeExpired	MultiAxisController_Itfs
30867	0x7893	HomingSafetyDistancePassed	MultiAxisController_Itfs
30868	0x7894	HomingStartPositionNotValid	MultiAxisController_Itfs
30869	0x7895	TravelTypeDeactivated	MultiAxisController_Itfs
31232	0x7a00	InvalidInterface	AxisGroupBasic_Itfs
31233	0x7a01	NoAccessToAxes	AxisGroupBasic_Itfs
31234	0x7a02	NoAccessReturnPossible	AxisGroupBasic_Itfs
31235	0x7a03	UI_Init_Not_IBasic	AxisGroupBasic_Itfs
31236	0x7a04	NoAxesLinked	AxisGroupBasic_Itfs

26873346/DE – 07/2021

Dez	Hex	Fehler	Bibliothek SEW_MOS_...
31237	0x7a05	ArrayIndexOutOfBounds	AxisGroupBasic_Itfs
31238	0x7a06	MaxNumberOfAxesExceeded	AxisGroupBasic_Itfs
31264	0x7a20	InterfaceNotValid	AxisGroupPositioningInterpolated_Itfs
31265	0x7a21	OutOfBound_LinkingOfAxes	AxisGroupPositioningInterpolated_Itfs
31266	0x7a22	TouchprobeCounterOverflow	AxisGroupPositioningInterpolated_Itfs
31267	0x7a23	TouchprobeCounterTimeout	AxisGroupPositioningInterpolated_Itfs
31296	0x7a40	LSPositiveInverterConfigured	LimitSwitchEvaluation_Itfs
31297	0x7a41	LSNegativeInverterConfigured	LimitSwitchEvaluation_Itfs
31298	0x7a42	PositiveLSHit	LimitSwitchEvaluation_Itfs
31299	0x7a43	NegativeLSHit	LimitSwitchEvaluation_Itfs
31300	0x7a44	InterfaceError	LimitSwitchEvaluation_Itfs
31301	0x7a45	QueryFailed	LimitSwitchEvaluation_Itfs
31302	0x7a46	IndexOutOfBounds	LimitSwitchEvaluation_Itfs
31303	0x7a47	NoAxesConnected	LimitSwitchEvaluation_Itfs
31304	0x7a48	LimitSwitchReversed	LimitSwitchEvaluation_Itfs
31305	0x7a49	BothLSHit	LimitSwitchEvaluation_Itfs
36960	0x9060	ModeNotValid	ModeAdministrator

26873346/DE – 07/2021

15.2.4 Untergeordnete Softwaremodule Meldungen

Dez	Hex	Meldung	Bibliothek SEW_MOS_...
91200	0x16440	eSEW_LicMgr_RepMisSiLic	IECLicenseManager
91201	0x16441	eSEW_LicMgr_RepMisPerLic	IECLicenseManager
91202	0x16442	eSEW_LicMgr_RepMisRunLic	IECLicenseManager
91203	0x16443	eSEW_LicMgr_TrialLicenseActive	IECLicenseManager
91204	0x16444	eSEW_LicMgr_TrialLicenseExpired	IECLicenseManager
91205	0x16445	eSEW_LicMgr_LicenseActive	IECLicenseManager
91206	0x16446	eSEW_LicMgr_DualUseLicenseActive	IECLicenseManager
91207	0x16447	eSEW_LicMgr_NotTestableLicenseActive	IECLicenseManager
91208	0x16448	eSEW_LicMgr_RuntimeTrialLicenseActive	IECLicenseManager
91209	0x16449	eSEW_LicMgr_RuntimeTrialLicenseExpired	IECLicenseManager
91210	0x1644a	eSEW_LicMgr_TrialLicenseActivated	IECLicenseManager
91232	0x16460	InterfaceNotValid	ErrorHandling_Itfs
91233	0x16461	ErrorHandling_NotYetInitialized	ErrorHandling_Itfs
91264	0x16480	LoggingNotSuccessful	LoggingAdapter_Itfs
91265	0x16481	LogbookOpeningFailed	LoggingAdapter_Itfs
93248	0x16c40	eSEW_ExSourc_OffOnLimit	SyncExtSource_Itfs
94208	0x17000	eSEW_FH_BufferToShort	FileHandler
94209	0x17001	eSEW_FH_CancleJobNotAllowed	FileHandler
96257	0x17801	InvalidInverterType	Controller_Itfs
96258	0x17802	InterfaceCycleTimesNotConnected	Controller_Itfs
96800	0x17a20	TouchprobeInterfaceNotValid	AxisGroupPositioningInterpolated_Itfs
96801	0x17a21	HomingInterfaceNotValid	AxisGroupPositioningInterpolated_Itfs
96802	0x17a22	TouchprobeCounterOverflow	AxisGroupPositioningInterpolated_Itfs
103920	0x195f0	DynamicValueTooLarge	DeviceAdapter_Itfs
103921	0x195f1	DynamicValueTooSmall	DeviceAdapter_Itfs
103922	0x195f2	InverterWarning	DeviceAdapter_Itfs

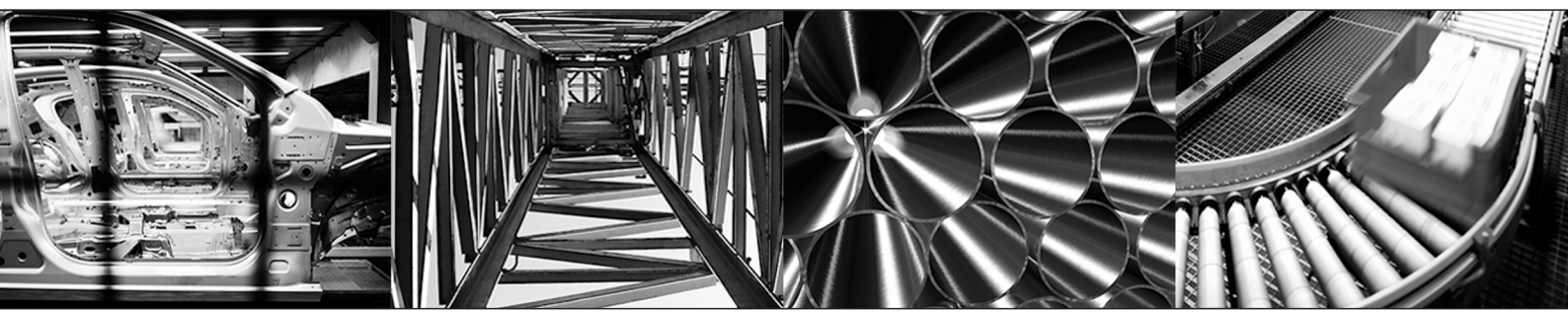
26873346/DE – 07/2021

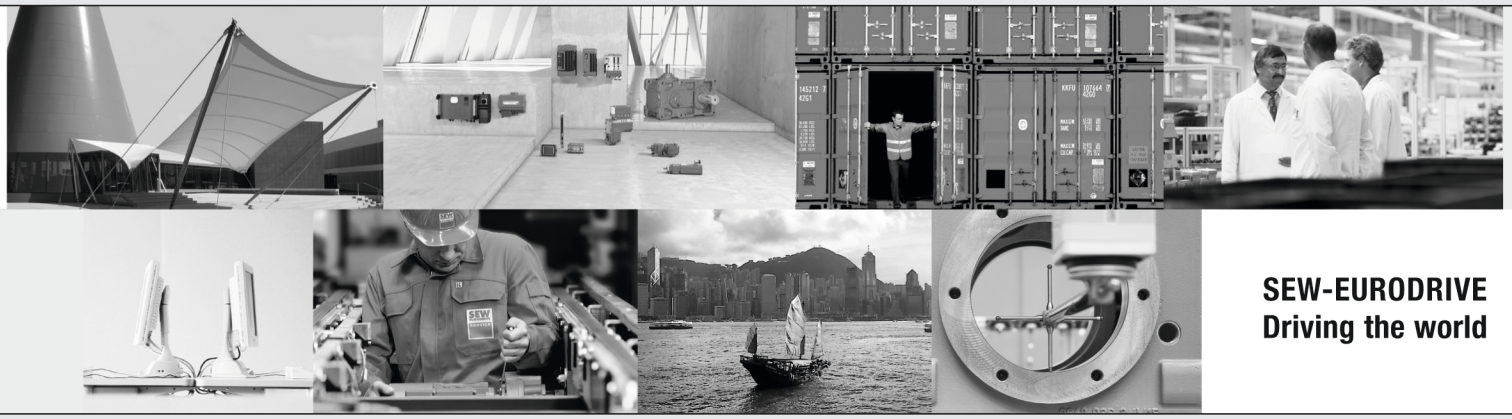
Stichwortverzeichnis

26873346/DE – 07/21

Z

26873346/DE – 07/21





SEW-EURODRIVE
Driving the world

SEW-EURODRIVE GmbH & Co KG
Ernst-Blickle-Str. 42
76646 BRUCHSAL
GERMANY
Tel. +49 7251 75-0
Fax +49 7251 75-1970
sew@sew-eurodrive.com
➞ **www.sew-eurodrive.com**