



ROBOTICS

Anwendungshandbuch

Externally Guided Motion



Trace back information:
Workspace 24B version a3
Checked in 2024-05-30
Skribenta version 5.5.019

Anwendungshandbuch

Externally Guided Motion

RobotWare 7.15

Dokumentnr: 3HAC073318-003

Revision: J

Die Informationen in diesem Handbuch können ohne vorherige Ankündigung geändert werden und stellen keine Verpflichtung von ABB dar. ABB übernimmt keinerlei Verantwortung für etwaige Fehler, die dieses Handbuch enthalten kann.

Wenn nicht ausdrücklich in vorliegendem Handbuch angegeben, gibt ABB für keine hierin enthaltenen Informationen Sachmängelhaftung oder Gewährleistung für Verluste, Personen- oder Sachschäden, Verwendbarkeit für einen bestimmten Zweck oder Ähnliches.

In keinem Fall kann ABB haftbar gemacht werden für Schäden oder Folgeschäden, die sich aus der Anwendung dieses Dokuments oder der darin beschriebenen Produkte ergeben.

Dieses Handbuch darf weder ganz noch teilweise ohne vorherige schriftliche Genehmigung von ABB vervielfältigt oder kopiert werden.

Zur späteren Verwendung aufbewahren.

Zusätzliche Kopien dieses Handbuchs können von ABB bezogen werden.

Übersetzung der Originalbetriebsanleitung.

Inhaltsverzeichnis

Überblick über dieses Handbuch	7
Produktdokumentation	9
Sicherheit	11
1 Einführung in Externally Guided Motion	13
1.1 Überblick	13
1.2 Einführung in EGM Position Stream	16
1.3 Einführung in EGM Position Guidance	17
1.4 Einführung in EGM Path Correction	20
2 Verwendung von Externally Guided Motion	21
2.1 Grundlegende Methode	21
2.2 Abarbeitungsstatus und Korrekturzustände	23
2.3 Eingangsdaten	25
2.4 Ausgangsdaten	28
2.5 Konfiguration	29
2.6 Koordinatensysteme	30
3 Das EGM-Sensorprotokoll	33
3.1 Überblick	33
3.1.1 EGM Sensorprotokollübersicht	33
3.1.2 Google Protocol Buffers	35
3.1.3 EGM Beschreibung des Sensorprotokolls	36
3.2 Aufbau eines EGM-Sensorkommunikationsendpunktes	40
3.3 Konfigurieren von UdpUc-Geräten	41
4 Systemparameter	43
4.1 Typ <i>External Motion Interface Data</i>	43
4.1.1 Der Typ External Motion Interface Data	43
4.1.2 Name	44
4.1.3 Level	45
4.1.4 Do Not Restart after Motors Off	46
4.1.5 Return to Program Position when Stopped	47
4.1.6 Default Ramp Time	48
4.1.7 Default Proportional Position Gain	49
4.1.8 Default Low Pass Filter Bandwidth	50
5 RAPID Referenzinformation	51
5.1 Instruktionen	51
5.1.1 EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition	51
5.1.2 EGMActMove - Vorbereiten einer EGM Bewegung mit Bahnkorrektur	56
5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel	58
5.1.4 EGMGetId - Erhält eine EGM Identität	64
5.1.5 EGMMoveC - Kreisförmige EGM-Bewegung mit Bahnkorrektur	65
5.1.6 EGMMoveL - Lineare EGM-Bewegung mit Bahnkorrektur	69
5.1.7 EGMReset - Zurücksetzen eines EGM Prozesses	73
5.1.8 EGMRunJoint - Ausführen einer EGM Bewegung mit einer Achswinkelposition ...	74
5.1.9 EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel	77
5.1.10 EGMSetupAI - Einrichtung analoger Eingangssignale für EGM	80
5.1.11 EGMSetupAO - Einrichtung analoger Ausgangssignale für EGM	83
5.1.12 EGMSetupGI - Einrichtung von Gruppeneingangssignalen für EGM	86
5.1.13 EGMSetupLTAPP - Einrichtung des LTAPP-Protokolls für EGM	89
5.1.14 EGMSetupUC - Einrichtung des UdpUc Protokolls für EGM	91
5.1.15 EGMStop - Anhalten einer EGM Bewegung	94
5.1.16 EGMStreamStart - Das EGM-Positions-Streaming starten.	96

5.1.17	EGMStreamStop - Das EGM-Positions-Streaming stoppen.	98
5.1.18	EGMWaitCond - Auf den EGM-Prozess warten.	99
5.2	Funktionen	101
5.2.1	EGMGetState - Erhält den gleichen EGM Zustand	101
5.3	Datentypen	102
5.3.1	egmframetype - Definiert Rahmentypen für EGM	102
5.3.2	egmident - Identifiziert einen spezifischen EGM Prozess	103
5.3.3	egm_minmax - Konvergenzkriterien für EGM	105
5.3.4	egmstate - Definiert den Zustand für EGM	106
5.3.5	egmcorrstate - Definiert den Korrekturzustand für EGM	107
5.3.6	egmstopmode - Definiert die Stopp-Modi für EGM	108
5.4	Codebeispiele	109
5.4.1	Verwendung des EGM-Positionsdatenstroms	109
5.4.2	Verwendung von EGM Position Guidance mit einem UdpUc-Gerät	112
5.4.3	Verwendung von EGM Position Guidance mit Signalen als Eingang	114
5.4.4	Verwendung von EGM Path Correction mit verschiedenen Protokolltypen	120
6	UdpUc Codebeispiele	123
Index		125

Überblick über dieses Handbuch

Über dieses Handbuch

Dieses Handbuch enthält Informationen über die RobotWare-Option Externally Guided Motion [3124-1], oft als EGM bezeichnet.

Verwendung

In diesem Handbuch erfahren Sie, was Externally Guided Motion ist und wie es verwendet wird. Außerdem bietet dieses Handbuch Informationen zu den RAPID-Komponenten und Systemparametern in Bezug auf Externally Guided Motion und Verwendungsbeispiele.

Wer sollte dieses Handbuch lesen?

Dieses Handbuch ist in erster Linie für Programmierer gedacht.

Voraussetzungen

Der Leser muss mit Folgendem vertraut sein:

- Industrieroboter und deren Terminologie
- Die RAPID-Programmiersprache
- Systemparameter und ihre Konfiguration

Referenzen

Referenz	Dokumentnummer
Anwendungshandbuch - Steuerungssoftware OmniCore	3HAC066554-003
Bedienungsanleitung - OmniCore	3HAC065036-003
Bedienungsanleitung - RobotStudio	3HAC032104-003
Technisches Referenzhandbuch - RAPID Overview	3HAC065040-003
Technisches Referenzhandbuch - RAPID Instruktionen, Funktionen und Datentypen	3HAC065038-003
Technisches Referenzhandbuch - Systemparameter	3HAC065041-003

Revisionen

Revision	Beschreibung
A	Veröffentlicht mit RobotWare 7.0.
B	Veröffentlicht mit RobotWare 7.0.1. <ul style="list-style-type: none"> • Veränderter Standardwert für den Parameter Default Ramp Time auf Seite 48.

Fortsetzung auf nächster Seite

Revision	Beschreibung
C	<p>Veröffentlicht mit RobotWare 7.1.</p> <ul style="list-style-type: none">• Beschreibung des <i>K-Faktors</i> (Standard der „Propotional Position Gain“) hinzugefügt in Abschnitt Eingangsdaten auf Seite 25.• Hinweis in Bezug auf eine unvorhersehbare Bewegung hinzugefügt in Abschnitt Default Low Pass Filter Bandwidth auf Seite 50.• Beschränkungen aktualisiert für die Instruktionen EGMRunJoint - Ausführen einer EGM Bewegung mit einer Achswinkelposition auf Seite 74 und EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel auf Seite 77.• Beschränkungen aktualisiert im Abschnitt Überblick auf Seite 13.• Argument <code>LATR</code> entfernt aus den Instruktionen <code>EGMSetupAI</code>, <code>EGMSetupAO</code> und <code>EGMSetupGI</code>. Es ist nicht länger verfügbar.• Neuen Setup-Abschnitt hinzugefügt in Einführung in EGM Position Guidance auf Seite 17.
D	<p>Veröffentlicht mit RobotWare 7.2.</p> <ul style="list-style-type: none">• Aktualisierung der Abschnitte Einführung in EGM Position Guidance auf Seite 17 und Konfigurieren von UdpUc-Geräten auf Seite 41 mit Informationen zum Systemparameter-Typ UDP Unicast Device.• Abschnitt Aufbau eines EGM-Sensorkommunikationsendpunktes auf Seite 40 aktualisiert.• Abschnitt Überblick auf Seite 33 aktualisiert mit Informationen zur EGM-Sensor-Protokolldatenstruktur.
E	<p>Veröffentlicht mit RobotWare 7.6.</p> <ul style="list-style-type: none">• Es wurden <code>moveIndex</code> und <code>CollisionIndex</code> zu EgmRobot auf Seite 37 hinzugefügt.• Der Positionsmodus in <i>EGM Position Guidance</i> unterstützt jetzt auch YuMi- und SCARA-Roboter.
F	<p>Veröffentlicht mit RobotWare 7.7.</p> <ul style="list-style-type: none">• <i>EGM Position Guidance</i> kann für 4-achsige Palletierungsroboter verwendet werden.• Aktualisierte Informationen zum Senden von RAPID-Daten und E/A-Signalen. <code>RAPIDfromRobot</code> und <code>RAPIDfromSensor</code> zu EgmRobot auf Seite 37 hinzugefügt. Datentyp <code>EGM_MAX_RAPID_DNUM</code> und Argumente zu den Instruktionen <code>EGMActJ</code> und <code>EGMActPose</code> hinzugefügt.• Informationen zur proportionalen Steuerungsverstärkung siehe Eingangsdaten auf Seite 25.
G	<p>Veröffentlicht mit RobotWare 7.8.</p> <ul style="list-style-type: none">• Pfad zur EGM-Ordner korrigiert in Das EGM-Sensorprotokoll auf Seite 33.
H	<p>Veröffentlicht mit RobotWare 7.13.</p> <ul style="list-style-type: none">• Geringfügige Korrekturen.
J	<p>Veröffentlicht mit RobotWare 7.15.</p> <ul style="list-style-type: none">• Aktualisierungen im Abschnitt Abarbeitungsstatus und Korrekturzustände auf Seite 23.• Hinzugefügter Abschnitt egmcorrstate - Definiert den Korrekturzustand für EGM auf Seite 107.

Produktdokumentation

Kategorien für Anwenderdokumentation von ABB Robotics

Die Anwenderdokumentation von ABB Robotics ist in mehrere Kategorien unterteilt. Die Liste beruht auf der Informationsart in den Dokumenten, unabhängig davon, ob es sich um Standardprodukte oder optionale Produkte handelt.



Tipp

Sie finden alle Dokumente über das myABB-Unternehmensportal www.abb.com/myABB.

Produkthandbücher

Manipulatoren, Steuerungen, DressPack und die meiste andere Hardware werden mit einem **Produkthandbuch** geliefert, das generell folgendes umfasst:

- Sicherheitsinformationen.
- Installation und Inbetriebnahme (Beschreibung der mechanischen Installation und der elektrischen Anschlüsse).
- Wartung (Beschreibung aller erforderlichen vorbeugenden Wartungsmaßnahmen einschließlich der entsprechenden Intervalle und der Lebensdauer der Teile).
- Reparatur (Beschreibung aller empfohlenen Reparaturvorgänge, einschließlich des Austauschs von Ersatzteilen).
- Kalibrierung.
- Fehlerbehebung.
- Stilllegung.
- Referenzinformation (Sicherheitsstandards, Einheitenumrechnung, Schraubverbindungen, Werkzeuglisten).
- Ersatzteilliste mit den entsprechenden Abbildungen (oder Referenzen zu separaten Ersatzteillisten).
- Referenzen zu den Schaltplänen.

Technische Referenzhandbücher

In den technischen Referenzhandbüchern werden die Referenzinformationen für Robotics-Produkte, wie Schmierung, RAPID-Sprache und Systemparameter, beschrieben.

Anwendungshandbücher

Bestimmte Anwendungen (z. B. Software- oder Hardware-Optionen) werden in **Anwendungshandbüchern** beschrieben. Ein Anwendungshandbuch kann eine oder mehrere Anwendungen beschreiben.

Ein Anwendungshandbuch enthält im Allgemeinen folgende Informationen:

- Zweck der Anwendung (Aufgabe und Nutzen).

Fortsetzung auf nächster Seite

- Enthaltenes Material (z. B. Kabel, E/A-Karten, RAPID-Instruktionen, Systemparameter, Software)
- Installieren von enthaltener oder erforderlicher Hardware.
- Bedienungsanleitung für die Anwendung.
- Beispiele für die Verwendung der Anwendung.

Bedienungsanleitungen

In den Bedienungsanleitungen wird die Handhabung der Produkte in der Praxis beschrieben. Diese Handbücher richten sich an die Personen, die direkten Bedienungskontakt mit dem Produkt haben, also Bediener der Produktionszelle, Programmierer und Wartungsmitarbeiter.

Sicherheit

Sicherheit des Personals

Der Roboter ist sehr schwer und übt unabhängig von seiner Geschwindigkeit eine extrem hohe Kraft aus. Auf eine Pause oder einen längeren Halt der Bewegung kann eine gefährliche, plötzliche Bewegung folgen. Selbst wenn ein Bewegungsmuster vorgegeben ist, kann ein externes Signal den Betrieb beeinflussen und eine unvorhergesehene Bewegung auslösen.

Daher ist es wichtig, beim Betreten von abgesicherten Räumen alle Sicherheitsbestimmungen einzuhalten.



WARNUNG

Programmänderungen sollten vor der Übergabe in die Produktion immer validiert und getestet werden, um Menschen und Eigentum zu schützen. Stellen Sie sicher, dass es möglich ist, den Roboter mit einer Schutzvorrichtung anzuhalten.

Sicherheitsbestimmungen

Vor dem ersten Einsatz des Roboters müssen Sie sich unbedingt mit den Sicherheitsbestimmungen im Handbuch *Sicherheitshandbuch für den Roboter - Manipulator und IRC5 oder OmniCore-Steuerung* vertraut machen.

Diese Seite wurde absichtlich leer gelassen

1 Einführung in Externally Guided Motion

1.1 Überblick

Zweck

Externally Guided Motion (EGM) bietet die folgenden drei Merkmale:

- *EGM Position Stream:*
Die aktuellen und geplanten Positionen der mechanischen Einheiten werden in einer RAPID-Task an ein externes Gerät gesendet.
- *EGM Position Guidance:*
Der Roboter folgt nicht der programmierten Bahn in RAPID sondern einer Bahn, die von einem externen Gerät erzeugt wurde.
- *EGM Path Correction:*
Die programmierte Roboterbahn wird durch Messungen eines externen Geräts geändert/korrigiert.

EGM Position Stream

Der Zweck von EGM Position Stream besteht darin, externe Geräte mit den aktuellen und geplanten Positionen von mechanischen Einheiten auszustatten, die von der Robotersteuerung gesteuert werden.

Einige Anwendungsbeispiele sind:

- Laserschweißen, bei dem der Laserkopf den Laserstrahl dynamisch steuert.
- Alle Roboter-gemounteten Geräte, die den „Roboter“-TCP mit einer externen Steuerung steuern.

EGM Position Guidance

Der Zweck von *EGM Position Guidance* ist die Verwendung eines externen Geräts, um Positionsdaten für einen oder mehrere Roboter zu generieren. Die Roboter werden an diese vorgegebene Position bewegt.

Anwendungsbeispiele sind:

- Legen Sie ein Objekt (z. B. Autotür oder -fenster) an einen Ort (z. B. Karosserie), der von einem externen Sensor bestimmt wurde.
- Griff in die Kiste. Nehmen Sie mithilfe eines externen Sensors Objekte aus einer Kiste, um das Objekt und seine Position zu bestimmen.

EGM Path Correction

Der Zweck von *EGM Path Correction* ist die Verwendung von auf dem Roboter montierten externen Geräten, um Bahnkorrekturdaten für einen oder mehrere Roboter zu generieren. Die Roboter werden an der korrigierten Bahn entlang bewegt, bei der es sich um die programmierte Bahn mit zusätzlich gemessenen Korrekturen handelt.

Anwendungsbeispiele sind:

- Nahtverfolgung.
- Verfolgung von Objekten, die sich in der Nähe einer bekannten Bahn bewegen.

Fortsetzung auf nächster Seite

1 Einführung in Externally Guided Motion

1.1 Überblick

Fortsetzung

Enthaltene Komponenten

Die RobotWare-Option *Externally Guided Motion* bietet Ihnen Folgendes:

- Instruktionen zum Starten und Stoppen von EGM Position Stream.
- Instruktionen zum Einrichten, Aktivieren und Zurücksetzen von EGM Position Guidance.
- Instruktionen zum Einrichten, Aktivieren und Zurücksetzen von EGM Path Correction.
- Instruktionen zum Initiieren von EGM Position Guidance-Bewegungen, mit der Ausführung von RAPID synchronisiert oder nicht, und um diese zu stoppen.
- Instruktionen zum Durchführen von EGM Path Correction-Bewegungen.
- Eine Funktion zum Abrufen des aktuellen EGM-Status.
- Systemparameter zum Konfigurieren von EGM und zum Einstellen der Standardwerte.

Einschränkungen

EGM unterstützt kein koordiniertes MultiMove.

Einschränkungen für EGM Position Stream

- EGM Position Stream ist nur mit der UdpUc-Kommunikation verfügbar.
- Werkzeugdaten und Ladedaten können während eines aktiven Positionsdatenstroms nicht dynamisch geändert werden.
- Es ist nicht möglich, Positionen koordinierter MultiMove-Systeme zu streamen.
- Absolute Accuracy wird nicht unterstützt, wenn das Streaming mit `EGMStreamStart` gestartet wird. Es wird jedoch unterstützt, wenn es mit `EGMActXXX\StreamStart` gestartet wird.
- EGM Position Stream ist nicht mit EGM Path Correction kompatibel.
- Mechanische Einheiten dürfen nicht aktiviert oder deaktiviert werden, wenn EGM Position Stream aktiv ist.

Einschränkungen für EGM Position Guidance

- Muss in einem exakten Punkt starten und enden.
- Die erste Bewegung, die nach einem Neustart der Steuerung ausgeführt wird, kann keine EGM-Bewegung sein.
- Der Positionsmodus unterstützt 6-Achsen-Roboter, 4-Achsen-Palletierroboter, YuMi-Roboter und SCARA-Roboter.
- Es ist nicht möglich, Linearbewegungen mit EGM Position Guidance auszuführen, da EGM Position Guidance keine Interpolatorfunktionalität enthält. Die tatsächliche Bahn des Roboters hängt von der Roboterkonfiguration, der Startposition und den erzeugten Positionsdaten ab.
- EGM Position Guidance unterstützt kein koordiniertes MultiMove.
- Es gibt eine Beschränkung auf eine mechanische Einheit pro Bewegungstask.

Fortsetzung auf nächster Seite

- Es ist nicht möglich, EGM Position Guidance zu verwenden, um eine mechanische Einheit in einem sich bewegenden Werkobjekt zu führen.
- Wenn der Roboter in die Nähe einer Singularität gelangt, d. h., wenn zwei Roboterachsen fast parallel sind, wird die Roboterbewegung mit einer Fehlermeldung angehalten. In dieser Situation kann der Roboter nur aus der Singularität heraus verschoben werden.
- Wenn EGM aktiv ist, kann sich Motion Supervision anders verhalten, als während normalen Bewegungen. Die empfohlene Aktion nach einer Kollision ist die Deaktivierung von EGM und das Starten der EGM-Sequenz von Beginn an.

Einschränkungen für EGM Path Correction

- Unterstützt nur 6-achsige Roboter.
- Muss in einem exakten Punkt starten und enden.
- Das externe Gerät muss auf dem Roboter montiert sein.
- Korrekturen können nur im Bahn-Koordinatensystem angewendet werden.
- Es können nur Positionskorrekturen in Y und Z durchgeführt werden. Es ist nicht möglich, Ausrichtungskorrekturen oder Korrekturen in X (der Bahnrichtung/Tangente) durchzuführen.
- Wenn EGM aktiv ist, kann sich Motion Supervision anders verhalten, als während normalen Bewegungen. Die empfohlene Aktion nach einer Kollision ist die Deaktivierung von EGM und das Starten der EGM-Sequenz von Beginn an.

1 Einführung in Externally Guided Motion

1.2 Einführung in EGM Position Stream

1.2 Einführung in EGM Position Stream

Was ist EGM Positionsdatenstrom

EGM Position Stream ist nur für UdpUc-Kommunikation verfügbar. Es bietet die Möglichkeit, geplante und tatsächliche Positionsdaten der mechanischen Einheit (z. B. Roboter, Positionierer, Verfahreinheit etc.) von der Robotersteuerung in regelmäßigen Abständen zu senden. Der Nachrichteninhalt wird von Google Protobuf-Definitionsdatei *egm.proto* angegeben. Der zyklische Kommunikationskanal (UDP) kann über die High-Priority-Netzwerkumgebung der Robotersteuerung ausgeführt werden, die einen stabilen Datenaustausch bis zu 250 Hz sicherstellt. Für jeden Bewegungstask muss ein Kommunikationskanal vorhanden sein.

EGM Position Stream kann zusammen mit EGM Position Guidance genutzt werden.

1.3 Einführung in EGM Position Guidance

Was ist EGM Position Guidance

EGM Position Guidance wurde für fortgeschrittene Benutzer entwickelt und bietet eine untergeordnete Schnittstelle zur Robotersteuerung, indem die Bahnplanung, die verwendet werden kann, wenn schnell reagierende Roboterbewegungen benötigt werden, umgangen wird. EGM Position Guidance kann zum schnellen Lesen aus und Schreiben in Positionen des Bewegungssystems verwendet werden. Dies kann je nach Robotertyp alle 4 ms mit einer Steuerungsverzögerung von 10 bis 20 ms erfolgen. Die Referenzen können entweder durch die Verwendung von Achsenwerten oder einer Position festgelegt werden. Die Position kann in jedem Werkobjekt, das während der EGM Position Guidance-Bewegung nicht bewegt wird, definiert werden.



Hinweis

In allen weiteren Beschreibungen von EGM ist die tatsächliche Abtastzeit auf einem realen Robotersystem 4,032 ms und circa 4 ms auf einem virtuellen Robotersystem.

Das notwendige Filtern, die Überwachung von Referenzen und die Statusbehandlung wird von EGM Position Guidance behandelt. Beispiele der Statusbehandlung sind Programmstart/-stopp, Not-Halt usw.

Der Hauptvorteil von EGM Position Guidance ist die Schnelligkeit und die niedrige Verzögerung/Wartezeit im Vergleich zu anderen Mitteln der externen Bewegungssteuerung. Die Zeit zwischen dem Schreiben einer neuen Position, bis diese angegebene Position beginnt, die tatsächliche Roboterposition zu beeinflussen, beträgt normalerweise etwa 20 ms.

EGM behandelt *Absolute Accuracy*.

EGM Position Guidance kann mit logischen Einstellungen (E/A, etc. einstellen) kombiniert werden oder durch Aktivierung anderer Steuermodi, z. B. Verwendung der Force Control-Montageinstruktionen. Dies kann durch die Verwendung eines optionalen Arguments erzielt werden, um nicht auf EGM-Konvergenz (\NoWaitCond) zu warten.

Was EGM Position Guidance nicht macht

EGM geht direkt zur Erstellung der Motorreferenz, d. h. es bietet keine Bahnplanung. Dies bedeutet, dass Sie eine Bewegung nicht zu einem Positionsziel zuordnen können und eine lineare Bewegung erwarten können. Es ist nicht möglich, entweder eine Bewegung einer bestimmten Geschwindigkeit oder eine Bewegung, die eine bestimmte Zeit in Anspruch nehmen soll, zuzuordnen.

Fortsetzung auf nächster Seite

1 Einführung in Externally Guided Motion

1.3 Einführung in EGM Position Guidance

Fortsetzung

Für das Anordnen dieser Bewegungen wird die Bahnplanung benötigt. Wir verweisen Sie auf die Standardbewegungsanweisungen in RAPID, d. h. MoveL, MoveJ usw.

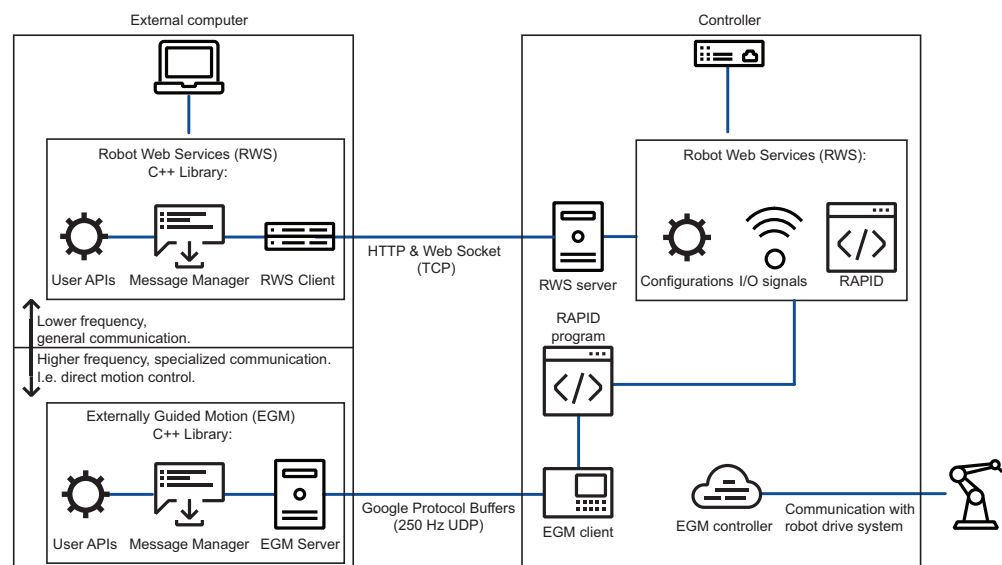


WARNUNG

Da die Bahnplanung von EGM in der Robotersteuerung umgangen wird, wird die Roboterbahn direkt von der Benutzereingabe erstellt. Daher muss sichergestellt werden, dass der Fluss von Positionsreferenzen, der an die Steuerung gesendet wird, so gleichmäßig wie möglich ist. Der Roboter reagiert schnell auf alle Positionsreferenzen, die an die Steuerung gesendet werden, auch auf falsche Referenzen.

Einrichtung von EGM Position Guidance

Wenn EGM Position Guidance richtig eingerichtet ist, kommuniziert das externe Gerät über ein Protokoll mit der Steuerung und steuert auf diese Weise die Bewegung des Roboters. Die folgende Abbildung zeigt die schematische Übersicht der möglichen Einrichtung einer EGM-Anwendung:



Damit diese funktioniert, müssen die folgenden Schritte ausgeführt werden:

- 1 Richten Sie Ihr RobotWare-System einschließlich der RobotWare-Option *Externally Guided Motion* ein.



Hinweis

Weitere Informationen dazu welche Robotertypen und Optionen in das Setup eingeschlossen werden können finden Sie in [Einschränkungen auf Seite 14](#).

- 2 Richten Sie Ihr UdpUc-Gerät so mit einem EGM-Sensorprotokoll ein, dass es mit der Robotersteuerung kommunizieren kann. Siehe [Das](#)

Fortsetzung auf nächster Seite

[EGM-Sensorprotokoll auf Seite 33](#) für grundlegende Anweisungen zum Aufbau eines EGM-Sensorkommunikationsendpunktes.



Hinweis

Codebeispiele sind verfügbar in der RobotWare-Distribution. Siehe [UdpUc Codebeispiele auf Seite 123](#).

- 3 Richten Sie den RAPID-Code mit Definition der Details der EGM Position Guidance-Funktionalität ein. Siehe [Grundlegende Methode auf Seite 21](#) für eine Beschreibung der im Code benötigten Basiselemente sowie [Verwendung von EGM Position Guidance mit einem UdpUc-Gerät auf Seite 112](#) für ein vollständiges Beispiel des RAPID-Codes.
- 4 Das Gerät, das die Eingabedaten für EGM bereitstellt, muss als UDP Unicast Device (UdpUc)-Gerät konfiguriert sein. Diese Konfiguration wird in RobotStudio mit den Systemparametern vom Typ *UDP Unicast Device* in der Parametergruppe *Communication* vorgenommen. Definieren Sie die IP-Adresse des Gerätes und legen Sie das Übertragungsprotokoll mit UDPUC fest. Siehe [Konfigurieren von UdpUc-Geräten auf Seite 41](#).



Hinweis

Nach dieser Konfigurationsänderung muss die Steuerung neu gestartet werden. Das Gerät kann nach dem Neustart von EGM verwendet werden, um einen Roboter zu führen.

1 Einführung in Externally Guided Motion

1.4 Einführung in EGM Path Correction

1.4 Einführung in EGM Path Correction

Was ist EGM Path Correction

EGM Path Correction ermöglicht es dem Benutzer, eine programmierte Roboterbahn zu korrigieren. Das Gerät oder der Sensor, das/der zum Messen der tatsächlichen Bahn verwendet wird, muss auf dem Werkzeugflansch montiert sein und es muss möglich sein, das Sensorkoordinatensystem zu kalibrieren.

Die Korrekturen werden im Bahn-Koordinatensystem durchgeführt, das seine X-Achse von der Tangente der Bahn erhält, die Y-Achse ist ein Kreuzprodukt der Bahntangente und die Z-Richtung des aktiven Werkzeug-Koordinatensystem und die Z-Achse ist das Kreuzprodukt der X-Achse und der Y-Achse.

EGM Path correction muss an einem Feinpunkt beginnen und enden. Die Sensormessungen können mit einer Vielfachen von 48 ms bereitgestellt werden.

2 Verwendung vonExternally Guided Motion

2.1 Grundlegende Methode

Grundlegende Methode für EGM Position Stream

EGM Position Stream ist verfügbar, wenn UdpUc für die Kommunikation mit dem externen Gerät verwendet wird. EGM Position Stream kann auf zwei verschiedene Arten gestartet werden. Eine besteht darin, `EGMStreamStart` zu verwenden, die andere, `EGMActJoint\StreamStart` oder `EGMActPose\StreamStart`. EGM Position Stream wird automatisch durch `EGMStop`, `EGMReset` und wenn eine `EGMRunJoint` oder `EGMRunPose` Instruktion abgeschlossen ist, gestoppt. Es existiert auch eine spezielle Instruktion, `EGMStreamStop`, um den Datenstrom zu stoppen.

Positions-Streaming unterstützt keine dynamische Änderung von Werkzeug oder Last. Wenn `EGMStreamStart` zum Starten des Positionsdatenstroms verwendet wird, werden das aktive Werkzeug und die Last an die Steuerung übergeben. Wenn `EGMActJoint` oder `EGMActPose` verwendet wird, werden das aktive Werkzeug und die Last (oder, wenn angegeben, das angegebene Werkzeug und/oder die Last) an die Steuerung übergeben. Diese Werkzeug- und Lastdaten werden dann von EGM zur Berechnung von Positionen verwendet, bis der Positionsdatenstrom gestoppt wird. Für jeden Bewegungstask muss ein separater Positionsdatenstrom gestartet werden.

	Aktion
1	Registrieren Sie einen EGM-Client und erhalten Sie eine EGM-Identität. Diese Identität wird dann für die Verknüpfung der Einrichtung, Aktivierung, Bewegung, Deaktivierung usw. für einen bestimmten EGM-Zweck verwendet. Der EGM-Zustand ist immer noch <code>EGM_STATE_CONNECTED</code> .
2	Rufen Sie die EGM Einrichtungsinstruktion <code>EGMSetupUC</code> auf, um die externe Ausrüstung mithilfe der Protokollverbindung UdpUc einzurichten.
3	Entweder: A Starten Sie den Positionsdatenstrom mit der Instruktion <code>EGMStreamStart</code> . B Starten Sie den Positionsdatenstrom mit <code>EGMActJoint</code> oder <code>EGMActPose</code> mit dem optionalen Argument <code>\StreamStart</code> . Der EGM-Zustand lautet nun <code>EGM_STATE_RUNNING</code> .
4	EGM Position Stream ist aktiv und sendet die tatsächliche und geplante Position, bis er gestoppt wird.
5	A Bei Start mit <code>EGMStreamStart</code> : Den Positionsdatenstrom mit <code>EGMStreamStop</code> stoppen. B Bei Start mit <code>EGMActJoint</code> oder <code>EGMActPose</code> : Stoppen Sie den Positionsdatenstrom mit <code>EGMStop</code> oder <code>EGMReset</code> . Der EGM-Zustand wechselt zurück zu <code>EGM_STATE_CONNECTED</code> .

Fortsetzung auf nächster Seite

2 Verwendung von Externally Guided Motion

2.1 Grundlegende Methode

Fortsetzung

Grundlegende Methode für EGM Position Guidance

Dies ist die allgemeine Methode, um einen Roboter mithilfe eines externen Geräts (Sensor) zu bewegen bzw. zu führen, um das Ziel für die Bewegung vorzugeben.

	Aktion
1	Verschieben Sie den Roboter an einen Feinpunkt.
2	Registrieren Sie einen EGM-Client und erhalten Sie eine EGM-Identität. Diese Identität wird dann für die Verknüpfung der Einrichtung, Aktivierung, Bewegung, Deaktivierung usw. für einen bestimmten EGM-Zweck verwendet. Der EGM-Zustand ist immer noch EGM_STATE_CONNECTED.
3	Eine EGM-Einrichtungsinstruktion für die Einrichtung der Positionsdatenquelle mithilfe von Signalen oder der Protokollverbindung UdpUc aufrufen.
4	Wählen Sie aus, ob die Position als Achsenwerte oder als eine Position angegeben werden soll, und geben Sie die Konvergenzkriterien für die Position an, d. h. wann die Position als erreicht betrachtet wird.
5	Wenn die Position ausgewählt wurde, definieren Sie, welche Koordinatensysteme verwendet werden, um die Zielposition zu definieren und auf welches Koordinatensystem die Bewegung angewendet werden soll.
6	Geben Sie dem Stopp-Modus ein optionales Timeout, und lassen Sie ihn die Bewegung selbst ausführen. Jetzt ist der EGM-Status EGM_STATE_RUNNING. Dies ist der Fall, wenn sich der Roboter bewegt.
7	Die EGM-Bewegung stoppt, wenn die Position als erreicht betrachtet wird, d. h. die Konvergenzkriterien wurden erfüllt. Jetzt ist der EGM-Status wieder EGM_STATE_CONNECTED.

Grundlegende Methode für EGM Path Correction

Dies ist die allgemeine Methode für die Korrektur einer programmierten Bahn mit EGM Path Correction.

	Aktion
1	Verschieben Sie den Roboter an einen Feinpunkt.
2	Registrieren Sie einen EGM-Client und erhalten Sie eine EGM-Identität. Diese Identität wird dann für die Verknüpfung der Einrichtung, Aktivierung, Bewegung, Deaktivierung usw. für einen bestimmten EGM-Zweck verwendet. Der EGM-Zustand ist immer noch EGM_STATE_CONNECTED.
3	Eine EGM-Einrichtungsinstruktion für die Einrichtung der Positionsdatenquelle mithilfe von Signalen oder der Protokollverbindung UdpUc aufrufen.
4	Definieren Sie das Sensor-Korrekturkoordinatensystem, das immer ein Werkzeug-Koordinatensystem ist.
5	Führen Sie die Bewegung selbst aus. Der EGM-Status ist jetzt EGM_STATE_RUNNING.
	Am Endpunkt kehrt EGM in den Zustand EGM_STATE_CONNECTED zurück.
6	Um eine EGM-Identität zur Verwendung mit einem anderen Sensor freizugeben, müssen Sie EGM zurücksetzen, wodurch EGM auf den Status EGM_STATE_DISCONNECTED zurückgesetzt wird.

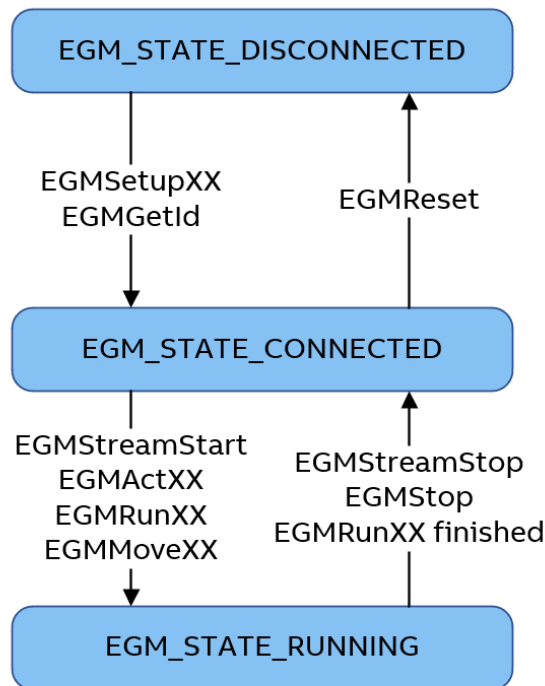
2.2 Abarbeitungsstatus und Korrekturzustände

Beschreibung der Abarbeitungsstatus

Dies sind die Abarbeitungsstatus von EGM auf der Robotersteuerung:

Wert	Beschreibung
EGM_STATE_DISCONNECTED	Der EGM Zustand des spezifischen Prozesses ist undefiniert. Es ist keine EGM Korrektur verbunden/aktiv.
EGM_STATE_CONNECTED	Die Einrichtung wurde vorgenommen, aber es ist keine EGM-Bewegung aktiv.
EGM_STATE_RUNNING	Der spezifizierte EGM Prozess wird bereits ausgeführt. Die EGM-Bewegung ist aktiv, d. h. der Roboter wird bewegt.

Die Übergänge zwischen den verschiedenen Status entsprechen der untenstehenden Abbildung.



xx1400001082

Die RAPID-Instruktionen `EGMRunJoint` und `EGMRunPose` ändern den Zustand von `EGM_STATE_CONNECTED` zu `EGM_STATE_RUNNING`, wenn die Konvergenzkriterien für die Zielposition nicht erfüllt wurden und die Timeout-Zeit nicht abgelaufen ist. Wenn eine dieser Bedingungen zutrifft, ändern sich der EGM-Zustand erneut zu `EGM_STATE_CONNECTED` und die Instruktionen endet. D. h., dass die RAPID-Abarbeitung bis zur nächsten Instruktion weiter ausgeführt wird.

Wenn EGM den Status `EGM_STATE_RUNNING` hat und die RAPID-Ausführung gestoppt wird, tritt EGM in den Status `EGM_STATE_CONNECTED`. Beim Programm-Neustart kehrt EGM in den Status `EGM_STATE_RUNNING` zurück.

Fortsetzung auf nächster Seite

2 Verwendung von Externally Guided Motion

2.2 Abarbeitungsstatus und Korrekturzustände

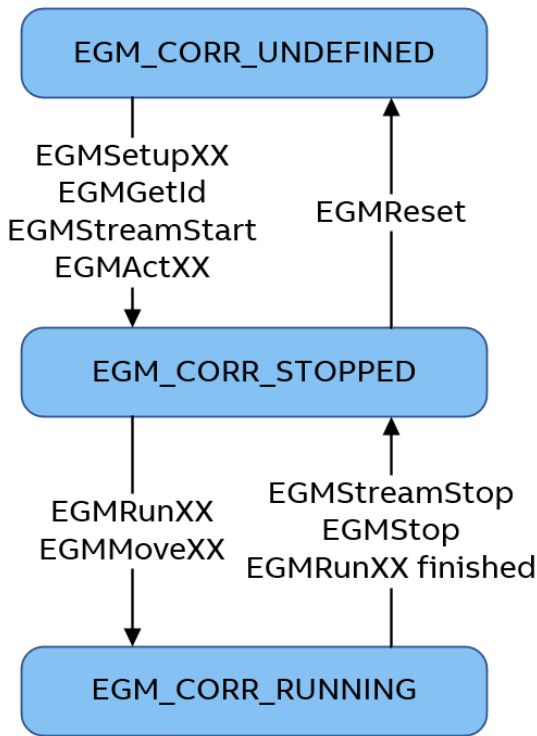
Fortsetzung

Wenn der Programmzeiger unter Verwendung von **PZ --> Main** oder **PZ --> Cursor** bewegt wird, wird der Status **EGM** auf **EGM_STATE_CONNECTED** geändert, wenn der Status **EGM_STATE_RUNNING** war.

Beschreibung der Korrekturzustände

Dies sind die Korrekturzustände von EGM auf der Robotersteuerung:

Wert	Beschreibung
EGM_CORR_UNDEFINED	Der EGM-Korrekturzustand des spezifischen Prozesses ist undefiniert. EGMGetId wurde nicht ausgeführt oder EGM wurde zurück-gesetzt.
EGM_CORR_STOPPED	Dieser Korrekturzustand wird eingenommen, sobald EGMGetId ausgeführt wird, aber noch keine EGM Bewegung aktiv ist.
EGM_CORR_RUNNING	Der angegebene Prozess EGM wird abgearbeitet, d. h. der Roboter wird bewegt.
EGM_CORR_ERROR	Ein Fehler bei der Positionsführung oder Bahnkorrektur ist aufgetreten, während der Prozess EGM abgearbeitet wurde.



xx2400000640

Um festzustellen, ob das EGM-Streaming gestartet wurde, überprüfen Sie, ob der EGM-Zustand **EGM_STATE_RUNNING** ist und der Korrekturzustand **EGM_STATE_STOPPED** oder **EGM_STATE_RUNNING**.

2.3 Eingangsdaten

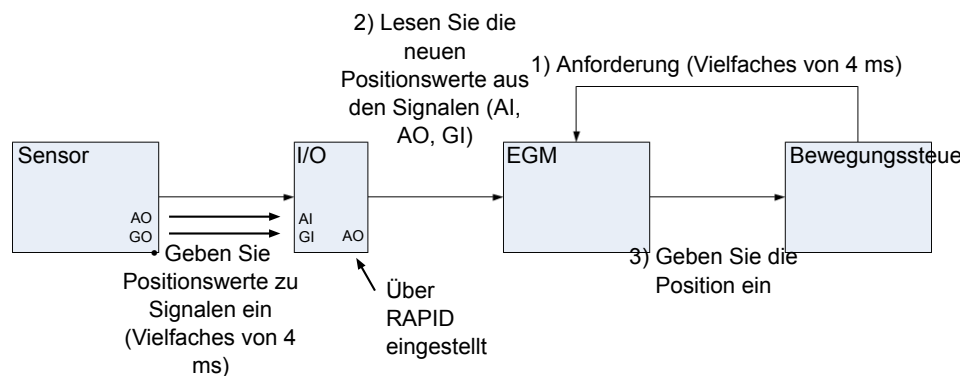
Eingangsdaten für EGM Position Guidance

Die Quelle für die Eingangsdaten wird mithilfe der EGM-Einrichtungsinstruktionen ausgewählt. Die drei ersten Instruktionen wählen eine Signalschnittstelle und die letzte Instruktion eine UdpUc-Schnittstelle aus (*User Datagram Protocol Unicast Communication*).

Instruktionen	Beschreibung
EGMSetupAI	Einrichtung analoger Eingangssignale für EGM
EGMSetupAO	Einrichtung analoger Ausgangssignale für EGM
EGMSetupGI	Einrichtung von Gruppeneingangssignalen für EGM
EGMSetupUC	Einrichten des UdpUc-Protokolls für EGM

Eingangsdaten für EGM enthalten hauptsächlich Positionsdaten entweder als Achsen oder als Position, d. h. kartesische Position plus Ausrichtung.

Nachfolgend wird der Datenfluss für die Signalschnittstelle dargestellt:



xx1400002016

- 1 Die Bewegungssteuerung ruft EGM auf.
 - 2 EGM liest die Positionswerte aus den Signalen.
 - 3 EGM schreibt die Positionsdaten in die Bewegungssteuerung.
- Der Sensor schreibt die Positionsdaten in die Signale.

Wenn Signale als Datenquelle verwendet werden, ist der Eingang auf 6 für den Roboter begrenzt, d. h. 6 Achsenwerte oder 3 kartesische Positionswerte (x, y, z) plus 3 Eulersche Winkelwerte (rx, ry, rz) und bis zu 6 Werte für zusätzliche Achsen. Bei Verwendung des EGM-Achsenmodus bei einem 7-achsigen Roboter liefert der Eingang der ersten zusätzlichen Achse die Position für die zusätzlichen Roboterachsen.

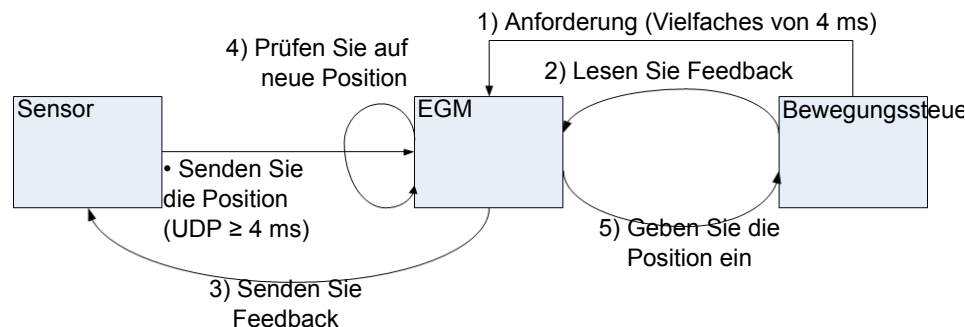
Fortsetzung auf nächster Seite

2 Verwendung von Externally Guided Motion

2.3 Eingangsdaten

Fortsetzung

Der Datenfluss für die UdpUc-Schnittstelle wird nachfolgend dargestellt:



xx1400002017

- 1 Die Bewegungssteuerung ruft EGM auf.
 - 2 EGM liest die Feedbackdaten aus der Bewegungssteuerung.
 - 3 EGM sendet die Feedbackdaten an den Sensor.
 - 4 EGM überprüft die UDP-Warteschlange auf Meldungen vom Sensor.
 - 5 Wenn eine Meldung vorliegt, liest EGM die nächste Meldung und Schritt 5 schreibt die Positionsdaten in die Bewegungssteuerung. Wenn keine Positionsdaten gesendet wurden, fährt die Bewegungssteuerung mit der Verwendung der letzten Positionsdaten, die zuvor von EGM geschrieben wurden, fort.
- Der Sensor sendet Positionsdaten an die Steuerung (EGM). Wir empfehlen, dies mit Schritt 3 zu verbinden. Dann befindet sich der Sensor in Phase mit der Steuerung.

Die Steuerungsschleife beruht auf dem folgenden Verhältnis zwischen Geschwindigkeit und Position:

$speed = k * (pos_ref - pos) + speed_ref$	<p>k - Faktor (Standard der „Proportional Position Gain“)</p> <p>pos_ref - Referenzposition</p> <p>pos - gewünschte Position</p> <p>$speed_ref$ - Referenzgeschwindigkeit</p>
---	---

Instruktionen zur Implementierung des UdpUc-Protokolls für ein externes Gerät finden Sie unter [Das EGM-Sensorprotokoll auf Seite 33](#). Dort finden Sie außerdem eine Beschreibung der Eingangsdaten.

Proportionale Steuerungsverstärkung k

Die Proportionale Steuerungsverstärkung k wird zur Berechnung einer Geschwindigkeitskorrektur verwendet, mit der die Roboterposition in Richtung der Referenzposition pos_ref gesteuert wird.

Die Eingabe $speed_ref$ ist für die Vorsteuerung einer gewünschten Geschwindigkeit gedacht, z. B. bei der Verfolgung eines beweglichen Ziels oder einer zeitlich veränderlichen Referenzposition. In diesen Fällen sollte die Eingabe $speed_ref$

Fortsetzung auf nächster Seite

der gewünschten aktuellen Geschwindigkeit des beweglichen Ziels oder der zeitlich veränderlichen Referenzposition entsprechen.

Der resultierende Wert Drehzahl ist die resultierende Geschwindigkeit, die intern von der Robotersteuerung zur Berechnung der Achsenreferenzen verwendet wird, die an die untergeordnete Servo-Steuerung gesendet wird.

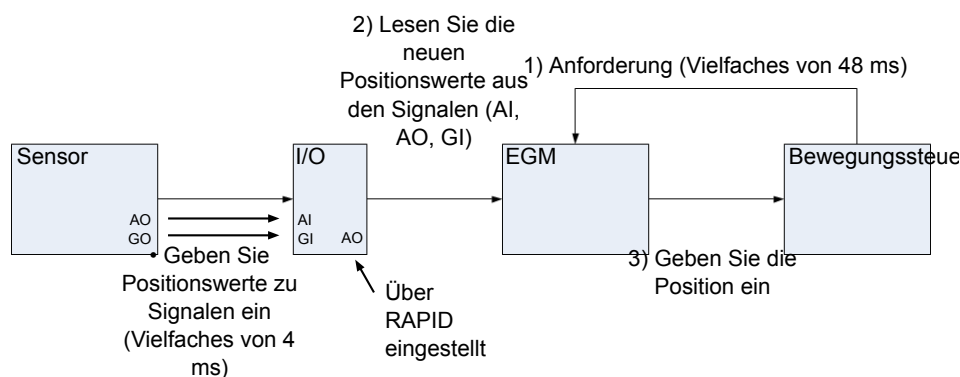
Eingangsdaten für EGM Path Correction

Die Quelle für die Eingangsdaten wird mithilfe der EGM-Einrichtungsinstruktionen ausgewählt. Die drei ersten Instruktionen wählen eine Signalschnittstelle und die letzte Instruktion eine UdpUc-Schnittstelle aus (*User Datagram Protocol Unicast Communication*).

Instruktionen	Beschreibung
EGMSetupAI	Einrichtung analoger Eingangssignale für EGM
EGMSetupAO	Einrichtung analoger Ausgangssignale für EGM
EGMSetupGI	Einrichtung von Gruppeneingangssignalen für EGM
EGMSetupUC	Einrichten des UdpUc-Protokolls für EGM

Eingangsdaten für EGM enthalten vor allem Positionsdaten.

Nachfolgend wird der Datenfluss für die Signalschnittstelle dargestellt:



xx1400002016

- 1 Die Bewegungssteuerung ruft EGM auf.
- 2 Die Messdaten (Y- und Z-Werte) werden aus den Signalen gelesen oder vom Sensor in Vielfachen von etwa 48 abgerufen.
- 3 EGM berechnet die Positionskorrektur und schreibt diese in die Bewegungssteuerung. Wenn das UdpUc-Protokoll verwendet wird, wird Feedback an den Sensor gesendet.

2 Verwendung von Externally Guided Motion

2.4 Ausgangsdaten

2.4 Ausgangsdaten

Beschreibung

Ausgangsdaten sind nur für die UdpUc-Schnittstelle verfügbar.

Instruktionen zur Implementierung des UdpUc-Protokolls für ein externes Gerät finden Sie unter [Das EGM-Sensorprotokoll auf Seite 33](#). Dort finden Sie außerdem eine Beschreibung der Ausgangsdaten.

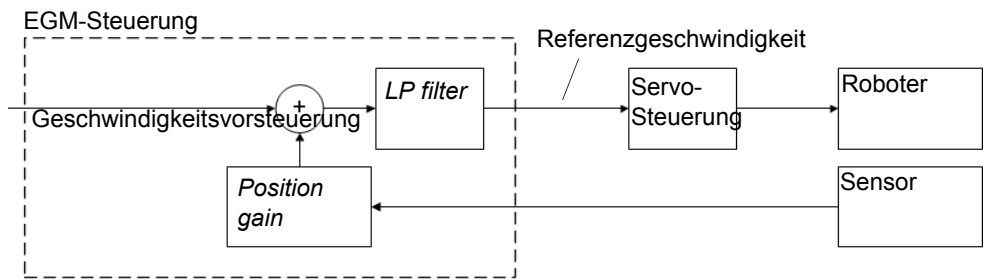
2.5 Konfiguration

Konfiguration für den EGM-Positionsdatenstrom

Die Standardkonfiguration, die in RobotWarevordefiniert ist, funktioniert für jedes Positions-Streaming. Sie funktioniert auch mit der gleichen Konfiguration wie für die EGM-Positionsdatensteuerung.

Konfiguration für EGM Position Guidance

Das EGM-Verhalten kann mithilfe der Systemparameter vom Typ*External Motion Interface Data* Parametergruppe *Motion* beeinflusst werden. Eine Beschreibung aller verfügbarer EGM-Parameter finden Sie unter [Systemparameter auf Seite 43](#).
Nachfolgend finden Sie eine genauere Beschreibung der beiden Parameter, die die EGM-Steuerungsschleife beeinflussen. Die Abbildung zeigt eine vereinfachte Ansicht des EGM-Steuerungssystems.



xx1400001083

Default proportional Position Gain	Der Parameter <i>Position Gain</i> (Positionsverstärkung) in der Abbildung beeinflusst die Ansprechempfindlichkeit in der Bewegung zur Zielposition, die vom Sensor in Bezug auf die aktuelle Roboterposition vorgegeben wird. Je höher der Wert, desto schneller die Reaktion.
Default Low Pass Filter Bandwidth Time	Der Parameter <i>LP Filter</i> in der Abbildung ist der Standardwert, der verwendet wird, um die Geschwindigkeitsleistung aus EGM zu filtern.

Konfiguration für EGM Path Correction

Die Konfiguration für EGM Path Correction muss *Level* auf *Path* gesetzt haben. Keiner der weiteren Werte wird verwendet.

2.6 Koordinatensysteme

Rahmen für den EGM-Positionsdatenstrom

Positions-Streaming unterstützt keine dynamische Änderung von Werkzeug oder Last. Wenn `EGMStreamStart` zum Starten des Positionsdatenstroms verwendet wird, werden das aktive Werkzeug und die Last an die Steuerung übergeben. Wenn `EGMActJoint` oder `EGMActPose` verwendet wird, werden das aktive Werkzeug und die Last oder das angegebene Werkzeug und/oder die Last an die Steuerung übergeben. Diese Werkzeug- und Lastdaten werden dann von EGM zur Berechnung von Positionen verwendet, bis der Positionsdatenstrom gestoppt wird.

Koordinatensysteme für EGM Position Guidance

EGM kann in zwei verschiedenen Modi ausgeführt werden, Achsenmodus und Positionsmodus. Der folgende Abschnitt gilt nur für den EGM-Positionsmodus.

Für den Achsenmodus werden keine Referenz-Koordinatensysteme benötigt, da beide Sensorwerte und die Positionswerte Achsenwinkel sind, die relativ zur Kalibrierposition jeder Achse in Grad angegeben werden. Für den Positionsmodus werden jedoch Referenz-Koordinatensysteme benötigt. Die Messungen des Sensors und die Richtungen für die Positionsänderung können nur relativ zu Referenz-Koordinatensystemen angegeben werden.

Die RAPID-Instruktion `EGMActPose` definiert alle Koordinatensysteme, die in EGM verfügbar sind:

Koordinatensystem	Beschreibung
Werkzeug	Die Werkzeugdaten, die für den EGM-Prozess verwendet werden, werden mit dem optionalen <code>\Tool</code> -Argument definiert.
Werkobjekt	Die Werkobjektdaten, die für den EGM-Prozess verwendet werden, werden mit dem optionalen <code>\Wobj</code> -Argument definiert.
Korrektur	Das Koordinatensystem, das verwendet wird, um die endgültige Bewegungsrichtung anzugeben, wird vom obligatorischen <code>CorrFrame</code> -Argument definiert.
Sensor	Das Koordinatensystem, das verwendet wird, um die Sensordaten auszuwerten, wird vom obligatorischen <code>SensorFrame</code> -Argument definiert.

Werkzeuge und Werkobjekte

Das Werkzeug und das Werkobjekt können nur in zwei Kombinationen definiert werden:

- 1 Wenn das Werkzeug am Roboter angeschlossen wird, muss das Werkobjekt befestigt werden.
- 2 Wenn das Werkzeug befestigt wird, muss das Werkobjekt am Roboter angeschlossen werden.

Fortsetzung auf nächster Seite



Hinweis

Es ist nicht möglich, ein Werkobjekt oder Werkzeug zu verwenden, das an einer anderen mechanischen Einheit als dem EGM-Roboter angeschlossen ist.

Vordefinierte Koordinatensystemtypen

Für die Koordinatensysteme `CorrFrame` und `SensorFrame` ist es ebenfalls notwendig zu wissen, worauf sie sich beziehen. Diese Informationen werden mithilfe des vordefinierten Koordinatensystemtyps im Datentyp `egmframetype` angegeben:

Wert	Beschreibung
<code>EGM_FRAME_BASE</code>	Das Koordinatensystem wird relativ zum Basis-Koordinatensystem (Positionsmodus) definiert.
<code>EGM_FRAME_TOOL</code>	Das Koordinatensystem wird relativ zu <code>tool0</code> (Positionsmodus) definiert.
<code>EGM_FRAME_WOBJ</code>	Das Koordinatensystem wird relativ zum verwendeten Werkobjekt (Positionsmodus) definiert.
<code>EGM_FRAME_WORLD</code>	Das Koordinatensystem wird relativ zum Weltkoordinatensystem (Positionsmodus) definiert.
<code>EGM_FRAME_JOINT</code>	Die Werte sind Achsenwerte (Achsenmodus).

Koordinatensysteme für EGM Path Correction

EGM Path Correction kann nur im Pose-Modus ausgeführt werden.

Die RAPID-Instruktion `EGMActMove` definiert den einzigen Rahmen, der für die EGM Path Correction benötigt wird. Werkzeug- und Arbeitsobjekt sind in `EGMMoveL` oder `EGMMoveC` angegeben.

Werkzeuge und Werkobjekte

Das Werkzeug muss mit dem Roboter verbunden sein, und das Werkobjekt kann durch eine andere mechanische Einheit fixiert oder bewegt werden.

Diese Seite wurde absichtlich leer gelassen

3 Das EGM-Sensorprotokoll

3.1 Überblick

3.1.1 EGM Sensorprotokollübersicht

Kommunikations- und Transportprotokoll

Das EGM-Sensorprotokoll wurde für eine schnelle Kommunikation zwischen einer Robotersteuerung und einem Kommunikationsendpunkt mit minimalen Mehraufwand entwickelt.

Der Kommunikationsendpunkt ist normalerweise ein Sensor, daher wird von nun an *Sensor* anstelle des Kommunikationsendpunkts verwendet. Manchmal ist der Sensor mit einem PC verbunden, und der PC überträgt die Sensordaten an den Roboter. Der Zweck des Sensorprotokolls ist es, Sensordaten häufig zwischen der Robotersteuerung und den Sensoren zu kommunizieren. EGM sensor protocol verwendet Google Protocol Buffers für die Codierung und UDP als Transportprotokoll. Google Protocol Buffers wurde aufgrund seiner Geschwindigkeit und Sprachneutralität ausgewählt. UDP wurde als Transportprotokoll ausgewählt, da die gesendeten Daten *Echtzeit*-Daten sind, die mit hoher Frequenz gesendet werden. Wenn Pakete verloren gehen, ist es nicht hilfreich die Daten erneut zu senden.

Datenstrukturen und Systemparameter

Die EGM sensor protocol-Datenstrukturen werden von der EGM proto-Datei definiert. Sensorname, IP-Adresse und Portnummer von Sensoren werden in den Systemparametern konfiguriert. Es können maximal acht Sensoren konfiguriert werden.

Nachrichten und Warteschlangenbehandlung

Der Sensor verhält sich wie ein Server und er muss eine erste Meldung von der Robotersteuerung empfangen, bevor er etwas an den Roboter senden kann. Meldungen können nach dieser ersten Meldung unabhängig voneinander in beide Richtungen gesendet werden. Anwendungen, die das Protokoll verwenden, können die Verwendung beschränken. Das Protokoll selbst verfügt jedoch nicht über eine integrierte Synchronisierung von Anforderungsantworten oder über eine Überwachung von verlorenen Meldungen. Es gibt keine besonderen Meldungen zur Verbindung oder zur Trennung, sondern nur Daten, die unabhängig voneinander in beide Richtungen fließen können. Die erste Meldung vom Roboter ist eine Datenmeldung. Man muss auch bedenken, dass ein Sender einer UDP-Meldung weiterhin Meldungen sendet, auch wenn die Warteschlange des Empfängers möglicherweise bereits voll ist. Der Empfänger muss sicherstellen, dass er seine Warteschlange leert.

Standardmäßig sendet und liest der Roboter die Daten vom Sensor alle 4 Millisekunden, unabhängig davon, wann die Daten vom Sensor gesendet werden. Diese Zykluszeit kann auf ein Vielfaches von 4 ms geändert werden, indem das

Fortsetzung auf nächster Seite

3 Das EGM-Sensorprotokoll

3.1.1 EGM Sensorprotokollübersicht

Fortsetzung

optionale Argument `\SampleRate` der RAPID Instruktionen `EGMStreamStart`, `EGMActJoint` oder `EGMActPose` verwendet wird.

Für jeden Bewegungstask ist ein separater UDP-Kanal erforderlich.

3.1.2 Google Protocol Buffers

Überblick über Protobuf

Google Protocol Buffers oder Protobuf ist eine Möglichkeit, Daten auf sehr effiziente Weise zu serialisieren/deserialisieren. Protobuf ist im Allgemeinen 10–100-mal schneller als XML. Im Internet gibt es viele Informationen über Protobuf und *Google overview* ist ein guter Anfang.

Kurz gesagt, Meldungsstrukturen werden in einer *.proto*-Datei beschrieben. Die *.proto*-Datei wird anschließend kompiliert. Der Compiler erzeugt einen serialisierten/deserialisierten Code, der anschließend von der Anwendung verwendet wird. Die Anwendung liest eine Meldung aus dem Netzwerk, führt die Deserialisierung aus, erzeugt eine Meldung, ruft die Serialisierungsmethode auf und sendet anschließend die Meldung.

Es ist möglich, Protobuf in den meisten Programmiersprachen zu verwenden, da Protobuf eine neutrale Sprache ist. Je nach Sprache gibt es viele verschiedene Implementierungen.

Der Hauptvorteil von Protobuf ist, dass Protobuf-Meldungen in ein Binärformat serialisiert werden, wodurch das Debuggen von Paketen mithilfe einer Netzwerkanalyse erheblich erschwert wird.

Tools von Drittanbietern

Bis auf das *Google C++*-Werkzeug haben wir außerdem die folgenden Werkzeuge und Codes von Drittanbietern überprüft:

- *Nanopb* erzeugt einen C-Code. Eine dynamische Speicherbelegung wird nicht benötigt.
- *Protobuf-net*, eine Google Protobuf .NET-Bibliothek.
- *Protobuf-csharp* eine Google Protobuf .NET-Bibliothek, C# API ist vergleichbar mit Google C++ API.



Hinweis

Beachten Sie, dass der oben genannte Code Open Source ist, was bedeutet, dass Sie in der Lizenz prüfen müssen, ob der Code mit dem Produkt verwendet werden darf.

3 Das EGM-Sensorprotokoll

3.1.3 EGM Beschreibung des Sensorprotokolls

3.1.3 EGM Beschreibung des Sensorprotokolls

Datenstruktur

Das EGM-Sensorprotokoll ist kein Anforderungs- bzw. Antwortprotokoll. Der Sensor kann Daten auf jeder Frequenz senden, nachdem der Sensor die erste Meldung vom Roboter empfängt.

Das EGM-Sensorprotokoll verfügt über zwei Hauptdatenstrukturen, *EgmRobot* und *EgmSensor*. *EgmRobot* wird vom Roboter gesendet und *EgmSensor* wird vom Sensor gesendet. Alle Meldungsfelder in beiden Datenstrukturen sind als optional definiert, was bedeutet, dass ein Feld in einer Meldung vorhanden sein kann oder nicht. Anwendungen, die *Google Protocol Buffers* verwenden, müssen überprüfen, ob optionale Felder vorhanden sind.

Zur Google protobuf-Datenstruktur kann das *wiederholte* Element gehören, also eine Liste von Elementen des gleichen Typs. Die Anzahl von *wiederholten* Elementen beträgt maximal sechs Elemente in dem EGM sensor protocol, mit der Ausnahme von *EGMRAPIDdata*, das einen Höchstwert von 40 hat.

Sensorprotokollmeldungen

Dieser Abschnitt beschreibt einige der Sensorprotokollmeldungen. Beachten Sie, dass neue optionale Felder vorhanden sein können, auf die dieses Handbuch nicht eingeht.

EgmHeader

EgmHeader ist *EgmRobot* und *EgmSensor* gemeinsam.

```
message EgmHeader
{
    optional uint32 seqno = 1; // sequence number (to be able to find
        lost messages)
    optional uint32 tm = 2; // time stamp in milliseconds

    enum MessageType {
        MSGTYPE_UNDEFINED = 0;
        MSGTYPE_COMMAND = 1; // for future use
        MSGTYPE_DATA = 2; // sent by robot controller
        MSGTYPE_CORRECTION = 3; // sent by sensor for position guidance
        MSGTYPE_PATH_CORRECTION = 4; // sent by sensor for path
            correction
    }

    optional MessageType mtype = 3 [default = MSGTYPE_UNDEFINED];
}
```

Variable	Beschreibung
seqno	Sequenznummer. Die Anwendungen müssen die Sequenznummer für jede Meldung, die sie senden, um eins erhöhen. Dadurch kann nach verlorenen Meldungen in einer Reihe von Meldungen gesucht werden.

Fortsetzung auf nächster Seite

Variable	Beschreibung
tm	Zeitstempel in Millisekunden. (Kann zur Überwachung von Verzögerungen verwendet werden).
mtype	Meldungstyp. Muss vom Sensor auf MSGTYPE_CORRECTION eingestellt werden und wird von der Robotersteuerung auf MSGTYPE_DATA eingestellt.

EgmRobot

```

message EgmRobot
{
    optional EgmHeader          header = 1;
    optional EgmFeedBack        feedBack = 2;
    optional EgmPlanned          planned = 3;

    optional EgmMotorState       motorState = 4;
    optional EgmMCISState        mciState = 5;
    optional bool                mciConvergenceMet = 6;
    optional EgmTestSignals      testSignals = 7;
    optional EgmRapidCtrlExecState rapidExecState = 8;
    optional EgmMeasuredForce    measuredForce = 9;
    optional double              utilizationRate=10;
    optional uint32              moveIndex=11;
    optional EgmCollisionInfo     CollisionInfo = 12;
    optional EgmRAPIDdata         RAPIDfromRobot = 13;
}

```

Variable	Beschreibung
header	Referenz zu EgmHeader.
feedback	Mögliche Rückmeldung, also Messposition für einen Roboter und zusätzliche Achsen. Weitere Informationen finden Sie in der Beschreibung von EgmFeedBack.
planned	Referenzposition für den Roboter und zusätzliche Achsen. Weitere Informationen finden Sie in der Beschreibung von EgmPlanned.
motorState	Motoren-ein-Zustand (Ein, Aus).
mciState ⁱ	Interner EGM-Zustand (Läuft, gestoppt, Fehler).
mciConvergenceMet	Boolesch zeigt, ob das in der EGM-Instruktion definierte Konvergenzkriterium erfüllt ist
testSignals	Nur für internen Gebrauch.
rapidExecState	RAPID-Programmabarbeitungsstatus (Läuft, gestoppt).
measuredForce	Gemessene Kontaktkraft. Zeigt die rohe Sensorkraft an, wenn die Kraftkontrolle nicht aktiv ist.

3 Das EGM-Sensorprotokoll

3.1.3 EGM Beschreibung des Sensorprotokolls

Fortsetzung

Variable	Beschreibung
utilizationRate	Dieser Wert ist ein Prozentwert, der anzeigt, wie viel der verfügbaren Roboterleistung erforderlich ist, um eine Bewegung gemäß dem aktuell gesendeten <code>EgmSensor</code> -Paket auszuführen. Wenn dieser Wert 100 % überschreitet, kann die Steuerung die volle Bewegung ggf. nicht vollständig ausführen. Dann ändert EGM die Referenz so, dass der Roboter nur eine Teilbewegung in die gewünschte Richtung ausführt. Wenn <code>EgmSensor</code> -Pakete fehlen oder verzögert sind, wird das zuletzt gesendete <code>EgmPackage</code> verwendet. Das kann auch dazu führen, dass die <code>utilizationRate</code> temporär sehr hoch ist.
moveIndex	Einfacher Indexzähler, der den Beginn der Bewegung und die Änderung des Ziels anzeigt.
CollisionInfo	Zeigt einen Kollisionshinweis für eine Achse an.
RAPIDfromRobot	E/A-Signale und Array von <code>dnum</code> , die von dem Roboter an den Sensor gesendet werden. Diese Daten können auf verschiedene Arten verwendet werden und erhöht die Flexibilität bei der Nutzung von EGM.

ⁱ Das Akronym „MCI“ bezieht sich auf die „Motion Correction Interface“ (Bewegungskorrekturschnittstelle), die interne Steuerungsschnittstelle, die EGM verwendet.

EgmSensor

```
message EgmSensor
{
    optional EgmHeader          header = 1;
    optional EgmPlanned         planned = 2;
    optional EgmSpeedRef        speedRef = 3;
    optional EgmRAPIDdata       RAPIDtoRobot = 4;
}
```

Variable	Beschreibung
header	Referenz zu <code>EgmHeader</code> .
planned	Referenzposition für Roboter und zusätzliche Achsen.
speedRef	Referenzgeschwindigkeit für Roboter und zusätzliche Achsen.
RAPIDfromSensor	E/A-Signale und Array von <code>dnum</code> , die von dem Sensor an den Roboter gesendet werden. Diese Daten können auf verschiedene Arten verwendet werden und erhöht die Flexibilität bei der Nutzung von EGM.

EgmPlanned

```
message EgmPlanned
{
    optional EgmJoints          joints = 1;
    optional EgmPose            cartesian = 2;
    optional EgmJoints external Joints = 3;
    optional EgmClock           time = 4;
}
```

Variable	Beschreibung
joints	Geplante Achsenposition für Roboter.

Fortsetzung auf nächster Seite

Variable	Beschreibung
cartesian	Geplante kartesische Position für Roboter.
externalJoints	Geplante Position für externe Achsen (6 Werte).
time	Zeitstempel, wann der Roboter und die externen Achsen die geplante Position erreichen.

EgmFeedBack

```
message EgmFeedBack
{
    optional EgmJoints          joints = 1;
    optional EgmPose            cartesian = 2;
    optional EgmJoints external Joints = 3;
    optional EgmClock           time = 4;
}
```


Variable	Beschreibung
joints	Gemessene Achsenposition für Roboter.
cartesian	Gemessene kartesische Position für Roboter.
externalJoints	Gemessene Position für externe Achsen (6 Werte).
time	Zeitstempel, als der Roboter und die externen Achsen in der Messposition waren.

3.2 Aufbau eines EGM-Sensorkommunikationsendpunktes

Erstellen eines EGM-Sensorkommunikationsendpunkts unter Verwendung von .Net

In diesem Handbuch wird vorausgesetzt, dass Sie mithilfe von Visual Studio erstellen und kompilieren und dass Sie mit diesem Arbeitsgang vertraut sind.

Nachfolgend finden Sie eine kurze Beschreibung zur Installation und Erstellung einer einfachen Testanwendung unter Verwendung von *protobuf-csharp-port*.

	Aktion
1	In Visual Studio, erzeugen Sie eine C#-Anwendung.
2	Wählen Sie Werkzeuge und dann NuGet Package Manager , und installieren Sie <i>Google.Protobuf</i> .
3	Im NuGet Package Manager installieren Sie auch <i>Google.Protobuf.Tools</i> .
4	Navigieren Sie zu <i>Solution package\packages\Google.Protobuf.Tools.3.xx.x\tools\windows_x64</i> .
5	Kopieren Sie den EGM-Ordner aus <i>C:\ProgramData\ABB\DistributionPackages\ABB.RobotWare-7.yy\RobotPackages\RobotControl_7.zz\utility</i> in <i>\packages\Google.Protobuf.Tools.3.xx.x\tools\windows_x64</i> .
6	Öffnen Sie eine cmd-Zeile und führen Sie „ <code>.\protoc .\egm\egm.proto --csharp_out=. \egm</code> “ aus. => die Egm.cs-Datei wird aufgebaut.  Hinweis Die <i>egm.proto</i> Syntax lautet <code>proto2</code> .
7	Fügen Sie die generierte Datei <i>egm.cs</i> zum Visual Studio-Projekt hinzu (vorhandenes Element hinzufügen).
8	Kopieren Sie den Beispielcode in die Visual Studio Windows Console-Anwendungsdatei (<i>egm-sensor.cs</i>) und kompilieren, verknüpfen und führen Sie sie aus.

3.3 Konfigurieren von UdpUc-Geräten

Konfigurieren von UdpUc-Geräten

UdpUc kommuniziert mit maximal acht Geräten über Udp. Die Geräte verhalten sich wie Server und die Robotersteuerung wie ein Client. Die Robotersteuerung initiiert die Verbindung mit dem Sensor.

Jeder UDP-Kanal ist als Gerät definiert, d. h. Sie müssen für jeden Bewegungstask, bei der Sie EGM verwenden möchten, ein Gerät einrichten.

Systemparameter

Dies ist eine kurze Beschreibung der Parameter, die beim Konfigurieren eines Geräts verwendet werden. Weitere Informationen über die Parameter siehe *Technisches Referenzhandbuch - Systemparameter*.

Diese Parameter gehören zum Typ *UDP Unicast Device* in der Parametergruppe *Communication*.

Parameter	Beschreibung
<i>Name</i>	Der Name der UDP Unicast Device-Instanz. Zum Beispiel <i>EGMsensor</i> .
<i>Type</i>	Der Typ des zu verwendenden UDP Unicast Device-Protokolls. Der einzige verfügbare UDP Unicast Device-Typ ist <i>UDPUC</i> .
<i>Remote Address</i>	Die IP-Adresse des externen Gerätes, z. B. ein Sensor.
<i>Remote Port Number</i>	Die Portnummer des Netzwerkknotens an, die identifiziert wird von <i>Remote Address</i> .
<i>Local Port Number</i>	Die Portnummer, auf der die Steuerung auf Broadcast-Nachrichten wartet.

Konfigurationsbeispiel

Das Gerät, das die Eingangsdaten für EGM enthält, muss folgendermaßen als UdpUc-Gerät konfiguriert werden:

Name	Type	Remote Address	Remote Port Number
UCdevice	UDPUC	192.168.10.20	6510

Diese Seite wurde absichtlich leer gelassen

4 Systemparameter

4.1 Typ *External Motion Interface Data*

4.1.1 Der Typ External Motion Interface Data

Überblick

In diesem Abschnitt wird der Typ *External Motion Interface Data* des Themas *Motion* beschrieben. Beschreibungen der einzelnen Parameter des Typs finden Sie in einem eigenen Informationsthema dieses Abschnitts.

Typenbeschreibung

Der Typ *External Motion Interface Data* enthält eine Reihe von Parametern, die die Merkmale für eine *External Motion Interface Data* bestimmen.

4 Systemparameter

4.1.2 Name

4.1.2 Name

Einordnung

Name gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

Der Name des *External Motion Interface Data*.

Verwendung

Dies ist die öffentliche Identität von *External Motion Interface Data*.

Der Parameter bedarf keinen Neustart der Steuerung, wenn dieser geändert wird.

Zulässige Werte

Eine Zeichenfolge mit maximal 32 Zeichen.

4.1.3 Level

Einordnung

Level gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

External Motion Interface Level bestimmt die Systemebene, auf der Korrekturen durchgeführt werden.

Verwendung

Für "Level" sind folgende Werte möglich:

Wert:	Name	Beschreibung:
0	Raw	Entspricht Rohkorrekturen, die kurz vor der Servo-Steuerung hinzugefügt wurden
1	Filtern	Bezieht sich auf zusätzliches Filtern der Korrekturen, leitet jedoch auch einige zusätzliche Verzögerungen und Wartezeiten ein
2	Bahn	Gilt für Bahnkorrekturen.

Der Parameter bedarf keinen Neustart der Steuerung, wenn dieser geändert wird.

Einschränkung

Wenn *Level 0* verwendet wird, dann ist ein Tiefpassfiltern nötig, um Vibrationen des Roboters zu vermeiden.

Zulässige Werte

Zulässige Werte sind Level 0, 1 oder 2

Der Standardwert ist 1.

4 Systemparameter

4.1.4 Do Not Restart after Motors Off

4.1.4 Do Not Restart after Motors Off

Einordnung

Do Not Restart after Motors Off gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

Do Not Restart after Motors Off gibt an, ob die *External Motion Interface*-Ausführung automatisch neu gestartet werden soll, nachdem sich die Steuerung im Status „Motoren aus“ befunden hat, beispielsweise nach einem Not-Halt.

Verwendung

Bei *False* (Standard) werden die Korrekturen in demselben Status wie beim Eintritt in den Status „Motoren aus“ weitergeführt.

Bei *True* wird die Ausführung mit allen Korrekturen im Standby-Modus weitergeführt.

Zulässige Werte

True oder False.

4.1.5 Return to Program Position when Stopped

Einordnung

Return to Program Position when Stopped gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

Return to Program Position when Stopped bestimmt, ob die Achsen, die zurzeit *External Motion Interface* ausführen, auf die programmierte Position zurückkehren müssen, sobald die Programmausführung angehalten wird.

Verwendung

Bei *False*, stoppen die Achsen an der aktuellen Position.

Bei *True*, bewegen sich die Achsen in Richtung der programmierten Startposition.

Einschränkung

Die Bewegung, die die Achsen zurück auf die programmierte Position bringt, wird im Achsenraum definiert. Sind die Achsen weit von der programmierten Position entfernt, wenn *Return to Programmed Position when Stopped* als *False* definiert ist, können unerwartete Verfahrbewegungen die Folge sein. Deswegen wird empfohlen, diesen Wert nur auf *False* einzustellen, wenn sicher ist, dass die Distanz von der programmierten Position zur Korrekturposition gering ist.

Zulässige Werte

True oder False.

4 Systemparameter

4.1.6 Default Ramp Time

4.1.6 Default Ramp Time

Einordnung

Default Ramp Time gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

Default Ramp Time definiert die standardmäßige Gesamtzeit zum Stoppen der *External Motion Interface*-Bewegungen des *External Motion Interface*-Ausführung gestoppt wird.

Verwendung

Dieser Wert legt fest, wie schnell die Geschwindigkeit vom *External Motion Interface* auf Null reduziert wird, wenn die Programmabarbeitung gestoppt wird, und wie schnell die Achsen zu der programmierten Position zurückkehren, wenn *Return to Programmed Position when Stopped True* ist.

Normalerweise kann dieser Wert kleiner als 1 sein. Während der Anwendung ist dieser Wert abzustimmen und zu prüfen. Große Roboter mit hoher Nutzlast bei hohen Geschwindigkeiten erfordern einen höheren Wert. Andererseits kann ein langsamer laufender kleiner Roboter mit geringer Nutzlast mit einem niedrigen Wert arbeiten, um schnell anzuhalten.



Hinweis

Da die Bewegung beim Hochfahren eine Achsenbewegung ist, weicht der Roboter beim Stopp von der gegenwärtigen Position und Guidance-Position ab.

Einschränkung

Der Wert beeinflusst nur den Teil der Bewegung, der von der *External Motion Interface*-Ausführung generiert wurde. Es beeinflusst keine gleichzeitigen Bewegungen die z.B. in RAPID in einer zweiten Bewegungstask programmiert wurden.

Zulässige Werte

Ein Wert von 0.005 bis 10.0 Sekunden.

Der Standardwert ist 0.5 Sekunden.

4.1.7 Default Proportional Position Gain

Einordnung

Default Proportional Position Gain gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

Default Proportional Position Gain definiert die standardmäßige Proportionalverstärkung für die Positionsregelung des "*External Motion Interface*".

Zulässige Werte

Ein Wert zwischen 0.0 und 20.0.
Der Standardwert ist 5.0.

4 Systemparameter

4.1.8 Default Low Pass Filter Bandwidth

4.1.8 Default Low Pass Filter Bandwidth

Einordnung

Default Low Pass Filter Bandwidth gehört zum Typ *External Motion Interface Data* unter dem Thema *Motion*.

Beschreibung

Default Low Pass Filter Bandwidth Time definiert die Standard-Bandbreite des Tiefpassfilters, der verwendet wird, um die Geschwindigkeitsleistung der *External Motion Interface*-Ausführung zu filtern.

Zulässige Werte

Ein Wert zwischen 0.0 und 100.0 Hz.

Der Standardwert ist 20.0 Hz.



Hinweis

Zu niedrige oder zu hohe Werte können zu unerwarteten Bewegungen führen.

5 RAPID Referenzinformation

5.1 Instruktionen

5.1.1 EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition

Verwendung

EGMActJoint aktiviert einen spezifischen EGM Prozess und definiert statische Daten für die sensorgeführte Bewegung zu einer Achswinkelposition, d. h., Daten, die selten zwischen den verschiedenen EGM Bewegungen verändert werden.

Grundlegende Beispiele

```
VAR egmident egmID1;
PERS pose pose1:=[[0,0,0], [1,0,0,0]];
CONST egm_minmax egm_minmax1:=[-1,1];
VAR signaldo RobotDO 1;
PERS dnum RobotOut{10}:= [3.0, 6.0, 9.0, 12.0, 15.0, 18.0, 21.0,
    24.0, 27.0, 30];
VAR signaldi RobotDI 0;
PERS dnum RobotIn{10}:= [33.0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_01
    \aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
    \aiR6rz:=ai_06;
EGMActJoint egmID1 \DOtoSensor:=RobotDO \DataToSensor:=RobotOut
    \DIfromSensor:=RobotDI \DataFromSensor:=RobotIn
    \J1:=egm_minmax1 \J3:=egm_minmax1 \J4:=egm_minmax1;
```

Argumente

```
EGMActJoint EGMid [\StreamStart] [\Tool] [\WObj] [\TLoad]
    [\DOtoSensor] [\DataToSensor] [\DIfromSensor]
    [\DataFromSensor] [\J1] [\J2] [\J3] [\J4] [\J5] [\J6] [\J7]
    [\LpFilter] [\SampleRate] [\MaxPosDeviation]
    [\MaxSpeedDeviation]
```

EGMid

Datentyp: egmident

EGM Identität.

[\StreamStart]

Datentyp: switch

StreamStart startet das Streaming von Positionsdaten an externe Geräte. Die Daten werden mit der mit \SampleRate definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei *egm.proto*.

StreamStart ist nur verfügbar, wenn EGM unter Verwendung von EGMSetupUC eingerichtet wird, d. h. das Protokoll UdpUc wird für die Kommunikation mit dem externen Gerät verwendet.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.1 EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition

Externally Guided Motion

Fortsetzung

`[\Tool]`

Datentyp: `tooldata`

Das Werkzeug, das für Bewegungen verwendet wird, die mit der Instruktion `EGMRunJoint` durchgeführt werden.

Das Argument `[\Tool]` ist optional. Der Standardwert, wenn das Argument ausgelassen wird, ist `tool0`.

`[\Wobj]`

Das Werkobjekt, das für Bewegungen verwendet wird, die mit der Instruktion `EGMRunJoint` durchgeführt werden.

Work Object

Datentyp: `wobjdata`

Das Werkobjekt (Objekt-Koordinatensystem), auf das sich die Roboterposition in der Instruktion bezieht.

Dieses Argument kann ausgelassen werden; in diesem Fall bezieht sich die Position auf das Welt-Koordinatensystem. Wenn allerdings ein stationärer TCP oder koordinierte externe Achsen verwendet werden, muss dieses Argument angegeben werden, damit eine kreisförmige Bahn in Relation zum Werkobjekt möglich ist.

`[\TLoad]`

Die Last, die für Bewegungen verwendet wird, die mit der Instruktion `EGMRunJoint` durchgeführt werden.

Total load

Datentyp: `loaddata`

Das Argument `\TLoad` beschreibt die in der Bewegung verwendete Gesamtlast. Die Gesamtlast ist die Summe der Werkzeuglast und der Nutzlast des Werkzeugs. Bei Verwendung des Arguments `\TLoad` werden die `loaddata` in den aktuellen `tooldata` nicht berücksichtigt.

Wenn das `\TLoad`-Argument auf `load0` gesetzt ist, wird das `\TLoad`-Argument nicht berücksichtigt, und stattdessen werden die `loaddata` in den aktuellen `tooldata` verwendet.

Damit das `\TLoad`-Argument verwendet werden kann, muss der Wert des Systemparameters `ModalPayloadMode` unbedingt auf 0 gesetzt werden. Wird `ModalPayloadMode` auf 0 eingestellt, kann die `GripLoad`-Instruktion nicht mehr genutzt werden.

Die Gesamtlast kann mit der Serviceroutine `LoadIdentify` ermittelt werden. Wurde der Systemparameter `ModalPayloadMode` auf 0 gesetzt, kann der Bediener beim Ausführen der Serviceroutine die `loaddata` vom Werkzeug in eine bereits vorhandene oder eine neue persistente `loaddata`-Variable kopieren.

Es ist möglich, über ein digitales Eingangssignal, das mit dem Systemeingang `SimMode` (Simulationsmodus) verbunden ist, einen Testlauf des Programms ohne Nutzlast durchzuführen. Wenn das digitale Eingangssignal auf 1 gesetzt wird,

Fortsetzung auf nächster Seite

5.1.1 EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition

Externally Guided Motion
Fortsetzung

werden die `loaddata` im optionalen Argument `\TLoad` nicht berücksichtigt und stattdessen die `loaddata` in den aktuellen `tooldata` verwendet.



Hinweis

Die Standardfunktion für den Umgang mit Nutzlasten ist die Instruktion `GripLoad`. Deshalb beträgt der Standardwert für den Systemparameter `ModalPayloadMode` 1.

`[\DOToSensor]`

Datentyp: `signaldo`

Ein E/A-Signal, das von der Robotersteuerung an eine externe Schnittstelle, z. B. einen Sensor, gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

`[\DataToSensor]`

Datentyp: `dnum`

Ein Array mit bis zu 40 `dnum`-Werten, das von der Robotersteuerung an eine externe Schnittstelle, z. B. einen Sensor, gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

`[\DIFromSensor]`

Datentyp: `signaldi`

Ein E/A-Signal, das von einer externen Schnittstelle, z. B. ein Sensor, an die Robotersteuerung gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

`[\DataFromSensor]`

Datentyp: `dnum`

Ein Array mit bis zu 40 `dnum`-Werten, das von einer externen Schnittstelle, z. B. ein Sensor, an die Robotersteuerung gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.1 EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition

Externally Guided Motion

Fortsetzung

`[\J1] [\J2] [\J3] [\J4] [\J5] [\J6] [\J7]`

Datentyp: `egm_minmax`

Die Konvergenzkriterien für Achse 1 bis 6 in Grad für 6-achsige Roboter und Achse 1 bis 7 in Grad für 7-achsige Roboter. Der Standardwert ist $\pm 0,5$ Grad.

Die Daten der Konvergenzkriterien können verwendet werden, um zu entscheiden, ob der Roboter die angeforderten Verbindungspositionen erreicht hat. Wenn die Differenz zwischen der angeforderten Verbindungsposition und der tatsächlichen Verbindungsposition im Bereich von `egm_minmax.min` und `egm_minmax.max` liegt, wird davon ausgegangen, dass das Gelenk seine angeforderte Position erreicht hat. Wenn für eine Verbindung, die in `EGMRunJoint` ausgewählt wurde, keine Konvergenzkriterien angegeben sind, wird der Standardwert verwendet.

Sobald alle Verbindungen, die in `EGMRunJoint` angegeben wurden, ihre angeforderten Positionen erreicht haben, hat der Roboter selbst seine angeforderte Position erreicht, und die RAPID-Abarbeitung fährt mit der nächsten RAPID-Instruktion fort.

`[\LpFilter]`

Datentyp: `num`

Low Pass Filter Bandwidth, in Hertz (Hz), zum Filtern von Sensorrauschen verwendet.

`[\SampleRate]`

Datentyp: `num`

Eingangsdaten Abtastlesegeschwindigkeit als Vielfaches von 4 Millisekunden. Gültige Werte sind 4, 8, 12, 16 usw.

Der Standardwert ist 4 Millisekunden.

`[\MaxPosDeviation]`

Datentyp: `num`

Maximale Achsabweichung von der einprogrammierten Position in Grad, d. h. von dem Feinpunkt, an dem die EGM-Bewegung begonnen hat. Es wird für alle Verbindungen der gleiche Wert verwendet.

Der Standardwert ist 1000 Grad.

`[\MaxSpeedDeviation]`

Datentyp: `num`

Maximal zulässige Achsgeschwindigkeitsveränderung in Grad/Sekunde. Alle Achsen werden in demselben Verhältnis reduziert, wenn eine Achse eingeschränkt ist.

Der Standardwert ist 1,0 Grad/Sekunde.

Einschränkungen

- Wenn `EGMActJoint` mehrmals mit derselben `EGMId` ausgeführt wird, werden die aktuellsten Aktivierungsdaten für `EGMRunJoint`-Instruktionen verwendet, die nachfolgen, bis eine neue `EGMActJoint` durchgeführt wird.
- `EGMActJoint` kann nur in RAPID-Bewegungstasks verwendet werden.

Fortsetzung auf nächster Seite

5.1.1 EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition

Externally Guided Motion
Fortsetzung

Syntax

```
EGMActJoint
[EGMid ':'<variable (VAR) of egmident>]
['\Tool ':'<persistent (PERS) of tooldata>]
['\Wobj ':'<persistent (PERS) of wobjdata>]
['\TLoad ':'<persistent (PERS) of loaddata>]
['\DToSensor ':'<variable> (VAR) of signaldo]
['\DataToSensor ':'<persistent> (PERS) of dnum]
['\DIFromSensor ':'<variable> (VAR) of signaldi]
['\DataFromSensor ':'<persistent> (PERS) of dnum]
['\J1 ':'<expression (IN) of egm_minmax>]
['\J2 ':'<expression (IN) of egm_minmax>]
['\J3 ':'<expression (IN) of egm_minmax>]
['\J4 ':'<expression (IN) of egm_minmax>]
['\J5 ':'<expression (IN) of egm_minmax>]
['\J6 ':'<expression (IN) of egm_minmax>]
['\J7 ':'<expression (IN) of egm_minmax>]
['\LpFilter ':'<expression (IN) of num>]
['\SampleRate ':'<expression (IN) of num>]
['\MaxPosDeviation ':'<expression (IN) of num>]
['\MaxSpeedDeviation ':'<expression (IN) of num>] ';'
```

Weitere Informationen

Information über	Siehe
Instruktion EGMRunJoint	EGMRunJoint - Ausführen einer EGM Bewegung mit einer Achswinkelposition auf Seite 74
Instruktion EGMStreamStart	EGMStreamStart - Das EGM-Positions-Streaming starten. auf Seite 96
Datentyp egm_minmax	egm_minmax - Konvergenzkriterien für EGM auf Seite 105
MoveJ	<i>Technisches Referenzhandbuch - RAPID Instruktionen, Funktionen und Datentypen</i>

5 RAPID Referenzinformation

5.1.2 EGMActMove - Vorbereiten einer EGM Bewegung mit Bahnkorrektur *Externally Guided Motion*

5.1.2 EGMActMove - Vorbereiten einer EGM Bewegung mit Bahnkorrektur

Verwendung

EGMActMove wird zur Aktivierung eines spezifischen EGM-Prozesses genutzt und definiert statische Daten für die Bahnkorrektur, d. h., Daten, die selten zwischen den verschiedenen EGM Bahnkorrekturbewegungen verändert werden.

Grundlegende Beispiele

Das folgende Beispiel veranschaulicht die Instruktion EGMActMove.

Beispiel 1:

```
VAR egmident EGMidl;  
PERS tooldata tLaser := [TRUE, [[148,50,326],  
    [0.3902618,-0.589657,-0.589656,0.3902630]],  
    [1,[-0.92,0,-0.39], [1,0,0,0], 0,0,0]];  
EGMGetId EGMidl;  
EGMSetupLTAPP ROB_1, EGMidl, "pathCorr", "OptSim", 1\LATR;  
EGMActMove EGMidl, tLaser.tframe\SampleRate:=48;
```

Dieses Programm registriert einen EGM-Prozess und richtet einen Sensor als Datenquelle (Sensor) ein, der das Kommunikationsprotokoll LTAPP verwendet und den Typ *Look Ahead* hat. Der Sensor soll die Achsentypdefinition Nummer 1 für die Verfolgung verwenden. Die Rate, mit der die Steuerung auf das Gerät und das Sensorkoordinatensystem zugreift, wird ebenfalls eingerichtet.

Argumente

```
EGMActMove EGMid, SensorFrame [\SampleRate]
```

EGMid

Datentyp: egmident
EGM Identität.

SensorFrame

Datentyp: pose
Sensorkoordinatensystem.

[\SampleRate]

Datentyp: num
Eingangsdaten Abtastlesegeschwindigkeit als Vielfaches von 24 Millisekunden.
Gültige Werte sind 24, 48, 72 usw.

Programmabarbeitung

Das Sensorkoordinatensystem und die Sensorabtastrate sind mit einer EGM-Identität verknüpft, bis sie entweder mit EGMReset zurückgesetzt oder von einer anderen EGMActMove-Instruktion verändert werden.

Syntax

```
EGMActMove  
    [EGMid ':='] <variable (VAR) of egmident> ','  
    [SensorFrame ':='] <expression (IN) of pose>
```

Fortsetzung auf nächster Seite

5.1.2 EGMActMove - Vorbereiten einer EGM Bewegung mit Bahnkorrektur

Externally Guided Motion
Fortsetzung

```
[ '\SampleRate ' := ' <expression (IN) of num> ' ; '
```

Weitere Informationen

Information über	Siehe
EGMReset	EGMReset - Zurücksetzen eines EGM Prozesses auf Seite 73

5 RAPID Referenzinformation

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel

Externally Guided Motion

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel

Verwendung

EGMActPose aktiviert einen spezifischen EGM Prozess und definiert statische Daten für die sensorgeführte Bewegung zu einem Positionsziel, d. h., Daten, die selten zwischen den verschiedenen EGM Bewegungen verändert werden.

Grundlegende Beispiele

```
VAR egmident egmID1;
PERS pose pose1:=[[0,0,0], [1,0,0,0]];
CONST egm_minmax egm_minmax_lin:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot:=[-0.1,0.2];
CONST pose posecor:=[[1200,400,900], [0,0,1,0]];
CONST pose posesens:=[[12.3313,-0.108707,416.142],
    [0.903899,-0.00320735,0.427666,0.00765917]];
VAR signaldo RobotDO 1;
PERS dnum RobotOut{10}:= [3.0, 6.0, 9.0, 12.0, 15.0, 18.0, 21.0,
    24.0, 27.0, 30];
VAR signaldi RobotDI 0;
PERS dnum RobotIn{10}:= [33.0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_01
    \aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
    \aiR6rz:=ai_06;
EGMActPose egmID1 \Tool:=tool0 \WObj:=wobj0, \DOtoSensor:=RobotDO
    \DataToSensor:=RobotOut \DIfromSensor:=RobotDI
    \DataFromSensor:=RobotInposecor, EGM_FRAME_WOBJ, posesens,
    EGM_FRAME_TOOL \x:=egm_minmax_lin \y:=egm_minmax_lin
    \z:=egm_minmax_lin \rx:=egm_minmax_rot \ry:=egm_minmax_rot
    \rz:=egm_minmax_rot \LpFilter:=20;
```

Argumente

```
EGMActPose EGMid [\StreamStart] [\Tool] [\WObj] [\TLoad]
    [\DOtoSensor] [\DataToSensor] [\DIfromSensor]
    [\DataFromSensor] CorrFrame CorrFrType SensorFrame
    SensorFrType [\x] [\y] [\z] [\rx] [\ry] [\rz] [\LpFilter]
    [\SampleRate] [\MaxPosDeviation] [\MaxSpeedDeviation]
```

EGMid

Datentyp: egmident

EGM Identität.

[\StreamStart]

Datentyp: switch

StreamStart **startet das Streaming von Positionsdaten an externe Geräte. Die Daten werden mit der mit \SampleRate definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei *egm.proto*.**

Fortsetzung auf nächster Seite

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel *Externally Guided Motion* *Fortsetzung*

`StreamStart` ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

`[\Tool]`

Datentyp: `tooldata`

Das Werkzeug, das für Bewegungen verwendet wird, die mit der Instruktion `EGMRunPose` durchgeführt werden.

Das Argument `[\Tool]` ist optional. Der Standardwert, wenn das Argument ausgelassen wird, ist `tool0`.

`[\Wobj]`

Das Werkobjekt, das für Bewegungen verwendet wird, die mit der Instruktion `EGMRunPose` durchgeführt werden.

Work Object

Datentyp: `wobjdata`

Das Werkobjekt (Objekt-Koordinatensystem), auf das sich die Roboterposition in der Instruktion bezieht.

Dieses Argument kann ausgelassen werden; in diesem Fall bezieht sich die Position auf das Welt-Koordinatensystem. Wenn allerdings ein stationärer TCP oder koordinierte externe Achsen verwendet werden, muss dieses Argument angegeben werden, damit eine kreisförmige Bahn in Relation zum Werkobjekt möglich ist.

`[\TLoad]`

Die Last, die für Bewegungen verwendet wird, die mit der Instruktion `EGMRunPose` durchgeführt werden.

Total load

Datentyp: `loaddata`

Das Argument `\TLoad` beschreibt die in der Bewegung verwendete Gesamtlast. Die Gesamtlast ist die Summe der Werkzeuglast und der Nutzlast des Werkzeugs. Bei Verwendung des Arguments `\TLoad` werden die `loaddata` in den aktuellen `tooldata` nicht berücksichtigt.

Wenn das `\TLoad`-Argument auf `load0` gesetzt ist, wird das `\TLoad`-Argument nicht berücksichtigt, und stattdessen werden die `loaddata` in den aktuellen `tooldata` verwendet.

Damit das `\TLoad`-Argument verwendet werden kann, muss der Wert des Systemparameters `ModalPayloadMode` unbedingt auf 0 gesetzt werden. Wird `ModalPayloadMode` auf 0 eingestellt, kann die `GripLoad`-Instruktion nicht mehr genutzt werden.

Die Gesamtlast kann mit der Serviceroutine `LoadIdentify` ermittelt werden. Wurde der Systemparameter `ModalPayloadMode` auf 0 gesetzt, kann der Bediener beim Ausführen der Serviceroutine die `loaddata` vom Werkzeug in eine bereits vorhandene oder eine neue persistente `loaddata`-Variable kopieren.

Es ist möglich, über ein digitales Eingangssignal, das mit dem Systemeingang `SimMode` (Simulationsmodus) verbunden ist, einen Testlauf des Programms ohne

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel

Externally Guided Motion

Fortsetzung

Nutzlast durchzuführen. Wenn das digitale Eingangssignal auf 1 gesetzt wird, werden die `loaddata` im optionalen Argument `\TLoad` nicht berücksichtigt und stattdessen die `loaddata` in den aktuellen `tooldata` verwendet.



Hinweis

Die Standardfunktion für den Umgang mit Nutzlasten ist die Instruktion `GripLoad`. Deshalb beträgt der Standardwert für den Systemparameter `ModalPayloadMode` 1.

[`\DOToSensor`]

Datentyp: `signaldo`

Ein E/A-Signal, das von der Robotersteuerung an eine externe Schnittstelle, z. B. einen Sensor, gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

[`\DataToSensor`]

Datentyp: `dnum`

Ein Array mit bis zu 40 `dnum`-Werten, das von der Robotersteuerung an eine externe Schnittstelle, z. B. einen Sensor, gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

[`\DIFromSensor`]

Datentyp: `signaldi`

Ein E/A-Signal, das von einer externen Schnittstelle, z. B. ein Sensor, an die Robotersteuerung gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

[`\DataFromSensor`]

Datentyp: `dnum`

Ein Array mit bis zu 40 `dnum`-Werten, das von einer externen Schnittstelle, z. B. ein Sensor, an die Robotersteuerung gesendet wird. Die Daten werden mit der mit `\SampleRate` definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei `egm.proto`. Die Option ist nur verfügbar, wenn EGM unter Verwendung von `EGMSetupUC` eingerichtet wird, d. h. das Protokoll `UdpUc` wird für die Kommunikation mit dem externen Gerät verwendet.

Fortsetzung auf nächster Seite

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel *Externally Guided Motion* *Fortsetzung*

CorrFrame

Datentyp: pose
Korrekturkoordinatensystem.

CorrFrType

Datentyp: egmframetype
Koordinatensystemtyp des Korrekturkoordinatensystems.

SensorFrame

Datentyp: pose
Sensorkoordinatensystem.

SensFrType

Datentyp: egmframetype
Koordinatensystemtyp des Sensorkoordinatensystems.

[\x] [\y] [\z]

Datentyp: egm_minmax
Konvergenzkriterien für x, y und z in Millimeter. Der Standardwert ist $\pm 1,0$ Millimeter. Die Daten der Konvergenzkriterien können verwendet werden, um zu entscheiden, ob der Roboter die angeforderte Position in der angegebenen Achsenrichtung erreicht hat. Wenn die Differenz zwischen der angeforderten Position und der tatsächlichen Position im Bereich von `egm_minmax.min` und `egm_minmax.max` liegt, wird davon ausgegangen, dass der Roboter seine angeforderte Position erreicht hat. Wenn für eine Achsenrichtung, die in `EGMRunPose` ausgewählt wurde, keine Konvergenzkriterien angegeben sind, wird der Standardwert verwendet. Sobald alle Achsen, die in `EGMRunPose` angegeben wurden, ihre angeforderten Positionen erreicht haben, hat der Roboter selbst seine angeforderte Position erreicht, und die RAPID-Abarbeitung fährt mit der nächsten RAPID-Instruktion fort.

[\rx] [\ry] [\rz]

Datentyp: egm_minmax
Konvergenzkriterien für die Drehung x, y und z in Grad. Der Standardwert ist $\pm 0,5$ Grad. Die Daten der Konvergenzkriterien können verwendet werden, um zu entscheiden, ob der Roboter die angeforderte Orientierung entlang der angegebenen Achse erreicht hat. Wenn die Orientierung zwischen der angeforderten Orientierung und der tatsächlichen Orientierung im Bereich von `egm_minmax.min` und `egm_minmax.max` liegt, wird davon ausgegangen, dass der Roboter seine angeforderte Orientierung erreicht hat. Wenn für eine Achsenorientierung, die in `EGMRunPose` ausgewählt wurde, keine Konvergenzkriterien angegeben sind, wird der Standardwert verwendet. Sobald alle Achsenorientierungen, die in `EGMRunPose` angegeben wurden, ihre angeforderte Orientierung erreicht haben, hat der Roboter selbst seine angeforderte Orientierung erreicht und die RAPID-Abarbeitung fährt mit der nächsten RAPID-Instruktion fort.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel

Externally Guided Motion

Fortsetzung

`[\LpFilter]`

Datentyp: num

Low Pass Filter Bandwidth, in Hertz (Hz), zum Filtern von Sensorrauschen verwendet.

Der Standardwert wird aus der Konfiguration der EGMSetupXX-Instruktion übernommen.

`[\SampleRate]`

Datentyp: num

Eingangsdaten Abtastlesegeschwindigkeit als Vielfaches von 4 Millisekunden.

Gültige Werte sind 4, 8, 12, 16 usw.

Der Standardwert ist 4 Millisekunden.

`[\MaxPosDeviation]`

Datentyp: num

Maximale Achsabweichung von der einprogrammierten Position in Grad, d. h. von dem Feinpunkt, an dem die EGM-Bewegung begonnen hat. Es wird für alle Verbindungen der gleiche Wert verwendet.

Der Standardwert ist 1000 Grad.

`[\MaxSpeedDeviation]`

Datentyp: num

Maximal zulässige Achsgeschwindigkeitsveränderung in Grad/Sekunde. Alle Achsen werden in demselben Verhältnis reduziert, wenn eine Achse eingeschränkt ist.

Der Standardwert ist 1,0 Grad/Sekunde.

Einschränkungen

- Wenn EGMActPose mehrmals mit derselben EGMid ausgeführt wird, werden die aktuellsten Aktivierungsdaten für EGMRunPose-Instruktionen verwendet, die nachfolgen, bis eine neue EGMActPose durchgeführt wird.
- EGMActPose kann nur in RAPID-Bewegungstasks verwendet werden.

Syntax

```
EGMActPose
[EGMid ':='] <variable (VAR) of egmident>
['\Tool ':=' <persistent (PERS) of tooldata>]
['\Wobj ':=' <persistent (PERS) of wobjdata>]
['\TLoad ':=' <persistent (PERS) of loaddata>] ',',
['\DOToSensor ':=' <variable> (VAR) of signaldo]
['\DataToSensor ':=' <persistent> (PERS) of dnum]
['\DIFromSensor ':=' <variable> (VAR) of signaldi]
['\DataFromSensor ':=' <persistent> (PERS) of dnum]
[CorrFrame ':='] <expression (IN) of pose> ',',
[CorrFrType ':='] <expression (IN) of egmframetype> ',',
[SensorFrame ':='] <expression (IN) of pose> ',',
[SensorFrType ':='] <expression (IN) of egmframetype>
```

Fortsetzung auf nächster Seite

5.1.3 EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel

Externally Guided Motion
Fortsetzung

```
[ '\x ' := <expression (IN) of egm_minmax> ]
[ '\y ' := <expression (IN) of egm_minmax> ]
[ '\z ' := <expression (IN) of egm_minmax> ]
[ '\rx ' := <expression (IN) of egm_minmax> ]
[ '\ry ' := <expression (IN) of egm_minmax> ]
[ '\rz ' := <expression (IN) of egm_minmax> ]
[ '\LpFilter ' := <expression (IN) of num> ]
[ '\SampleRate ' := <expression (IN) of num> ]
[ '\MaxPosDeviation ' := <expression (IN) of num> ]
[ '\MaxSpeedDeviation ' := <expression (IN) of num> ] ';'

```

Weitere Informationen

Information über	Siehe
Instruktion <code>EGMRunPose</code>	EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel auf Seite 77
Instruktion <code>EGMStreamStart</code>	EGMStreamStart - Das EGM-Positions-Streaming starten. auf Seite 96
Datentyp <code>egm_minmax</code>	egm_minmax - Konvergenzkriterien für EGM auf Seite 105

5 RAPID Referenzinformation

5.1.4 EGMGetId - Erhält eine EGM Identität

Externally Guided Motion

5.1.4 EGMGetId - Erhält eine EGM Identität

Verwendung

EGMGetId wird verwendet, um eine EGM Identität zu reservieren (EGMId). Diese Identität wird dann in allen anderen EGM RAPID Instruktionen und Funktionen verwendet, um einen bestimmten EGM Prozess zu identifizieren, der mit dem RAPID Bewegungstask verknüpft ist, von dem er verwendet wird.

Eine egmident wird über ihren Namen identifiziert, d. h., ein zweiter oder dritter Aufruf von EGMGetId mit der gleichen egmident wird weder einen neuen EGM Prozess reservieren, noch den Inhalt verändern.

Zur Freigabe einer egmident zur Verwendung durch andere EGM-Prozesse muss die RAPID-Instruktion EGMRReset verwendet werden.

Es ist möglich, maximal 4 unterschiedliche EGM Identitäten gleichzeitig zu verwenden.

Grundlegende Beispiele

```
VAR egmident egmID1;  
EGMGetId egmID1;
```

Argumente

EGMGetId EGMId

EGMId

Datentyp: egmident
EGM Identität.

Einschränkungen

- EGMGetId kann nur in RAPID-Bewegungstasks verwendet werden.

Syntax

```
EGMGetId  
[EGMId ':='] <variable (VAR) of egmident> ';;'
```

Weitere Informationen

Information über	Siehe
EGMRReset	EGMRReset - Zurücksetzen eines EGM Prozesses auf Seite 73

5.1.5 EGMMoveC - Kreisförmige EGM-Bewegung mit Bahnkorrektur

Externally Guided Motion

5.1.5 EGMMoveC - Kreisförmige EGM-Bewegung mit Bahnkorrektur

Verwendung

EGMMoveC wird verwendet, um den Werkzeugarbeitspunkt (TCP) kreisförmig mit Bahnkorrektor an ein vorgegebenes Ziel zu bewegen. Während der Bewegung bleibt die Orientierung gewöhnlich in Relation zum Kreisbogen unverändert.

Grundlegende Beispiele

Das folgende Beispiel veranschaulicht die Instruktion EGMMoveC.

Beispiel 1:

```
VAR egmident EGMid1;
PERS tooldata tReg := [TRUE, [[148,0,326],
                              [0.8339007,0,0.551914,0]], [1,[0,0,100], [1,0,0,0], 0,0,0]];
PERS tooldata tLaser := [TRUE, [[148,50,326],
                              [0.3902618,-0.589657,-0.589656,0.3902630]],
                              [1,[-0.92,0,-0.39], [1,0,0,0], 0,0,0]];
EGMGetId EGMid1;
EGMSetupLTAPP ROB_1, EGMid1, "pathCorr", "OptSim", 1\LATR;
EGMActMove EGMid1, tLaser.tframe\SampleRate:=48;
MoveL p6, v10, fine, tReg\WObj:=wobj0;
EGMMoveL EGMid1, p12, v10, z5, tReg\WObj:=wobj0;
EGMMoveL EGMid1, p7, v10, z5, tReg\WObj:=wobj0;
EGMMoveC EGMid1, p13, p14, v10, z5, tReg\WObj:=wobj0;
EGMMoveL EGMid1, p15, v10, fine, tReg\WObj:=wobj0;
MoveL p8, v1000, z10, tReg\WObj:=wobj0;
EGMReset EGMid1;
```

Dieses Programm registriert einen EGM-Prozess und richtet einen Sensor als Datenquelle (Sensor) ein, der das Kommunikationsprotokoll LTAPP verwendet und den Typ *Look Ahead* hat. Der Sensor soll die Achsentypdefinition Nummer 1 für die Verfolgung verwenden. Die Rate, mit der die Steuerung auf das Gerät und das Sensorkoordinatensystem zugreift, wird ebenfalls eingerichtet.

Der Roboter wird mit einer MoveL-Instruktion zum Startpunkt des Verfolgungspfadcs bewegt. Die EGMMove-Instruktionen führen eine Roboterbewegung mit Sensorkorrekturen durch.

Zuletzt wird der Roboter in eine Ausgangsposition bewegt, und die EGM-Identität wird veröffentlicht.

Argumente

```
EGMMoveC EGMid, CirPoint, ToPoint, Speed, Zone, Tool, [\Wobj]
[\TLoad] [\NoCorr]
```

EGMid

Datentyp: egmident
EGM Identität.

CirPoint

Datentyp: robtargert

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.5 EGMMoveC - Kreisförmige EGM-Bewegung mit Bahnkorrektur

Externally Guided Motion

Fortsetzung

Der Bogenpunkt des Roboters: Der Kreispunkt ist eine Position auf dem Kreisbogen zwischen dem Startpunkt und dem Zielpunkt. Für größte Genauigkeit sollte er ungefähr in der Mitte zwischen Start- und Zielpunkt platziert werden. Wenn er zu nahe an den Start- oder Zielpunkt gesetzt wird, gibt das Robotersystem evtl. eine Warnung aus. Der Bogenpunkt wird als benannte Position definiert oder direkt in der Instruktion gespeichert (mit einem * gekennzeichnet). Die Position der externen Achsen wird nicht verwendet.

ToPoint

Datentyp: `robtarg`

Der Zielpunkt des Roboters und der externen Achsen. Er wird als benannte Position definiert oder direkt in der Instruktion gespeichert (mit einem * gekennzeichnet).

Speed

Datentyp: `speeddata`

Die Geschwindigkeitsdaten, die für Bewegungen gelten. Geschwindigkeitsdaten definieren die Geschwindigkeit des TCP, der Werkzeugumorientierung und der externen Achsen.

Zone

Datentyp: `zonedata`

Zonendaten für die Bewegung. Zonendaten beschreiben die Größe der erzeugten Zonenbahn.

Tool

Datentyp: `tooldata`

Das verwendete Werkzeug, wenn sich der Roboter bewegt. Der Werkzeugarbeitspunkt ist der Punkt, der an den angegebenen Zielpunkt bewegt wird.

[\Wobj]

Work Object

Datentyp: `wobjdata`

Das Werkobjekt (Objekt-Koordinatensystem), auf das sich die Roboterposition in der Instruktion bezieht.

Dieses Argument kann ausgelassen werden; in diesem Fall bezieht sich die Position auf das Welt-Koordinatensystem. Wenn allerdings ein stationärer TCP oder koordinierte externe Achsen verwendet werden, muss dieses Argument angegeben werden, damit eine kreisförmige Bahn in Relation zum Werkobjekt möglich ist.

[\TLoad]

Total load

Datentyp: `loaddata`

Das Argument `\TLoad` beschreibt die in der Bewegung verwendete Gesamtlast. Die Gesamtlast ist die Summe der Werkzeuglast und der Nutzlast des Werkzeugs. Bei Verwendung des Arguments `\TLoad` werden die `loaddata` in den aktuellen `tooldata` nicht berücksichtigt.

Fortsetzung auf nächster Seite

5.1.5 EGMMoveC - Kreisförmige EGM-Bewegung mit Bahnkorrektur Externally Guided Motion Fortsetzung

Wenn das `\TLoad`-Argument auf `load0` gesetzt ist, wird das `\TLoad`-Argument nicht berücksichtigt, und stattdessen werden die `loaddata` in den aktuellen `tooldata` verwendet.

Damit das `\TLoad`-Argument verwendet werden kann, muss der Wert des Systemparameters `ModalPayLoadMode` unbedingt auf 0 gesetzt werden. Wird `ModalPayLoadMode` auf 0 eingestellt, kann die `GripLoad`-Instruktion nicht mehr genutzt werden.

Die Gesamtlast kann mit der Serviceroutine `LoadIdentify` ermittelt werden. Wurde der Systemparameter `ModalPayLoadMode` auf 0 gesetzt, kann der Bediener beim Ausführen der Serviceroutine die `loaddata` vom Werkzeug in eine bereits vorhandene oder eine neue persistente `loaddata`-Variable kopieren.

Es ist möglich, über ein digitales Eingangssignal, das mit dem Systemeingang `SimMode` (Simulationsmodus) verbunden ist, einen Testlauf des Programms ohne Nutzlast durchzuführen. Wenn das digitale Eingangssignal auf 1 gesetzt wird, werden die `loaddata` im optionalen Argument `\TLoad` nicht berücksichtigt und stattdessen die `loaddata` in den aktuellen `tooldata` verwendet.



Hinweis

Die Standardfunktion für den Umgang mit Nutzlasten ist die Instruktion `GripLoad`. Deshalb beträgt der Standardwert für den Systemparameter `ModalPayLoadMode` 1.

`[\NoCorr]`

Datentyp: `switch`

Die Bahnkorrektur ist abgeschaltet.

Programmabarbeitung

`EGMMoveC` bewegt den Roboter mit überlagerten Sensorkorrekturen entlang einer programmierten kreisförmigen Bahn. Während der Bewegung fordert die Instruktion Korrekturdaten vom Sensor mit der mit `EGMActMove` festgelegten Rate an. Wenn das optionale Argument `\NoCorr` vorhanden ist, wird der programmierten Bahn keine Korrektur hinzugefügt.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_UDPUC_COMM</code>	<p>Bei der Kommunikation mit dem <code>UdpUc</code>-Gerät ist ein Fehler aufgetreten.</p> <p>Im Synchron-Modus ist <code>ERR_UDPUC_COMM</code> immer ein behebbarer Fehler, der mit einer Fehlerbehandlung behoben werden kann.</p> <p>Im asynchronen Modus (<code>EGMRunX\NoWait</code>) wird <code>ERR_UDPUC_COMM</code> immer als schwerwiegender Fehler gemeldet, wenn EGM ausgeführt wird.</p>

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.5 EGMMoveC - Kreisförmige EGM-Bewegung mit Bahnkorrektur

Externally Guided Motion

Fortsetzung

Einschränkungen

- EGMMoveC kann nur in RAPID-Bewegungstasks verwendet werden.
- EGMMoveC kann nicht in einer UNDO-Behandlung oder RAPID-Routine ausgeführt werden, die mit einem der folgenden speziellen Systemereignisse verknüpft ist: PowerOn, Stop, QStop, Restart, Reset oder Step.

Syntax

```
EGMMoveC
  [GMid ':='] <variable (VAR) of egmident> ','
  [CirPoint ':='] < expression (IN) of robtargt> ','
  [ToPoint ':='] < expression (IN) of robtargt> ','
  [Speed ':='] < expression (IN) of speeddata> ','
  [Zone ':='] < expression (IN) of zonedata> ','
  [Tool ':='] < persistent (PERS) of tooldata>
  ['\WObj ':=' < persistent (PERS) of wobjdata>]
  ['\TLoad ':=' < persistent (PERS) of loaddata>]
  ['\NoCorr] ';'

```

Weitere Informationen

Information über	Siehe
MoveC	<i>Technisches Referenzhandbuch - RAPID Instruktionen, Funktionen und Datentypen</i>

5.1.6 EGMMoveL - Lineare EGM-Bewegung mit Bahnkorrektur

Verwendung

EGMMoveL wird verwendet, um den Werkzeugarbeitspunkt (TCP) linear mit Bahnkorrektur an ein vorgegebenes Ziel zu bewegen. Wenn der TCP stationär bleiben soll, kann mit dieser Instruktion auch das Werkzeug umorientiert werden.

Grundlegende Beispiele

Das folgende Beispiel veranschaulicht die Instruktion EGMMoveL.

Beispiel 1:

```
VAR egmident EGMid1;
PERS tooldata tReg := [TRUE, [[148,0,326],
[0.8339007,0,0.551914,0]], [1,[0,0,100], [1,0,0,0], 0,0,0]];
PERS tooldata tLaser := [TRUE, [[148,50,326],
[0.3902618,-0.589657,-0.589656,0.3902630]],
[1,[-0.92,0,-0.39], [1,0,0,0], 0,0,0]];
EGMGetId EGMid1;
EGMSetupLTAPP ROB_1, EGMid1, "pathCorr", "OptSim", 1\LATR;
EGMActMove EGMid1, tLaser.tframe\SampleRate:=48;
MoveL p6, v10, fine, tReg\WObj:=wobj0;
EGMMoveL EGMid1, p12, v10, z5, tReg\WObj:=wobj0;
EGMMoveL EGMid1, p7, v10, z5, tReg\WObj:=wobj0;
EGMMoveC EGMid1, p13, p14, v10, z5, tReg\WObj:=wobj0;
EGMMoveL EGMid1, p15, v10, fine, tReg\WObj:=wobj0;
MoveL p8, v1000, z10, tReg\WObj:=wobj0;
EGMReset EGMid1;
```

Dieses Programm registriert einen EGM-Prozess und richtet einen Sensor als Datenquelle (Sensor) ein, der das Kommunikationsprotokoll LTAPP verwendet und den Typ *Look Ahead* hat. Der Sensor soll die Achsentypdefinition Nummer 1 für die Verfolgung verwenden. Die Rate, mit der die Steuerung auf das Gerät und das Sensorkoordinatensystem zugreift, wird ebenfalls eingerichtet.

Der Roboter wird mit einer MoveL-Instruktion zum Startpunkt des Verfolgungspfadcs bewegt. Die EGMMove-Instruktionen führen eine Roboterbewegung mit Sensorkorrekturen durch.

Zuletzt wird der Roboter in eine Ausgangsposition bewegt, und die EGM-Identität wird veröffentlicht.

Argumente

```
EGMMoveL EGMid, ToPoint, Speed, Zone, Tool, [\Wobj] [\TLoad]
[\NoCorr]
```

EGMid

Datentyp: egmident
EGM Identität.

ToPoint

Datentyp: robtargert

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.6 EGMMoveL - Lineare EGM-Bewegung mit Bahnkorrektur

Externally Guided Motion

Fortsetzung

Der Zielpunkt des Roboters und der externen Achsen. Er wird als benannte Position definiert oder direkt in der Instruktion gespeichert (mit einem * gekennzeichnet).

Speed

Datentyp: speeddata

Die Geschwindigkeitsdaten, die für Bewegungen gelten. Geschwindigkeitsdaten definieren die Geschwindigkeit des TCP, der Werkzeugumorientierung und der externen Achsen.

Zone

Datentyp: zonedata

Zonendaten für die Bewegung. Zonendaten beschreiben die Größe der erzeugten Zonenbahn.

Tool

Datentyp: tooldata

Das verwendete Werkzeug, wenn sich der Roboter bewegt. Der Werkzeugarbeitspunkt ist der Punkt, der an den angegebenen Zielpunkt bewegt wird.

[\WObj]

Work Object

Datentyp: wobjdata

Das Werkobjekt (Objekt-Koordinatensystem), auf das sich die Roboterposition in der Instruktion bezieht.

Dieses Argument kann ausgelassen werden; in diesem Fall bezieht sich die Position auf das Welt-Koordinatensystem. Wenn allerdings ein stationärer TCP oder koordinierte externe Achsen verwendet werden, muss dieses Argument angegeben werden, damit eine kreisförmige Bahn in Relation zum Werkobjekt möglich ist.

[\TLoad]

Total load

Datentyp: loaddata

Das Argument \TLoad beschreibt die in der Bewegung verwendete Gesamtlast. Die Gesamtlast ist die Summe der Werkzeuglast und der Nutzlast des Werkzeugs. Bei Verwendung des Arguments \TLoad werden die loaddata in den aktuellen tooldata nicht berücksichtigt.

Wenn das \TLoad-Argument auf load0 gesetzt ist, wird das \TLoad-Argument nicht berücksichtigt, und stattdessen werden die loaddata in den aktuellen tooldata verwendet.

Damit das \TLoad-Argument verwendet werden kann, muss der Wert des Systemparameters ModalPayLoadMode unbedingt auf 0 gesetzt werden. Wird ModalPayLoadMode auf 0 eingestellt, kann die GripLoad-Instruktion nicht mehr genutzt werden.

Die Gesamtlast kann mit der Serviceroutine LoadIdentify ermittelt werden. Wurde der Systemparameter ModalPayLoadMode auf 0 gesetzt, kann der Bediener beim

Fortsetzung auf nächster Seite

Ausführen der Serviceroutine die `loaddata` vom Werkzeug in eine bereits vorhandene oder eine neue persistente `loaddata`-Variable kopieren.

Es ist möglich, über ein digitales Eingangssignal, das mit dem Systemeingang `SimMode` (Simulationsmodus) verbunden ist, einen Testlauf des Programms ohne Nutzlast durchzuführen. Wenn das digitale Eingangssignal auf 1 gesetzt wird, werden die `loaddata` im optionalen Argument `\TLoad` nicht berücksichtigt und stattdessen die `loaddata` in den aktuellen `tooldata` verwendet.



Hinweis

Die Standardfunktion für den Umgang mit Nutzlasten ist die Instruktion `GripLoad`. Deshalb beträgt der Standardwert für den Systemparameter `ModalPayloadMode` 1.

`[\NoCorr]`

Datentyp: `switch`

Die Bahnkorrektur ist abgeschaltet.

Programmabarbeitung

`EGMMoveL` bewegt den Roboter mit überlagerten Sensorkorrekturen entlang einer programmierten linearen Bahn. Während der Bewegung fordert die Instruktion Korrekturdaten vom Sensor mit der mit `EGMActMove` festgelegten Rate an. Wenn das optionale Argument `\NoCorr` vorhanden ist, wird der programmierten Bahn keine Korrektur hinzugefügt.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_UDPUC_COMM</code>	<p>Bei der Kommunikation mit dem <code>UdpUc</code>-Gerät ist ein Fehler aufgetreten.</p> <p>Im Synchron-Modus ist <code>ERR_UDPUC_COMM</code> immer ein behebbarer Fehler, der mit einer Fehlerbehandlung behoben werden kann.</p> <p>Im asynchronen Modus (<code>EGMRunX\NoWait</code>) wird <code>ERR_UDPUC_COMM</code> immer als schwerwiegender Fehler gemeldet, wenn EGM ausgeführt wird.</p>

Einschränkungen

- `EGMMoveL` kann nur in RAPID-Bewegungstasks verwendet werden.
- `EGMMoveL` kann nicht in einer UNDO-Behandlung oder RAPID-Routine ausgeführt werden, die mit einem der folgenden speziellen Systemereignisse verknüpft ist: `PowerOn`, `Stop`, `QStop`, `Restart`, `Reset` oder `Step`.

Syntax

```
EGMMoveL
    [EGMid '[:='] <variable (VAR) of egmident> ',']
```

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.6 EGMMoveL - Lineare EGM-Bewegung mit Bahnkorrektur

Externally Guided Motion

Fortsetzung

```
[ToPoint ':='] < expression (IN) of robtargget> ', '  
[Speed ':='] < expression (IN) of speeddata> ', '  
[Zone ':='] < expression (IN) of zonedata> ', '  
[Tool ':='] < persistent (PERS) of tooldata>  
['\WObj ':='] < persistent (PERS) of wobjdata>  
['\TLoad ':='] < persistent (PERS) of loaddata>  
['\NoCorr] ';'
```

Weitere Informationen

Information über	Siehe
MoveL	<i>Technisches Referenzhandbuch - RAPID Instruktionen, Funktionen und Datentypen</i>

5.1.7 EGMReset - Zurücksetzen eines EGM Prozesses

Verwendung

EGMReset setzt einen spezifischen EGM Prozess zurück (EGMId), d. h., die Reservierung wird aufgehoben.

Grundlegende Beispiele

```
VAR egmident egmID1;
PERS pose pose1:[[0,0,0], [1,0,0,0]];
CONST egm_minmax egm_minmax_lin:[-0.1,0.1];
CONST egm_minmax egm_minmax_rot:[-0.1,0.2];
CONST pose posecor:[[1200,400,900], [0,0,1,0]];
CONST pose posesens:[[12.3313,-0.108707,416.142],
[0.903899,-0.00320735,0.427666,0.00765917]];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_01
\aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
\aiR6rz:=ai_06;
EGMActPose egmID1 \Tool:=tool0 \WObj:=wobj0, posecor,
EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin
\y:=egm_minmax_lin \z:=egm_minmax_lin \rx:=egm_minmax_rot
\ry:=egm_minmax_rot \rz:=egm_minmax_rot \LpFilter:=20;
EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \rx \ry \rz
\RampInTime:=0.05;
EGMReset egmID1;
```

Argumente

EGMReset EGMId

EGMId

Datentyp: egmident
EGM Identität.

Syntax

```
EGMReset
[EGMId '[:'] <variable (VAR) of egmident>'];
```

5 RAPID Referenzinformation

5.1.8 EGMRUNJOINT - Ausführen einer EGM Bewegung mit einer Achswinkelposition *Externally Guided Motion*

5.1.8 EGMRUNJOINT - Ausführen einer EGM Bewegung mit einer Achswinkelposition

Verwendung

EGMRUNJOINT führt eine sensorgeführte Bewegung zu einer Achsposition von einem Feinpunkt für einen spezifischen EGM Prozess durch (EGMID) und definiert, welches Gelenk bewegt wird.

Grundlegende Beispiele

```
VAR egmident egmID1;
PERS pose pose1:=[[0,0,0],[1,0,0,0]];
CONST egm_minmax egm_minmax1:=[-1,1];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Joint \aiR1x:=ai_01
\aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
\aiR6rz:=ai_06;
EGMActJoint egmID1, \J1:=egm_minmax1 \J3:=egm_minmax1
\J4:=egm_minmax1;
EGMRUNJOINT egmID1, EGM_STOP_HOLD \J1 \J3 \RampInTime:=0.05;
```

Argumente

```
EGMRUNJOINT EGMID, Mode [\NoWaitCond] [\J1] [\J2] [\J3] [\J4] [\J5]
[\J6] [\J7] [\CondTime] [\RampInTime] [\RampOutTime]
[\PosCorrGain]
```

EGMID

Datentyp: egmident

EGM Identität.

Mode

Datentyp: egmstopmode

Definiert, wie die Bewegung beendet wird (EGM_STOP_HOLD, EGM_STOP_RAMP_DOWN)

[\NoWaitCond]

Datentyp: switch

Wenn dieses optionale Argument verwendet wird, gibt EGMRUNJOINT den Programmzeiger RAPID frei, bevor die Bewegung abgeschlossen ist. Anschließend muss die RAPID-Instruktion EGMWaitCond verwendet werden, um die Bewegung EGM Position Guidance abzuschließen. Zwischen EGMRUNJOINT und EGMWaitCond ist keine andere Bewegungsinstruktion erlaubt.

[\J1] [\J2] [\J3] [\J4] [\J5] [\J6] [\J7]

Datentyp: switch

Bewegen Sie Achse 1 bis 6 für 6-achsige Roboter und Achse 1 bis 7 für die 7-achsigen Roboter.

[\CondTime]

Datentyp: num

Fortsetzung auf nächster Seite

5.1.8 EGMRUNJOINT - Ausführen einer EGM Bewegung mit einer Achswinkelposition Externally Guided Motion Fortsetzung

Die Zeit in Sekunden, in der die Konvergenzkriterien, die in `EGMActJoint` definiert wurden, erfüllt sein müssen, bevor der Zielpunkt als erreicht angesehen wird und `EGMRUNJOINT` die RAPID-Abarbeitung freigibt, um zur nächsten Instruktion zu gehen.

Der Standardwert ist 1s.

`[\RampInTime]`

Datentyp: num

Definiert in Sekunden, wie schnell die Bewegung gestartet wird.

`[\RampOutTime]`

Datentyp: num

Definiert in Sekunden, wie schnell ein Heruntersetzen von EGM ausgeführt wird.

Dieser Parameter ist ohne Bedeutung, wenn der Parameter `Mode` auf `EGM_STOP_HOLD` gestellt wurde.

`[\PosCorrGain]`

Datentyp: num

Positionskorrekturverstärkung. Ein Wert zwischen 0 und 1, Standard 1.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_UDPUC_COMM</code>	<p>Bei der Kommunikation mit dem UdpUc-Gerät ist ein Fehler aufgetreten.</p> <p>Im Synchron-Modus ist <code>ERR_UDPUC_COMM</code> immer ein behebbarer Fehler, der mit einer Fehlerbehandlung behoben werden kann.</p> <p>Im asynchronen Modus (<code>EGMRUNX\NoWait</code>) wird <code>ERR_UDPUC_COMM</code> immer als schwerwiegender Fehler gemeldet, wenn EGM ausgeführt wird.</p>

Einschränkungen

- Vor der ersten Verwendung von `EGMRUNJOINT` muss der Roboter nach dem Start der Steuerung bewegt worden sein, indem eine `Move`-Instruktion über RAPID ausgeführt wird.
- Der Startpunkt für eine `EGMRUNJOINT` Bewegung muss ein Feinpunkt sein.
- `EGMRUNJOINT` kann nur in RAPID-Bewegungstasks verwendet werden.
- Wenn die Instruktion `EGMActPose` anstelle von `EGMActJoint` ausgeführt wurde, tritt der folgende schwerwiegende Fehler auf: **41826 EGM mode mismatch**.
- Wenn keiner der Switches `\J1` bis `\J7` angegeben ist, erfolgt keine Bewegung und die RAPID-Abarbeitung geht zur nächsten RAPID-Instruktion.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.8 EGMRunJoint - Ausführen einer EGM Bewegung mit einer Achswinkelposition

Externally Guided Motion

Fortsetzung

Syntax

```
EGMRunJoint
[EGMid ':'='] <variable (VAR) of egmident> ','
[Mode ':'='] <expression (IN) of egmstopmode>
['\NoWaitCond]
['\J1]
['\J2]
['\J3]
['\J4]
['\J5]
['\J6]
['\J7]
['\CondTime ':'='] <expression (IN) of num>]
['\RampInTime ':'='] <expression (IN) of num>]
['\RampOutTime ':'='] <expression (IN) of num>]
['\PosCorrGain ':'='] <expression (IN) of num>] ';' ;'
```

Weitere Informationen

Information über	Siehe
egmstopmode	egmstopmode - Definiert die Stopp-Modi für EGM auf Seite 108
MoveJ	<i>Technisches Referenzhandbuch - RAPID Instruktionen, Funktionen und Datentypen</i>

5.1.9 EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel

Externally Guided Motion

5.1.9 EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel

Verwendung

EGMRunPose führt eine sensorgeführte Bewegung zu einem Positionsziel von einem Feinpunkt für einen spezifischen EGM Prozess durch (EGMid) und definiert, welche Richtungen und Orientierungen verändert werden können.

Grundlegende Beispiele

```
VAR egmident egmID1;
PERS pose pose1:=[[0,0,0],[1,0,0,0]];
CONST egm_minmax egm_minmax_lin:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot:=[-0.1,0.2];
CONST pose posecor:=[[1200,400,900],[0,0,1,0]];
CONST pose posesens:=[[12.3313,-0.108707,416.142],
[0.903899,-0.00320735,0.427666,0.00765917]];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_01
\aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
\aiR6rz:=ai_06;
EGMActPose egmID1 \Tool:=tool0 \Wobj:=wobj0, posecor,
EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin
\y:=egm_minmax_lin \z:=egm_minmax_lin \rx:=egm_minmax_rot
\ry:=egm_minmax_rot \rz:=egm_minmax_rot \LpFilter:=20;
EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \rx \ry \rz
\RampInTime:=0.05;
```

Argumente

```
EGMRunPose EGMid, Mode [\NoWaitCond] [\x] [\y] [\z] [\rx] [\ry]
[\rz] [\CondTime] [\RampInTime] [\RampOutTime] [\Offset]
[\PosCorrGain]
```

EGMid

Datentyp: egmident

EGM Identität.

Mode

Datentyp: egmstopmode

Definiert, wie die Bewegung beendet wird (EGM_STOP_HOLD, EGM_STOP_RAMP_DOWN)

[\NoWaitCond]

Datentyp: switch

Wenn dieses optionale Argument verwendet wird, gibt EGMRunPose den Programmzeiger RAPID frei, bevor die Bewegung abgeschlossen ist. Anschließend muss die RAPID-Instruktion EGMWaitCond verwendet werden, um die Bewegung EGM Position Guidance abzuschließen. Zwischen EGMRunPose und EGMWaitCond ist keine andere Bewegungsinstruktion erlaubt.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.9 EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel

Externally Guided Motion

Fortsetzung

`[\x] [\y] [\z]`

Datentyp: `switch`

Bewegung in Richtung x, y und z.

`[\rx] [\ry] [\rz]`

Datentyp: `switch`

Umorientierung um die Achsen x, y und z.

`[\CondTime]`

Datentyp: `num`

Die Zeit in Sekunden, in der die Konvergenzkriterien, die in `EGMActPose` definiert wurden, erfüllt sein müssen, bevor der Zielpunkt als erreicht angesehen wird und `EGMRunPose` die RAPID-Abarbeitung freigibt, um zur nächsten Instruktion zu gehen.

Der Standardwert ist 1s.

`[\RampInTime]`

Datentyp: `num`

Definiert in Sekunden, wie schnell die Bewegung gestartet wird.

`[\RampOutTime]`

Datentyp: `num`

Definiert in Sekunden, wie schnell ein Heruntersetzen von EGM ausgeführt wird.

Dieser Parameter ist ohne Bedeutung, wenn der Parameter `Mode` auf `EGM_STOP_HOLD` gestellt wurde.

`[\Offset]`

Datentyp: `pose`

Möglichkeit zur Definition eines statischen Offsets zusätzlich zum Wert, der vom Sensor angegeben wird.

`[\PosCorrGain]`

Datentyp: `num`

Positionskorrekturverstärkung. Ein Wert zwischen 0 und 1, Standard 1.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_UDPUC_COMM</code>	Bei der Kommunikation mit dem UdpUc-Gerät ist ein Fehler aufgetreten. Im Synchron-Modus ist <code>ERR_UDPUC_COMM</code> immer ein behebbarer Fehler, der mit einer Fehlerbehandlung behoben werden kann. Im asynchronen Modus (<code>EGMRunX\NoWait</code>) wird <code>ERR_UDPUC_COMM</code> immer als schwerwiegender Fehler gemeldet, wenn EGM ausgeführt wird.

Fortsetzung auf nächster Seite

5.1.9 EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel

Externally Guided Motion
Fortsetzung

Einschränkungen

- Vor der ersten Verwendung von `EGMRunPose` muss der Roboter nach dem Start der Steuerung bewegt worden sein, indem eine `Move`-Instruktion über `RAPID` ausgeführt wird.
- Der Startpunkt für eine `EGMRunPose` Bewegung muss ein Feinpunkt sein.
- `EGMRunPose` kann nur in `RAPID`-Bewegungstasks verwendet werden.
- Wenn die Instruktion `EGMActJoint` anstelle von `EGMActPose` ausgeführt wurde, tritt der folgende schwerwiegende Fehler auf: *41826 EGM mode mismatch*.
- Wenn keiner der Switches `\x` bis `\rz` angegeben ist, erfolgt keine Bewegung und die `RAPID`-Abarbeitung geht zur nächsten `RAPID`-Instruktion.

Syntax

```
EGMRunPose
[EGMid ':='] <variable (VAR) of egmident>','
[Mode ':='] <expression (IN) of egmstopmode>
['\NoWaitCond]
['\x]
['\y]
['\z]
['\rx]
['\ry]
['\rz]
['\CondTime ':=' <expression (IN) of num>]
['\RampInTime ':=' <expression (IN) of num>]
['\RampOutTime ':=' <expression (IN) of num>]
['\Offset ':=' <expression (IN) of pose>]
['\PosCorrGain ':=' <expression (IN) of num>] ';'

```

Weitere Informationen

Information über	Siehe
egmstopmode	egmstopmode - Definiert die Stopp-Modi für EGM auf Seite 108

5 RAPID Referenzinformation

5.1.10 EGMSetupAI - Einrichtung analoger Eingangssignale für EGM

Externally Guided Motion

5.1.10 EGMSetupAI - Einrichtung analoger Eingangssignale für EGM

Verwendung

EGMSetupAI wird verwendet, um analoge Eingangssignale für einen spezifischen EGM Prozess einzurichten (EGMID), die als Quelle für die Positionszielwerte dienen, nach denen sich der Roboter und bis zu 6 zusätzliche Achsen richten.

EGM-Achsenmodus ist der einzige EGM-Modus, der 7-achsige Roboter unterstützt. Bei 7-achsigen Robotern bietet der Eingang der ersten zusätzlichen Achse die Position für die weiteren Roboterachsen.

Grundlegende Beispiele

```
VAR egmident egmID1;

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_01
\aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
\aiR6rz:=ai_06;
```

Argumente

```
EGMSetupAI MecUnit, EGMID, ExtConfigName [\Joint] | [\Pose] |
[\PathCorr] [\APTR] [\aiR1x] [\aiR2y] [\aiR3z] [\aiR4rx]
[\aiR5ry] [\aiR6rz] [\aiE1] [\aiE2] [\aiE3] [\aiE4] [\aiE5]
[\aiE6]
```

MecUnit

Datentyp: mecunit

Name der mechanischen Einheit.

EGMID

Datentyp: egmident

EGM Identität.

ExtConfigName

Datentyp: string

Der Name der "External Motion Interface Data" (Externe Bewegungsschnittstellen-Daten) entsprechend der im System definierten Parameter. Für weitere Informationen siehe *Technisches Referenzhandbuch - Systemparameter, Typ External Motion Interface Data*, Thema *Motion*.

[\Joint]

Datentyp: switch

Wählt Achsenbewegung zur Positionsführung.

Mindestens einer der Schalter \Joint, \Pose oder \PathCorr muss vorhanden sein.

[\Pose]

Datentyp: switch

Fortsetzung auf nächster Seite

Wählt Positionsbewegung zur Positionsführung.

Mindestens einer der Schalter `\Joint`, `\Pose` oder `\PathCorr` muss vorhanden sein.

`[\PathCorr]`

Datentyp: `switch`

Wählt die Bahnkorrektur.

Mindestens einer der Schalter `\Joint`, `\Pose` oder `\PathCorr` muss vorhanden sein.

`[\APTR]`

Datentyp: `switch`

Einrichten eines Sensors des Typs At-Point-Tracker zur Bahnkorrektur. Zum Beispiel `WeldGuide` oder `AWC`.

`\APTR` muss vorhanden sein, wenn `\PathCorr` verwendet wird.

`[\aiR1x] [\aiR2y] [\aiR3z]`

Datentyp: `signalai`

Legt das Signal fest, das die Werte x, y und z in Millimetern für die Positionsbewegung bereitstellt.

Legt das Signal fest, das die Winkel der Roboter Verbindung 1 bis 3 in Grad für die Verbindungsbewegung bereitstellt.

`[\aiR4rx] [\aiR5ry] [\aiR6rz]`

Datentyp: `signalai`

Legt das Signal fest, das die Werte für die Drehung x, y und z des Roboters in Grad für die Positionsbewegung bereitstellt.

Legt das Signal fest, das die Winkel der Roboter Verbindung 4 bis 6 in Grad für die Verbindungsbewegung bereitstellt.

`[\aiE1] [\aiE2] [\aiE3] [\aiE4] [\aiE5] [\aiE6]`

Datentyp: `signalai`

Legt das Signal fest, das die Position der zusätzlichen Achsen 1 bis 6 bereitstellt.

Bei Verwendung des EGM-Achsenmodus bei einem 7-achsigen Roboter liefert `\aiE1` die Position für die zusätzlichen Roboterachsen.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_NO_ALIASIO_DEF</code>	Die Signalvariable ist eine in RAPID angegebene Variable. Sie wurde nicht an ein E/A-Signal, das in der E/A-Konfiguration mit Instruktion <code>AliasIO</code> definiert wurde.
<code>ERR_NORUNUNIT</code>	Es ist kein Kontakt mit dem E/A-Gerät vorhanden.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.10 EGMSetupAI - Einrichtung analoger Eingangssignale für EGM

Externally Guided Motion

Fortsetzung

Name	Fehlerursache
ERR_SIG_NOT_VALID	Auf das E/A-Signal kann nicht zugegriffen werden (gilt nur für ICI-Feldbus).

Einschränkungen

- EGMSetupAI kann nur in RAPID-Bewegungstasks verwendet werden.
- Die mechanische Einheit muss ein TCP-Roboter sein.
- Mindestens ein Signal muss angegeben werden, sonst wird ein Fehler gesendet und die RAPID-Abarbeitung wird angehalten.

Syntax

```
EGMSetupAI
[ MecUnit ':' '=' <variable (VAR) of mecunit> ','
[ EGMid ':' '=' <variable (VAR) of egmident> ','
[ ExtConfigName ':' '=' <expression (IN) of string>
[ ['\"Joint\" | ['\"Pose\" | ['\"PathCorr\" ]
[ '\"APTR\"
[ '\"aiR1x ':' '=' <variable (VAR) of signalai>
[ '\"aiR2y ':' '=' <variable (VAR) of signalai>
[ '\"aiR3z ':' '=' <variable (VAR) of signalai>
[ '\"aiR4rx ':' '=' <variable (VAR) of signalai>
[ '\"aiR5ry ':' '=' <variable (VAR) of signalai>
[ '\"aiR6rz ':' '=' <variable (VAR) of signalai>
[ '\"aiE1 ':' '=' <variable (VAR) of signalai>
[ '\"aiE2 ':' '=' <variable (VAR) of signalai>
[ '\"aiE3 ':' '=' <variable (VAR) of signalai>
[ '\"aiE4 ':' '=' <variable (VAR) of signalai>
[ '\"aiE5 ':' '=' <variable (VAR) of signalai>
[ '\"aiE6 ':' '=' <variable (VAR) of signalai> ] ';' ]
```

Weitere Informationen

Information über	Siehe
Systemparameter	Systemparameter auf Seite 43

5.1.11 EGMSetupAO - Einrichtung analoger Ausgangssignale für EGM

Verwendung

EGMSetupAO wird verwendet, um AO Signale für einen spezifischen EGM Prozess einzurichten (EGMId), die als Quelle für die Positionszielwerte dienen, nach denen sich der Roboter und bis zu 6 zusätzliche Achsen richten.

EGM-Achsenmodus ist der einzige EGM-Modus, der 7-achsige Roboter unterstützt. Bei 7-achsigen Robotern bietet der Eingang der ersten zusätzlichen Achse die Position für die weiteren Roboterachsen.

Grundlegende Beispiele

```
VAR egmident egmID1;

EGMGetId egmID1;
EGMSetupAO ROB_1, egmID1, "default" \Pose \aoR1x:=ao_01
\aoR2y:=ao_02 \aoR3z:=ao_03 \aoR4rx:=ao_04 \aoR5ry:=ao_05
\aoR6rz:=ao_06;
```

Argumente

```
EGMSetupAO MecUnit, EGMId, ExtConfigName [\Joint] | [\Pose] |
[\PathCorr] [\APTR] [\aoR1x] [\aoR2y] [\aoR3z] [\aoR4rx]
[\aoR5ry] [\aoR6rz] [\aoE1] [\aoE2] [\aoE3] [\aoE4] [\aoE5]
[\aoE6]
```

MecUnit

Datentyp: mecunit

Name der mechanischen Einheit.

EGMId

Datentyp: egmident

EGM Identität.

ExtConfigName

Datentyp: string

Der Name der "External Motion Interface Data" (Externe Bewegungsschnittstellen-Daten) entsprechend der im System definierten Parameter. Für weitere Informationen siehe *Technisches Referenzhandbuch - Systemparameter, Typ External Motion Interface Data, Thema Motion*.

[\Joint]

Datentyp: switch

Wählt Achsenbewegung.

Mindestens einer der Schalter \Joint oder \Pose muss vorhanden sein.

[\Pose]

Datentyp: switch

Wählt Positionsbewegung.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.11 EGMSetupAO - Einrichtung analoger Ausgangssignale für EGM

Externally Guided Motion

Fortsetzung

Mindestens einer der Schalter `\Joint` oder `\Pose` muss vorhanden sein.

`[\PathCorr]`

Datentyp: `switch`

Wählt die Bahnkorrektur.

Mindestens einer der Schalter `\Joint`, `\Pose` oder `\PathCorr` muss vorhanden sein.

`[\APTR]`

Datentyp: `switch`

Einrichten eines Sensors des Typs At-Point-Tracker zur Bahnkorrektur. Zum Beispiel `WeldGuide` oder `AWC`.

`\APTR` muss vorhanden sein, wenn `\PathCorr` verwendet wird.

`[\aoR1x] [\aoR2y] [\aoR3z]`

Datentyp: `signalao`

Legt das Signal fest, das die Werte x, y und z in Millimetern für die Positionsbewegung bereitstellt.

Legt das Signal fest, das die Winkel der Roboter Verbindung 1 bis 3 in Grad für die Verbindungsbewegung bereitstellt.

`[\aoR4rx] [\aoR5ry] [\aoR6rz]`

Datentyp: `signalao`

Legt das Signal fest, das die Werte für die Drehung x, y und z des Roboters in Grad für die Positionsbewegung bereitstellt.

Legt das Signal fest, das die Winkel der Roboter Verbindung 4 bis 6 in Grad für die Verbindungsbewegung bereitstellt.

`[\aoE1] [\aoE2] [\aoE3] [\aoE4] [\aoE5] [\aoE6]`

Datentyp: `signalao`

Legt das Signal fest, das die Position der zusätzlichen Achsen 1 bis 6 bereitstellt.

Bei Verwendung des EGM-Achsenmodus bei einem 7-achsigen Roboter liefert `\aoE1` die Position für die zusätzlichen Roboterachsen.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_NO_ALIASIO_DEF</code>	Die Signalvariable ist eine in RAPID angegebene Variable. Sie wurde nicht an ein E/A-Signal, das in der E/A-Konfiguration mit Instruktion <code>AliasIO</code> definiert wurde.
<code>ERR_NORUNUNIT</code>	Es ist kein Kontakt mit dem E/A-Gerät vorhanden.
<code>ERR_SIG_NOT_VALID</code>	Auf das E/A-Signal kann nicht zugegriffen werden (gilt nur für ICI-Feldbus).

Fortsetzung auf nächster Seite

5.1.11 EGMSetupAO - Einrichtung analoger Ausgangssignale für EGM

Externally Guided Motion
Fortsetzung

Einschränkungen

- EGMSetupAO kann nur in RAPID-Bewegungstasks verwendet werden.
- Die mechanische Einheit muss ein TCP-Roboter sein.
- Mindestens ein Signal muss angegeben werden, sonst wird ein Fehler gesendet und die RAPID-Abarbeitung wird angehalten.

Syntax

```
EGMSetupAO
[MecUnit ':='] <variable (VAR) of mecunit> ','
[EGMid ':='] <variable (VAR) of egmident> ','
[ExtConfigName ':='] <expression (IN) of string>
[['\Joint'] | ['\Pose'] | ['\PathCorr']]
['\APTR']
['\aoR1x ':='] <variable (VAR) of signalao>
['\aoR2y ':='] <variable (VAR) of signalao>
['\aoR3z ':='] <variable (VAR) of signalao>
['\aoR4rx ':='] <variable (VAR) of signalao>
['\aoR5ry ':='] <variable (VAR) of signalao>
['\aoR6rz ':='] <variable (VAR) of signalao>
['\aoE1 ':='] <variable (VAR) of signalao>
['\aoE2 ':='] <variable (VAR) of signalao>
['\aoE3 ':='] <variable (VAR) of signalao>
['\aoE4 ':='] <variable (VAR) of signalao>
['\aoE5 ':='] <variable (VAR) of signalao>
['\aoE6 ':='] <variable (VAR) of signalao> ';'
;
```

Weitere Informationen

Information über	Siehe
Systemparameter	Systemparameter auf Seite 43

5 RAPID Referenzinformation

5.1.12 EGMSetupGI - Einrichtung von Gruppeneingangssignalen für EGM

Externally Guided Motion

5.1.12 EGMSetupGI - Einrichtung von Gruppeneingangssignalen für EGM

Verwendung

EGMSetupGI wird verwendet, um Gruppeneingangssignale für einen spezifischen EGM Prozess einzurichten (EGMid), die als Quelle für die Positionszielwerte dienen, nach denen sich der Roboter und bis zu 6 zusätzliche Achsen richten.

EGM-Achsenmodus ist der einzige EGM-Modus, der 7-achsige Roboter unterstützt. Bei 7-achsigen Robotern bietet der Eingang der ersten zusätzlichen Achse die Position für die weiteren Roboterachsen.

Grundlegende Beispiele

```
VAR egmident egmID1;

EGMGetId egmID1;
EGMSetupGI ROB_1, egmID1, "default" \Pose \giR1x:=gi_01
\giR2y:=gi_02 \giR3z:=gi_03 \giR4rx:=gi_04 \giR5ry:=gi_05
\giR6rz:=gi_06;
```

Argumente

```
EGMSetupGI MecUnit, EGMid, ExtConfigName [\Joint] | [\Pose] |
[\PathCorr] [\APTR] [\giR1x] [\giR2y] [\giR3z] [\giR4rx]
[\giR5ry] [\giR6rz] [\giE1] [\giE2] [\giE3] [\giE4] [\giE5]
[\giE6]
```

MecUnit

Datentyp: mecunit

Name der mechanischen Einheit.

EGMid

Datentyp: egmident

EGM Identität.

ExtConfigName

Datentyp: string

Der Name der "External Motion Interface Data" (Externe Bewegungsschnittstellen-Daten) entsprechend der im System definierten Parameter. Für weitere Informationen siehe *Technisches Referenzhandbuch - Systemparameter, Typ External Motion Interface Data*, Thema *Motion*.

[\Joint]

Datentyp: switch

Wählt Achsenbewegung.

Mindestens einer der Schalter \Joint oder \Pose muss vorhanden sein.

[\Pose]

Datentyp: switch

Wählt Positionsbewegung.

Fortsetzung auf nächster Seite

5.1.12 EGMSetupGI - Einrichtung von Gruppeneingangssignalen für EGM Externally Guided Motion Fortsetzung

Mindestens einer der Schalter `\Joint` oder `\Pose` muss vorhanden sein.

`[\PathCorr]`

Datentyp: `switch`

Wählt die Bahnkorrektur.

Mindestens einer der Schalter `\Joint`, `\Pose` oder `\PathCorr` muss vorhanden sein.

`[\APTR]`

Datentyp: `switch`

Einrichten eines Sensors des Typs At-Point-Tracker zur Bahnkorrektur. Zum Beispiel `WeldGuide` oder `AWC`.

`\APTR` muss vorhanden sein, wenn `\PathCorr` verwendet wird.

`[\giR1x] [\giR2y] [\giR3z]`

Datentyp: `signalgi`

Legt das Signal fest, das die Werte x, y und z in Millimetern für die Positionsbewegung bereitstellt.

Legt das Signal fest, das die Winkel der Roboter Verbindung 1 bis 3 in Grad für die Verbindungsbewegung bereitstellt.

`[\giR4rx] [\giR5ry] [\giR6rz]`

Datentyp: `signalgi`

Legt das Signal fest, das die Werte für die Drehung x, y und z des Roboters in Grad für die Positionsbewegung bereitstellt.

Legt das Signal fest, das die Winkel der Roboter Verbindung 4 bis 6 in Grad für die Verbindungsbewegung bereitstellt.

`[\giE1] [\giE2] [\giE3] [\giE4] [\giE5] [\giE6]`

Datentyp: `signalgi`

Legt das Signal fest, das die Position der zusätzlichen Achsen 1 bis 6 bereitstellt.

Bei Verwendung des EGM-Achsenmodus bei einem 7-achsigen Roboter liefert `\giE1` die Position für die zusätzlichen Roboterachsen.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_NO_ALIASIO_DEF</code>	Die Signalvariable ist eine in RAPID angegebene Variable. Sie wurde nicht an ein E/A-Signal, das in der E/A-Konfiguration mit Instruktion <code>AliasIO</code> definiert wurde.
<code>ERR_NORUNUNIT</code>	Es ist kein Kontakt mit dem E/A-Gerät vorhanden.
<code>ERR_SIG_NOT_VALID</code>	Auf das E/A-Signal kann nicht zugegriffen werden (gilt nur für ICI-Feldbus).

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.12 EGMSetupGI - Einrichtung von Gruppeneingangssignalen für EGM

Externally Guided Motion

Fortsetzung

Einschränkungen

- EGMSetupGI kann nur in RAPID-Bewegungstasks verwendet werden.
- Die mechanische Einheit muss ein TCP-Roboter sein.
- Gruppensignale können nur positive Werte verarbeiten. Daher ist ihre Verwendung in EGM eingeschränkt.
- Mindestens ein Signal muss angegeben werden, sonst wird ein Fehler gesendet und die RAPID-Abarbeitung wird angehalten.

Syntax

```
EGMSetupGI
[MecUnit ':='] <variable (VAR) of mecunit> ',',
[EGMid ':='] <variable (VAR) of egmident> ',',
[ExtConfigName ':='] <expression (IN) of string>
[['\Joint'] | ['\Pose'] | ['\PathCorr']]
['\APTR']
['\giR1x ':='] <variable (VAR) of signalgi>
['\giR2y ':='] <variable (VAR) of signalgi>
['\giR3z ':='] <variable (VAR) of signalgi>
['\giR4rx ':='] <variable (VAR) of signalgi>
['\giR5ry ':='] <variable (VAR) of signalgi>
['\giR6rz ':='] <variable (VAR) of signalgi>
['\giE1 ':='] <variable (VAR) of signalgi>
['\giE2 ':='] <variable (VAR) of signalgi>
['\giE3 ':='] <variable (VAR) of signalgi>
['\giE4 ':='] <variable (VAR) of signalgi>
['\giE5 ':='] <variable (VAR) of signalgi>
['\giE6 ':='] <variable (VAR) of signalgi> ';'

```

Weitere Informationen

Information über	Siehe
Systemparameter	Systemparameter auf Seite 43

5.1.13 EGMSetupLTAPP - Einrichtung des LTAPP-Protokolls für EGM

Verwendung

EGMSetupLTAPP wird verwendet, um ein *LTAPP*-Protokoll für einen spezifischen EGM-Prozess (EGMid) als Bahnkorrekturen einzurichten.

Grundlegende Beispiele

Das folgende Beispiel veranschaulicht die Instruktion EGMSetupLTAPP.

Beispiel 1:

```
VAR egmident EGMid1;
EGMGetId EGMid1;
EGMSetupLTAPP ROB_1, EGMid1, "pathCorr", "OptSim", 1\LATR;
```

Dieses Programm registriert einen EGM Prozess und richtet einen Sensor *OptSim* als Datenquelle (Sensor) ein, der das Kommunikationsprotokoll *LTAPP* verwendet und den Typ *Look Ahead* hat. Der Sensor soll die Achsentypdefinition Nummer 1 für die Verfolgung verwenden.

Argumente

```
EGMActMove MecUnit, EGMid, ExtConfigName, Device, JointType [\APTR]
| [\LATR]
```

MecUnit

Datentyp: mecunit
Name der mechanischen Einheit.

EGMid

Datentyp: egmident
EGM Identität.

ExtConfigName

Datentyp: string
Der Name der "External Motion Interface Data" (Externe Bewegungsschnittstellen-Daten) entsprechend der im System definierten Parameter. Weitere Informationen finden Sie unter *Technisches Referenzhandbuch - Systemparameter, Thema Motion, Typ External Motion Interface Data*.

Device

Datentyp: string
LTAPP Name des Geräts.

JointType

Datentyp: num
Definiert den Achsentyp, ausgedrückt als Nummer, den die Sensorausrüstung während der Bahnkorrektur verwenden soll.

[\APTR]

Datentyp: switch

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.13 EGMSetupLTAPP - Einrichtung des LTAPP-Protokolls für EGM

Externally Guided Motion

Fortsetzung

Einrichten eines Sensors des Typs At-Point-Tracker zur Bahnkorrektur. Zum Beispiel WeldGuide oder AWC.

Es muss entweder \APTR oder \LATR vorhanden sein.

[\LATR]

Datentyp: switch

Einrichten eines Sensors des Typs Look-Ahead-Tracker zur Bahnkorrektur. Zum Beispiel Laser Tracker.

Es muss entweder \APTR oder \LATR vorhanden sein.

Programmabarbeitung

EGMSetupLTAPP verbindet die Kenndaten des verwendeten Sensors mit einer EGM Identität. Diese EGM-Identität kann dann in verschiedenen EGMActMove- und EGMMove-Instruktionen verwendet werden.

Syntax

```
EGMSetupLTAPP
[MecUnit ':='] <variable (VAR) of mecunit> ','
[EGMid ':='] <variable (VAR) of egmident> ','
[ExtConfigName ':='] <expression (IN) of string> ','
[Device ':='] <expression (IN) of string> ','
[JointType ':='] <expression (IN) of num>
[['\APTR] | ['\LATR]] ';' ;
```

5.1.14 EGMSetupUC - Einrichtung des UdpUc Protokolls für EGM

Verwendung

EGMSetupUC wird verwendet, um ein UdpUc Protokoll für einen spezifischen EGM Prozess einzurichten (EGMId), das als Quelle für die Positionszielwerte dient, nach dem sich der Roboter und bis zu 6 zusätzliche Achsen richten.

EGM-Achsenmodus ist der einzige EGM-Modus, der 7-achsige Roboter unterstützt. Bei 7-achsigen Robotern bietet der Eingang der ersten zusätzlichen Achse die Position für die weiteren Roboterachsen.

Grundlegende Beispiele

```
VAR egmident egmID1;
VAR string egmSensor:="egmSensor:";
EGMGetId egmID1;
EGMSetupUC ROB_1, egmID1, "default", egmSensor\Pose;
```

Argumente

```
EGMSetupUC MecUnit, EGMId, ExtConfigName, UCDevice [\Joint] |
[\Pose] | [\PathCorr] [\APTR] | [\LATR] [\CommTimeout]
```

MecUnit

Datentyp: mecunit
Name der mechanischen Einheit.

EGMId

Datentyp: egmident
EGM Identität.

ExtConfigName

Datentyp: string
Der Name der "External Motion Interface Data" (Externe Bewegungsschnittstellen-Daten) entsprechend der im System definierten Parameter. Für weitere Informationen siehe *Technisches Referenzhandbuch - Systemparameter, Typ External Motion Interface Data, Thema Motion*.

UCDevice

Datentyp: string
UdpUc Name des Geräts.

[\Joint]

Datentyp: switch
Wählt Achsenbewegung zur Positionsführung. Mindestens einer der Schalter \Joint, \Pose oder \PathCorr muss vorhanden sein.

[\Pose]

Datentyp: switch

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.14 EGMSetupUC - Einrichtung des UdpUc Protokolls für EGM

Externally Guided Motion

Fortsetzung

Wählt Positionsbewegung zur Positionsführung.

Mindestens einer der Schalter `\Joint`, `\Pose` oder `\PathCorr` muss vorhanden sein.

`[\PathCorr]`

Datentyp: `switch`

Wählt die Bahnkorrektur.

Mindestens einer der Schalter `\Joint`, `\Pose` oder `\PathCorr` muss vorhanden sein.

`[\APTR]`

Datentyp: `switch`

Einrichten eines Sensors des Typs At-Point-Tracker zur Bahnkorrektur. Zum Beispiel WeldGuide oder AWC.

Es muss entweder `\APTR` oder `\LATR` vorhanden sein.

`[\LATR]`

Datentyp: `switch`

Einrichten eines Sensors des Typs Look-Ahead-Tracker zur Bahnkorrektur. Zum Beispiel Laser Tracker.

Es muss entweder `\APTR` oder `\LATR` vorhanden sein.

`[\CommTimeout]`

Datentyp: `num`

Timeout für die Kommunikation mit dem externen UdpUC-Gerät in Sekunden.

Fehlerbehandlung

Die folgenden behebbaren Fehler werden erzeugt und können in einer Fehlerbehandlungsroutine behoben werden. Die Systemvariable `ERRNO` wird gesetzt auf:

Name	Fehlerursache
<code>ERR_UDPUC_COMM</code>	Bei der Kommunikation mit dem UdpUc-Gerät ist ein Fehler aufgetreten. Im Synchron-Modus ist <code>ERR_UDPUC_COMM</code> immer ein behebbarer Fehler, der mit einer Fehlerbehandlung behoben werden kann. Im asynchronen Modus (<code>EGMRunX\NoWait</code>) wird <code>ERR_UDPUC_COMM</code> immer als schwerwiegender Fehler gemeldet, wenn EGM ausgeführt wird.

Einschränkungen

- EGMSetupUC kann nur in RAPID-Bewegungstasks verwendet werden.
- Die mechanische Einheit muss ein TCP-Roboter sein.

Syntax

```
EGMSetupUC
    [MecUnit ':='] <variable (VAR) of mecunit> ', '
    [EGMid ':='] <variable (VAR) of egmident> ', '
```

Fortsetzung auf nächster Seite

5.1.14 EGMSetupUC - Einrichtung des UdpUc Protokolls für EGM
Externally Guided Motion
Fortsetzung

```
[ExtConfigName ':='] <expression (IN) of string> ','  
[UCDevice ':='] <expression (IN) of string>  
[['\Joint'] | ['\Pose'] | ['\PathCorr']]  
[['\APTR'] | ['\LATR']]  
['\CommTimeout ':=' <expression (IN) of num>] ';'`
```

Weitere Informationen

Information über	Siehe
Systemparameter	Systemparameter auf Seite 43

5 RAPID Referenzinformation

5.1.15 EGMStop - Anhalten einer EGM Bewegung *Externally Guided Motion*

5.1.15 EGMStop - Anhalten einer EGM Bewegung

Verwendung

EGMStop hält einen spezifischen EGM Prozess an (EGMId).

Grundlegende Beispiele

Im Bewegungstask RAPID:

```
VAR egmident egmID1;
PERS pose pose1:=[[0,0,0], [1,0,0,0]];
CONST egm_minmax egm_minmax_lin:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot:=[-0.1,0.2];
CONST pose posecor:=[[1200,400,900], [0,0,1,0]];
CONST pose posesens:=[[12.3313,-0.108707,416.142],
[0.903899,-0.00320735,0.427666,0.00765917]];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1 \Pose \aiR1x:=ai_01 \aiR2y:=ai_02
\aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05 \aiR6rz:=ai_06;
EGMActPose egmID1 \Tool:=tool0 \Wobj:=wobj0, posecor,
EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin
\y:=egm_minmax_lin \z:=egm_minmax_lin \rx:=egm_minmax_rot
\ry:=egm_minmax_rot \rz:=egm_minmax_rot \LpFilter:=20;
EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \rx \ry \rz
\RampInTime:=0.05;
```

In einer TRAP Routine:

```
EGMStop egmID1, EGM_STOP_HOLD;
```

Argumente

EGMStop EGMId, Mode [\RampOutTime]

EGMId

Datentyp: egmident

EGM Identität.

Mode

Datentyp: egmstopmode

Definiert, wie die Bewegung beendet wird (EGM_STOP_HOLD,
EGM_STOP_RAMP_DOWN)

[\RampOutTime]

Datentyp: num

Definiert in Sekunden, wie schnell ein Heruntersetzen von EGM ausgeführt wird.

**Dieser Parameter ist ohne Bedeutung, wenn der Parameter Mode auf
EGM_STOP_HOLD gestellt wurde.**

Einschränkungen

- EGMStop kann nur in RAPID-Bewegungstasks verwendet werden.

Fortsetzung auf nächster Seite

Syntax

```
EGMStop  
[EGMid ':='] <variable (VAR) of egmident>','  
[Mode ':='] <expression (IN) of egmstopmode>  
['\RampOutTime ':=' <expression (IN) of num>] ';' ;
```

5 RAPID Referenzinformation

5.1.16 EGMStreamStart - Das EGM-Positions-Streaming starten.

Externally Guided Motion

5.1.16 EGMStreamStart - Das EGM-Positions-Streaming starten.

Verwendung

EGMStreamStart startet das Streamen von Positionsdaten für einen bestimmten EGM-Prozess (EGMid).

Grundlegendes Beispiel

Das folgende Beispiel veranschaulicht die Instruktion EGMStreamStart.

Beispiel 1:

```
VAR egmident egmID1;

EGMGetId egmID1;
EGMSetupUC ROB_1, egmID1, "default", "UCdevice"\Joint;
EGMStreamStart egmID;
MoveAbsJ jpos20, v100, z20, Weldgun;
MoveAbsJ jpos10\NoEOffs, v1000, fine, Weldgun;
EGMStreamStop egmID1;
EGMReset egmID1;
```

Argumente

EGMStreamStart EGMid [\SampleRate];

EGMid

Datentyp: egmident

EGM Identität.

[\SampleRate]

Datentyp: num

Eingangsdaten Abtastlesegeschwindigkeit als Vielfaches von 4 Millisekunden.

Gültige Werte sind 4, 8, 12, 16 usw.

Der Standardwert ist 4 Millisekunden.

Programmabarbeitung

EGMStreamStart startet das Streaming von Positionsdaten an externe Geräte. Die Daten werden mit der mit \SampleRate definierten Zyklizität gesendet und der Inhalt entspricht der Protokollspezifikation in der Google Protobuf-Definitionsdatei *egm.proto*.

Einschränkungen

EGMStreamStart ist nur verfügbar, wenn EGM unter Verwendung von EGMSetupUC eingerichtet wird, d. h. das Protokoll UdpUc wird für die Kommunikation mit dem externen Gerät verwendet.

Syntax

```
EGMStreamStart
[EGMid ']:='] <variable (VAR) of egmident>;'
```

Fortsetzung auf nächster Seite

5.1.16 EGMStreamStart - Das EGM-Positions-Streaming starten.

Externally Guided Motion

Fortsetzung

```
[ '\ ' SampleRate ' := ' <expression (IN) of num> ] ', '
```

5 RAPID Referenzinformation

5.1.17 EGMStreamStop - Das EGM-Positions-Streaming stoppen. *Externally Guided Motion*

5.1.17 EGMStreamStop - Das EGM-Positions-Streaming stoppen.

Verwendung

EGMStreamStop Stoppt das Streaming von Positionsdaten für einen bestimmten EGM-Prozess (EGMid).

Grundlegendes Beispiel

Das folgende Beispiel veranschaulicht die Instruktion EGMStreamStop.

Beispiel 1:

```
VAR egmident egmID1;

EGMGetId egmID1;
EGMSetupUC ROB_1, egmID1, "default", "UCdevice"\Joint;
EGMStreamStart egmID;
MoveAbsJ jpos20, v100, z20, Weldgun;
MoveAbsJ jpos10\NoEOffs, v1000, fine, Weldgun;
EGMStreamStop egmID1;
EGMReset egmID1;
```

Argumente

EGMStreamStop EGMid;

EGMid

Datentyp: egmident
EGM Identität.

Programmabarbeitung

EGMStreamStop stoppt das Streaming von Positionsdaten an externe Geräte.

Einschränkungen

EGMStreamStop ist nur verfügbar, wenn EGM unter Verwendung von EGMSetupUC eingerichtet wird, d. h. das Protokoll UdpUc wird für die Kommunikation mit dem externen Gerät verwendet.

Syntax

```
EGMStreamStop  
[EGMid ']:='] <variable (VAR) of egmident>;'
```

5.1.18 EGMWaitCond - Auf den EGM-Prozess warten.

Verwendung

EGMWaitCond wird verwendet, um auf einen bestimmten EGM-Prozess (EGMid) zu warten.

Grundlegendes Beispiel

Das folgende Beispiel veranschaulicht die Instruktion EGMWaitCond.

Beispiel 1:

```
VAR egmident egmID1;
PERS pose pose1:=[[0,0,0],[1,0,0,0]];
CONST egm_minmax egm_minmax_lin:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot:=[-0.1,0.2];
CONST pose posecor:=[[1200,400,900],[0,0,1,0]];
CONST pose posesens:=[[12.3313,-0.108707,416.142],
                      [0.903899,-0.00320735,0.427666,0.00765917]];

EGMGetId egmID1;
EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_01
\aiR2y:=ai_02 \aiR3z:=ai_03 \aiR4rx:=ai_04 \aiR5ry:=ai_05
\aiR6rz:=ai_06;
EGMActPose egmID1 \Tool:=tool0 \WObj:=wobj0, posecor,
EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin
\y:=egm_minmax_lin \z:=egm_minmax_lin \rx:=egm_minmax_rot
\ry:=egm_minmax_rot \rz:=egm_minmax_rot \LpFilter:=20;
EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \rx \ry
\rz\RampInTime:=0.05;
SetDO doSignal1, 1;
...
EGMWaitCond
```

Argumente

EGMWaitCond EGMId;

EGMid

Datentyp: egmident
EGM Identität.

Programmabarbeitung

EGMWaitCond wartet auf den Abschluss einer EGMRunJoint/Pose-Instruktion. Wenn die Bewegung abgeschlossen ist, bevor EGMWaitCond ausgeführt wird, wird die Programmausführung mit der nächsten RAPID-Instruktion unmittelbar fortgesetzt.

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.1.18 EGMWaitCond - Auf den EGM-Prozess warten.

Externally Guided Motion

Fortsetzung

Einschränkungen

Wenn `EGMRunJoint` oder `EGMRunPose` mit dem optionalen Argument `\NoWaitCond` verwendet wird, darf keine Bewegungsinstruktion verwendet werden, bevor die EGM Position Guidance unter Verwendung von `EGMWaitCond` abgeschlossen ist.

Syntax

```
EGMWaitCond  
[EGMid ':='] <variable (VAR) of egmident>;'
```

5.2 Funktionen

5.2.1 EGMGetState - Erhält den gleichen EGM Zustand

Verwendung

EGMGetState ruft den Zustand eines EGM-Prozesses (EGMId) ab. Er wird auch verwendet, um den Korrekturstatus zu ermitteln.

Grundlegende Beispiele

```
VAR egmident egmID1;
VAR egmstate egmState1:= EGM_STATE_DISCONNECTED;
VAR egmcorrstate egmCorrState1:= EGM_CORR_UNDEFINED;

EGMGetId egmID1;
egmState1 := EGMGetState(egmID1\CorrectionState:=egmCorrState1);
```

Rückgabewert

Datentyp: egmstate

Der aktuelle Zustand des EGM Prozesses, der durch die im Argument angegebene EGM Identität identifiziert wird.

Argumente

EGMGetState (EGMId [\CorrectionState])

EGMId

Datentyp: egmident

EGM Identität.

[\CorrectionState]

Datentyp: egmcorrstate

Der aktuelle Korrekturzustand des EGM-Prozesses, der durch die im Argument angegebene EGM Identität identifiziert wird.

Einschränkungen

- **EGMGetState** kann nur in RAPID-Bewegungstasks verwendet werden.
- Die mechanische Einheit muss ein TCP-Roboter sein.

Syntax

```
EGMGetState '('
  [EGMId ':='] < variable (VAR) of egmident >
  ['\'CorrectionState ':='] <expression (VAR) of egmcorrstate>] ')'
```

Weitere Informationen

Information über	Siehe
egmstate	egmstate - Definiert den Zustand für EGM
egmcorrstate	egmcorrstate - Definiert den Korrekturzustand für EGM

5 RAPID Referenzinformation

5.3.1 egmframetype - Definiert Rahmentypen für EGM

Externally Guided Motion

5.3 Datentypen

5.3.1 egmframetype - Definiert Rahmentypen für EGM

Verwendung

egmframetype wird verwendet, um Koordinatensystemtypen für Korrekturen und Sensormessungen in EGM zu definieren.

Beschreibung

egmframetype ist zur Verwendung in den Instruktionen `EGMActJ` und `EGMActPose` vorgesehen.

Grundlegende Beispiele

```
CONST egm_minmax egm_minmax_lin1:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot1:=[-0.1,0.2];

EGMActPose egmID1\Tool:=tFroniusCMT\WObj:=wobj0, posecor,
EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin
\y:=egm_minmax_lin \z:=egm_minmax_lin \rx:=egm_minmax_rot
\ry:=egm_minmax_rot \rz:=egm_minmax_rot \LpFilter:=20;
```

Vorgegebene Werte

Wert	Beschreibung
EGM_FRAME_BASE	Das Koordinatensystem wird relativ zum Basis-Koordinatensystem (Positionsmodus) definiert.
EGM_FRAME_TOOL	Das Koordinatensystem wird relativ zum verwendeten Werkzeug (Positionsmodus) definiert.
EGM_FRAME_WOBJ	Das Koordinatensystem wird relativ zum verwendeten Werkobjekt (Positionsmodus) definiert.
EGM_FRAME_WORLD	Das Koordinatensystem wird relativ zum Weltkoordinatensystem (Positionsmodus) definiert.
EGM_FRAME_JOINT	Die Werte sind Achsenwerte (Achsenmodus).

Eigenschaften

egmframetype ist ein Alias-Datentyp für `num`.

Weitere Informationen

Information über	Siehe
EGMActJ	EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition auf Seite 51
EGMActPose	EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel auf Seite 58

5.3.2 egmident - Identifiziert einen spezifischen EGM Prozess *Externally Guided Motion*

5.3.2 egmident - Identifiziert einen spezifischen EGM Prozess

Verwendung

egmident identifiziert einen spezifischen EGM Prozess.

Beschreibung

Eine `egmident` wird durch die Funktion `EGMGetId` reserviert. Sie wird dann verwendet, um die Instruktionen `EGMSetupXX`, `EGMActX`, `EGMRunX` und `EGMReset` zu identifizieren und mit der gleichen EGM-Operation zu verknüpfen.

Eine `egmident` wird über ihren Namen identifiziert, d. h., ein zweiter oder dritter Aufruf von `EGMGetId` mit der gleichen `egmident` wird weder einen neuen Prozess reservieren, noch den Inhalt verändern. Nur `EGMReset` gibt eine `egmident` frei.

Grundlegende Beispiele

```
VAR egmident egmID1;
VAR egmstate egmSt1;

TASK PERS wobjdata wobj_EGM1:=[FALSE, TRUE, "", [[500,700,900],
    [1,0,0,0]], [[0,0,0], [1,0,0,0]]];
CONST pose posecor:=[[1200,400,900], [0,0,1,0]];
CONST pose posesens:=[[12.3313,-0.108707,416.142],
    [0.903899,-0.00320735,0.427666,0.00765917]];
CONST egm_minmax egm_minmax_lin1:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot1:=[-0.1,0.2];
CONST egm_minmax egm_minmax_joint1:=[-0.1,0.1];

PROC testAI()
    EGMReset egmID1;
    EGMGetId egmID1;
    mvHome;
    mvHome_EGMLinear;

    egmSt1:=EGMGetState(egmID1);
    TPWrite "EGM state 1: " \Num:=egmSt1;

    IF egmSt1<=EGM_STATE_CONNECTED THEN
        EGMSetupAI ROB_1, egmID1, "default" \Pose \aiR1x:=ai_MoveX
            \aiR2y:=ai_MoveY \aiR3z:=ai_MoveZ \aiR5ry:=ai_RotY
            \aiR6rz:=ai_RotZ;
    ENDIF

    EGMActPose egmID1 \Tool:=tFroniusCMT \WObj:=wobj0, posecor,
        EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin1
        \y:=egm_minmax_lin1 \z:=egm_minmax_lin1 \rx:=egm_minmax_rot1
        \ry:=egm_minmax_rot1 \rz:=egm_minmax_rot1 \LpFilter:=20;
    EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \rx \ry \rz
        \RampInTime:=0.05;

    egmSt1:=EGMGetState(egmID1);
    IF egmSt1=EGM_STATE_CONNECTED THEN
        TPWrite "Reset lin 1";
```

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.3.2 egmident - Identifiziert einen spezifischen EGM Prozess

Externally Guided Motion

Fortsetzung

```
        EGMReset egmID1;  
    ENDIF  
ENDPROC
```

Einschränkungen

Es gibt bis zu 4 gleichzeitige Instanzen, die für jede RAPID Aufgabe verfügbar sind.

Eigenschaften

egmident ist ein Datentyp ohne einen Wert (Non-Value). Er wird durch Aufruf von EGMGetId festgelegt.

Weitere Informationen

Information über	Siehe
EGMGetId	EGMGetId - Erhält eine EGM Identität auf Seite 64

5.3.3 egm_minmax - Konvergenzkriterien für EGM

Verwendung

egm_minmax wird für die Definition der Konvergenzkriterien für EGM bis Ende verwendet.

Beschreibung

egm_minmax ist zur Verwendung in den Instruktionen `EGMActJ` und `EGMActPose` vorgesehen.

Komponenten

Min

Datentyp: num

Minimale Abweichung

Definiert den Mindestwert der Positionsabweichung. Der Standardwert ist 0,5 Grad.

Max

Datentyp: num

Maximale Abweichung

Definiert den Maximalwert der Positionsabweichung. Der Standardwert ist 0,5 Grad.

Grundlegende Beispiele

```
CONST egm_minmax egm_minmax_lin1:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot1:=[-0.1,0.2];

EGMActPose egmID1\Tool:=tFroniusCMT\WObj:=wobj0, posecor,
    EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin1
    \y:=egm_minmax_lin1 \z:=egm_minmax_lin1 \rx:=egm_minmax_rot1
    \ry:=egm_minmax_rot1 \rz:=egm_minmax_rot1 \LpFilter:=20;
```

Eigenschaften

Egm_minmax verfügt über folgende Einheiten:

- Millimeter für x, y und z bei linearer Bewegung.
- Grad für rx, ry und rz für lineare Bewegungen und Achsbewegungen.

Struktur

```
< dataobject of egm_minmax >
  < min of num >
  < max of num >
```

Weitere Informationen

Information über	Siehe
EGMActJ	EGMActJoint - Vorbereiten einer EGM Bewegung für eine Achswinkelposition auf Seite 51
EGMActPose	EGMActPose - Vorbereiten einer EGM Bewegung für ein Positionsziel auf Seite 58

5 RAPID Referenzinformation

5.3.4 egmstate - Definiert den Zustand für EGM

Externally Guided Motion

5.3.4 egmstate - Definiert den Zustand für EGM

Verwendung

egmstate wird verwendet, um den Zustand für Korrekturen und Sensormessungen in EGM zu definieren.

Beschreibung

egmstate ist der Rückgabewert der Funktion `EGMGetState`.

Grundlegende Beispiele

```
VAR egmstate egmSt1;  
VAR egmident egmID1;  
  
EGMReset egmID1;  
EGMGetId egmID1;  
  
egmSt1:=EGMGetState(egmID1);  
TPWrite "EGM state: "\Num:=egmSt1;
```

Vorgegebene Werte

Wert	Beschreibung
EGM_STATE_DISCONNECTED	Der EGM Zustand des spezifischen Prozesses ist undefiniert. Es ist kein Einrichtungsvorgang aktiv.
EGM_STATE_CONNECTED	Der spezifizierte EGM Prozess ist nicht aktiviert. Die Einrichtung wurde vorgenommen, aber keine EGM-Bewegung ist aktiv.
EGM_STATE_RUNNING	Der spezifizierte EGM Prozess wird bereits ausgeführt. Die EGM-Bewegung ist aktiv , d. h. der Roboter wird bewegt.

Eigenschaften

egmstate ist ein Alias-Datentyp für `num`.

Weitere Informationen

Information über	Siehe
EGMGetState	EGMGetState - Erhält den gleichen EGM Zustand auf Seite 101

5.3.5 egmcorrstate - Definiert den Korrekturzustand für EGM

Verwendung

`egmcorrstate` wird verwendet, um den Zustand für Korrekturen und Sensormessungen in EGM zu definieren.

Beschreibung

`egmcorrstate` ist der Rückgabewert des optionalen Arguments `\CorrectionState` der Funktion `EGMGetState`.

Grundlegende Beispiele

```
VAR egmstate egmSt1;  
VAR egmcorrstate egmCorrSt1;  
VAR egmident egmID1;  
  
EGMReset egmID1;  
EGMGetId egmID1;  
  
egmSt1 := EGMGetState(egmID1\CorrectionState:=egmCorrSt1);  
TPWrite "EGM state: "\Num:=egmSt1;  
TPWrite "EGM correction state: "\Num:=egmCorrSt1;
```

Vorgegebene Werte

Wert	Beschreibung
EGM_CORR_UNDEFINED	Der EGM-Korrekturzustand des spezifischen Prozesses ist undefiniert. EGMGetId wurde nicht ausgeführt oder EGM wurde zurückgesetzt.
EGM_CORR_STOPPED	Dieser Korrekturzustand wird eingenommen, sobald EGMGetId ausgeführt wird, aber noch keine EGM Bewegung aktiv ist.
EGM_CORR_RUNNING	Der angegebene Prozess EGM wird abgearbeitet, d. h. der Roboter wird bewegt.
EGM_CORR_ERROR	Ein Fehler bei der Positionsführung oder Bahnkorrektur ist aufgetreten, während der Prozess EGM abgearbeitet wurde.

Eigenschaften

`egmstate` ist ein Alias-Datentyp für `num`.

Weitere Informationen

Information über	Siehe
EGMGetState	EGMGetState - Erhält den gleichen EGM Zustand auf Seite 101

5 RAPID Referenzinformation

5.3.6 egmstopmode - Definiert die Stopp-Modi für EGM

Externally Guided Motion

5.3.6 egmstopmode - Definiert die Stopp-Modi für EGM

Verwendung

egmstopmode wird verwendet, um die Stopp-Modi für Korrekturen und Sensormessungen in EGM zu definieren.

Beschreibung

egmstopmode ist zur Verwendung in den Instruktionen `EGMRunJoint`, `EGMRunPose` und `EGMStop` vorgesehen.

Grundlegende Beispiele

Vom RAPID-Bewegungstask:

```
VAR egmstate egmSt1;
VAR egmident egmID1;

EGMReset egmID1;
EGMGetId egmID1;
CONST egm_minmax egm_minmax_lin1:=[-0.1,0.1];
CONST egm_minmax egm_minmax_rot1:=[-0.1,0.2];

EGMActPose egmID1 \Tool:=tFroniusCMT \WObj:=wobj0, posecor,
    EGM_FRAME_WOBJ, posesens, EGM_FRAME_TOOL \x:=egm_minmax_lin
    \y:=egm_minmax_lin \z:=egm_minmax_lin \rx:=egm_minmax_rot
    \ry:=egm_minmax_rot \rz:=egm_minmax_rot \LpFilter:=20;
EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \rx \ry \rz
    \RampInTime:=0.05;
```

Von einem RAPID TRAP oder Hintergrundtask:

```
EGMStop egmID1, EGM_STOP_RAMP_DOWN\RampOutTime:=5.0;
```

Vorgegebene Werte

Wert	Beschreibung
EGM_STOP_HOLD	Behält die EGM Endposition bei.
EGM_STOP_RAMP_DOWN	Kehrt von der EGM Endposition zur Startposition zurück.

Eigenschaften

egmstopmode ist ein Alias-Datentyp für num.

Weitere Informationen

Information über	Siehe
EGMRunJoint	EGMRunJoint - Ausführen einer EGM Bewegung mit einer Achswinkelposition auf Seite 74
EGMRunPose	EGMRunPose - Ausführen einer EGM Bewegung mit einem Positionsziel auf Seite 77
EGMStop	EGMStop - Anhalten einer EGM Bewegung auf Seite 94

5.4 Codebeispiele

5.4.1 Verwendung des EGM-Positionsdatenstroms

Beschreibung

Das Gerät, das die Eingangsdaten für EGM enthält, muss zunächst als UdpUc-Gerät konfiguriert werden. Siehe [Konfigurieren von UdpUc-Geräten auf Seite 41](#).

Jetzt kann das Gerät von EGM verwendet werden, um Positionen von mechanischen Einheiten zu einem externen Gerät zu streamen. Einfache Beispiele finden Sie unten.

Positionen können von mehreren Bewegungstasks gestreamt werden, aber Sie müssen für jeden Bewegungstask einen Kommunikationskanal verwenden.

Beispiele

Verwendung von EGMStreamStart und EGMStreamStop für eine mechanische Einheit

Diese Methode ist der einfachste Weg zur Verwendung von EGM Position Stream, ist jedoch nicht genau für Roboter mit Absolute Accuracy oder hoher Last.

```
VAR egmident egmID1;
EGMGetId egmID1;
! Set up the EGM data source: UdpUc server using device "UCdevice"
  and configuration "default"
EGMSetupUC ROB_1, egmID1, "default", "UCdevice"\Joint;
! Start the position stream for T_ROB1 including active external
  axis. Cycle time is 16 ms.
EGMStreamStart egmID1\SampleRate:=16;
! Run your program - streaming is active
MoveAbsJ jpos20, v100, z20, Weldgun;
...
...
MoveAbsJ jpos10\NoEOffs, v1000, fine, Weldgun;
! Stop the position stream - but it is not necessary if you want
  to stream until the controller shuts down
EGMStreamStop egmID1;
EGMReset egmID1;
```

Verwendung von EGMActXX\StreamStart für eine mechanische Einheit

Diese Methode wird bevorzugt, wenn Sie einen Roboter mit Absolute Accuracy haben, da die RAPID-Instruktionen EGMActPose und EGMActJoint die Daten für Werkzeug und Last an die Steuerung weitergeben.

```
VAR egmident egmID1;
! Used tool
TASK PERS tooldata Weldgun:=[TRUE,[[12.3313,-0.108707,416.142],
  [0.903899,-0.00320735,0.427666,0.00765917]],
  [2.6,[-111.1,24.6,386.6],[1,0,0,0],0,0,0.072]]];
! limits for cartesian convergence: +-1 mm
CONST egm_minmax egm_minmax_lin1:=[-1,1];
! limits for orientation convergence: +-2 degrees
CONST egm_minmax egm_minmax_rot1:=[-2,2];
```

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.4.1 Verwendung des EGM-Positionsdatenstroms

Fortsetzung

```
! Correction frame offset: none
VAR pose corr_frame_offs=[[0,0,0],[1,0,0,0]];
EGMGetId egmID1;
! Set up the EGM data source: UdpUc server using device "UCdevice"
  and configuration "default"
EGMSetupUC ROB_1, egmID1, "default", "UCdevice"\Joint;
! Correction frame is the World coordinate system and the sensor
  measurements are relative to the tool frame of the used tool
  (Weldgun). Start the position stream for T_ROB1 including
  active external axis. Cycle time is 16 ms.
EGMActPose egmID1\StreamStart\Tool:= Weldgun, corr_frame_offs,
EGM_FRAME_WORLD, Weldgun.tframe, EGM_FRAME_TOOL
\X:=egm_minmax_lin1\Y:=egm_minmax_lin1\Z:=egm_minmax_lin1
\RX:=egm_minmax_rot1\RY:=egm_minmax_rot1\RZ:=egm_minmax_rot1
\LpFilter:=20;
! Run your program - streaming is active
MoveAbsJ jpos20, v100, z20, Weldgun;
...
...
MoveAbsJ jpos10\NoEOffs, v1000, fine, Weldgun;
! Stop the position stream - but this is not necessary if you want
  to stream until the controller shuts down
EGMStreamStop egmID1;
EGMReset egmID1;
```

Verwendung von EGMStreamStart und EGMStreamStop für mehrere mechanische Einheiten

Dieses Beispiel bezieht sich auf ein MultiMove-System mit zwei Robotern, die jeweils auf einer Verfahreinheit montiert sind.

RAPID-Task für Roboter 1:

```
VAR egmident egmID1;
! Activate the mechanical unit for the track motion
ActUnit TRACK1;
EGMReset egmID1;
EGMGetId egmID1;
! Set up the EGM streaming destination for ROB1, including active
  additional axis, using device "UCdevice1" and configuration
  "default"
EGMSetupUC ROB_1, egmID1, "default", "UCdevice1"\Joint;
EGMStreamStart egmID1;
! Start the position stream for ROB1 including active additional
  axis. Cycle time is 4 ms (default).
! Run your program - streaming is active
MoveJ p10, v1000, z50, Weldgun;
...
...
MoveAbsJ jpos10\NoEOffs, v1000, fine, Weldgun;
! Stop the position stream
EGMStreamStop egmID1;
! Deactivate the mechanical unit for the track motion
DeactUnit TRACK1;
```

Fortsetzung auf nächster Seite

RAPID-Task für Roboter 2:

```
VAR egmident egmID2;
! Activate the mechanical unit for the track motion
ActUnit TRACK2;
EGMReset egmID2;
EGMGetId egmID2;
! Set up the EGM streaming destination for ROB2, including active
  additional axis, using device "UCdevice2" and configuration
  "default"
EGMSetupUC ROB_2, egmID2, "default", "UCdevice2"\Joint;
! Start the position stream for ROB2 including active additional
  axis. Cycle time is 4 ms (default).
EGMStreamStart egmID2;
! Run your program - streaming is active
MoveJ p10, v1000, z50, PKI_500;
...
...
MoveAbsJ jpos10\NoEOffs, v1000, fine, PKI_500;
! Stop the position stream
EGMStreamStop egmID2;
! Deactivate the mechanical unit for the track motion
DeactUnit TRACK2;
```

5.4.2 Verwendung von EGM Position Guidance mit einem UdpUc-Gerät

Beschreibung

Das Gerät, das die Eingangsdaten für EGM enthält, muss zunächst als UdpUc-Gerät konfiguriert werden. Siehe [Konfigurieren von UdpUc-Geräten auf Seite 41](#).

Jetzt kann das Gerät von EGM verwendet werden, um einen Roboter zu führen. Nachfolgend finden Sie ein einfaches Beispiel:

Beispiel

```
MODULE EGM_test
  VAR egmident egmID1;
  VAR egmstate egmSt1;

  ! limits for cartesian convergence: +-1 mm
  CONST egm_minmax egm_minmax_lin1:=[-1,1];
  ! limits for orientation convergence: +-2 degrees
  CONST egm_minmax egm_minmax_rot1:=[-2,2];

  ! Start position
  CONST jointtarget
    jpos10:=[[0,0,0,0,40,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  ! Used tool
  TASK PERS tooldata
    tFroniusCMT:=[TRUE,[[12.3313,-0.108707,416.142],
    [0.903899,-0.00320735,0.427666,0.00765917]],
    [2.6,[-111.1,24.6,386.6],[1,0,0,0],0,0,0.072]];
  ! corr-frame: wobj, sens-frame: wobj
  TASK PERS wobjdata wobj_EGM1:=[FALSE,TRUE,"",
    [[150,1320,1140],[1,0,0,0]], [[0,0,0],[1,0,0,0]]];
  ! Correction frame offset: none
  VAR pose corr_frame_offs:=[[0,0,0],[1,0,0,0]];

  PROC main()
    ! Move to start position. Fine point is demanded.
    MoveAbsJ jpos10\NoEOffs, v1000, fine, tFroniusCMT;
    testuc;
  ENDPROC

  PROC testuc()
    EGMReset egmID1;
    EGMGetId egmID1;

    egmSt1:=EGMGetState(egmID1);
    TPWrite "EGM state: "\Num:=egmSt1;

    IF egmSt1 <= EGM_STATE_CONNECTED THEN
      ! Set up the EGM data source: UdpUc server using device
      "EGMsensor:" and configuration "default"
      EGMSetupUC ROB_1, egmID1, "default", "EGMsensor:"\pose;
    ENDIF
```

Fortsetzung auf nächster Seite

5.4.2 Verwendung von EGM Position Guidance mit einem UdpUc-Gerät*Fortsetzung*

```
! Correction frame is the World coordinate system and the
  sensor measurements are relative to the tool frame of
  the used tool (tFroniusCMT)
EGMActPose egmID1\Tool:=tFroniusCMT, corr_frame_offs,
  EGM_FRAME_WORLD, tFroniusCMT.tframe, EGM_FRAME_TOOL
  \x:=egm_minmax_lin1 \y:=egm_minmax_lin1
  \z:=egm_minmax_lin1 \rx:=egm_minmax_rot1
  \ry:=egm_minmax_rot1 \rz:=egm_minmax_rot1 \LpFilter:=20;
! Run: the convergence condition has to be fulfilled during
  2 seconds before RAPID execution continues to the next
  instruction
EGMRunPose egmID1, EGM_STOP_HOLD \x \y \z \CondTime:=2
  \RampInTime:=0.05;

egmSt1:=EGMGetState(egmID1);
IF egmSt1 = EGM_STATE_CONNECTED THEN
  TPWrite "Reset EGM instance egmID1";
  EGMReset egmID1;
ENDIF
ENDPROC
ENDMODULE
```

5.4.3 Verwendung von EGM Position Guidance mit Signalen als Eingang

Beschreibung

Alle Signale, die zusammen mit EGM verwendet werden, müssen in der E/A-Konfiguration des Systems definiert werden, d. h., die Signale, die mit `EGMSetupAI`, `EGMSetupAO` oder `EGMSetupGI` eingerichtet werden. Danach können die Signale von EGM verwendet werden, um einen Roboter zu führen.

Das folgende RAPID-Programmbeispiel verwendet analoge Ausgangssignale als Eingang. Der Hauptgrund für die analogen Ausgangssignale ist, dass sie leichter simuliert werden können als analoge Eingangssignale. In einer echten Anwendungsgruppe können Eingangssignale und analoge Eingangssignale üblicher sein.

Der Einfachheit halber wurden die analogen Ausgangssignale in den untenstehenden Beispielen vor der `EGMRun`-Instruktion auch auf einen konstanten Wert eingestellt. Normalerweise aktualisiert ein externes Gerät die Signalwerte, um die gewünschten Roboterpositionen anzugeben.

Im zweiten untenstehenden Beispiel wird dargestellt, wie ein 7-achsiger Roboter im EGM-Achsenmodus verwendet werden kann.

Beispiel 1:

```
MODULE EGM_test
VAR egmident egmID1;
VAR egmident egmID2;

CONST egm_minmax egm_minmax_lin1:=[-1,1];
CONST egm_minmax egm_minmax_rot1:=[-2,2];
CONST egm_minmax egm_minmax_joint1:=[-0.1,0.1];

CONST robtarget p20:=[[150,1320,1140],
[0.000494947,0.662278,-0.749217,-0.00783173], [0,0,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p30:=[[114.50,1005.42,1410.38],
[0.322151,-0.601023,0.672381,0.287914], [0,0,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST jointtarget
jpos10:=[[0,0,0,0,35,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST pose posecor:=[[1200,400,900],[1,0,0,0]];
CONST pose posesens:=[[12.3313,-0.108707,416.142],
[0.903899,-0.00320735,0.427666,0.00765917]];

! corr-frame: world, sens-frame: world
VAR pose posecor0:=[[0,0,0],[1,0,0,0]];
VAR pose posesen0:=[[0,0,0],[1,0,0,0]];

TASK PERS tooldata tFroniusCMT:=[TRUE,[[12.3313,-0.108707,416.142],
[0.903899,-0.00320735,0.427666,0.00765917]],
[2.6,[-111.1,24.6,386.6],[1,0,0,0],0,0,0.072]];
TASK PERS loaddata load1:=[5,[0,1,0],[1,0,0,0],0,0,0];
```

Fortsetzung auf nächster Seite

5.4.3 Verwendung von EGM Position Guidance mit Signalen als Eingang

Fortsetzung

```

! corr-frame: wobj, sens-frame: wobj
TASK PERS wobjdata
    wobj_EGM1:=[FALSE,TRUE,"",[150,1320,1140],[1,0,0,0]],
    [[0,0,0],[1,0,0,0]];
VAR pose posecor1:=[[0,0,0],[1,0,0,0]];
VAR pose posesen1:=[[0,0,0],[1,0,0,0]];
TASK PERS wobjdata
    wobj_EGM2:=[FALSE,TRUE,"",[0,1000,1000],[1,0,0,0]],
    [[0,0,0],[1,0,0,0]];
VAR pose posecor2:=[[150,320,0],[1,0,0,0]];
VAR pose posesen2:=[[150,320,0],[1,0,0,0]];

PROC main()
MoveAbsJ jpos10\NoEOffs, v1000, fine, tFroniusCMT;
testAO;
ENDPROC

PROC testAO()
! Get two different EGM identities. They will be used for two
different eGM setups.
EGMGetId egmID1;
EGMGetId egmID2;

! Set up the EGM data source: Analog output signals and
configuration "default"
! One guidance using Pose mode and one using Joint mode
EGMSetupAO ROB_1, egmID1, "default" \Pose \aoR1x:=ao_MoveX
\aoR2y:=ao_MoveY \aoR3z:=ao_MoveZ \aoR5ry:=ao_RotY
\aoR6rz:=ao_RotZ;
EGMSetupAO ROB_1, egmID2, "default" \Joint \aoR1x:=ao_MoveX
\aoR2y:=ao_MoveY \aoR3z:=ao_MoveZ \aoR4rx:=ao_RotX
\aoR5ry:=ao_RotY \aoR6rz:=ao_RotZ;

! Move to the starting point - fine point is needed.
MoveJ p30, v1000, fine, tool0;
! Set the signals
SetAO ao_MoveX, 150;
SetAO ao_MoveY, 1320;
SetAO ao_MoveZ, 900;
! Correction frame is the World coordinate system and the sensor
measurements are also relative to the world frame
! No offset is defined (posecor0 and posesen0)
EGMActPose egmID1 \Tool:=tFroniusCMT \WObj:=wobj0 \TLoad:=load1,
posecor0, EGM_FRAME_WORLD, posesen0, EGM_FRAME_WORLD
\X:=egm_minmax_lin1 \Y:=egm_minmax_lin1 \Z:=egm_minmax_lin1
\Rx:=egm_minmax_rot1 \Ry:=egm_minmax_rot1 \Rz:=egm_minmax_rot1
\LpFilter:=20 \SampleRate:=16 \MaxPosDeviation:=1000;
! Run: keep the end position without returning to the start position
EGMRunPose egmID1,
    EGM_STOP_HOLD\X\Y\Z\RampInTime:=0.05\PosCorrGain:=1;

! Move to the starting point - fine point is needed.

```

Fortsetzung auf nächster Seite

5.4.3 Verwendung von EGM Position Guidance mit Signalen als Eingang

Fortsetzung

```
MoveJ p20, v1000, fine, tFroniusCMT;
! Set the signals
SetAO ao_MoveX, 150;
SetAO ao_MoveY, 1320;
SetAO ao_MoveZ, 1100;
! Run with the same frame definitions: ramp down to the start
  position after having reached the EGM end position
EGMRunPose egmID1,
  EGM_STOP_RAMP_DOWN\X\Y\Z\RampInTime:=0.05\PosCorrGain:=1;

! Move to the starting point - fine point is needed.
MoveJ p30, v1000, fine, tool0;
! Set the signals
SetAO ao_MoveX, 50;
SetAO ao_MoveY, -20;
SetAO ao_MoveZ, -20;
! Correction frame is the Work object wobj_EGM1 and the sensor
  measurements are also relative to the same work object. No
  offset is defined (posecor1 and posesen1)
EGMActPose egmID1 \Tool:=tFroniusCMT \Wobj:=wobj_EGM1 \TLoad:=load1,
  posecor1, EGM_FRAME_WOBJ, posesen1, EGM_FRAME_WOBJ
  \X:=egm_minmax_lin1 \Y:=egm_minmax_lin1 \Z:=egm_minmax_lin1
  \Rx:=egm_minmax_rot1 \Ry:=egm_minmax_rot1 \Rz:=egm_minmax_rot1
  \LpFilter:=20;
! Run: keep the end position without returning to the start position
EGMRunPose egmID1,
  EGM_STOP_HOLD\X\Y\Z\RampInTime:=0.05\PosCorrGain:=1;

! Move to the starting point - fine point is needed.
MoveJ p20, v1000, fine, tFroniusCMT;
! Set the signals
SetAO ao_MoveX, 0;
SetAO ao_MoveY, 0;
SetAO ao_MoveZ, 0;
! Correction frame is the Work object wobj_EGM2 and the sensor
  measurements are also relative to the same work object. This
  time an offset is defined for the correction frame (posecor2),
  and for the sensor frame (posesen2)
EGMActPose egmID1 \Tool:=tFroniusCMT \Wobj:=wobj_EGM2 \TLoad:=load1,
  posecor2, EGM_FRAME_WOBJ, posesen2, EGM_FRAME_WOBJ
  \X:=egm_minmax_lin1 \Y:=egm_minmax_lin1 \Z:=egm_minmax_lin1
  \Rx:=egm_minmax_rot1 \Ry:=egm_minmax_rot1 \Rz:=egm_minmax_rot1
  \LpFilter:=20;
! Run: keep the end position without returning to the start position
EGMRunPose egmID1,
  EGM_STOP_HOLD\X\Y\Z\RampInTime:=0.05\PosCorrGain:=1;

! Move to the starting point - fine point is needed.
MoveJ p20, v1000, fine, tFroniusCMT;
! Set the signals
SetAO ao_MoveX, 0;
SetAO ao_MoveY, 0;
```

Fortsetzung auf nächster Seite

5.4.3 Verwendung von EGM Position Guidance mit Signalen als Eingang

Fortsetzung

```

SetAO ao_MoveZ, 0;
! Correction frame is of tool type and the sensor measurements are
  relative to the work object wobj_EGM2. This time an offset
  is defined for the correction frame (posecor2), and for the
  sensor frame (posesen2)
EGMActPose egmID1 \Tool:=tFroniusCMT \WObj:=wobj_EGM2, posecor2,
  EGM_FRAME_TOOL, posesen2, EGM_FRAME_WOBJ \x:=egm_minmax_lin1
  \y:=egm_minmax_lin1 \z:=egm_minmax_lin1 \rx:=egm_minmax_rot1
  \ry:=egm_minmax_rot1 \rz:=egm_minmax_rot1 \LpFilter:=20;
EGMRunPose egmID1,
  EGM_STOP_HOLD\x\y\z\RampInTime:=0.05\PosCorrGain:=1;

! Move to the starting point - fine point is needed.
MoveJ p20, v1000, fine, tFroniusCMT\TLoad:=load1;
! Set the signals
SetAO ao_MoveX, 150;
SetAO ao_MoveY, 1320;
SetAO ao_MoveZ, 1100;
! Same as last, but with tool0 and wobj0
EGMActPose egmID1, posecor2, EGM_FRAME_TOOL, posesen2,
  EGM_FRAME_WOBJ \x:=egm_minmax_lin1 \y:=egm_minmax_lin1
  \z:=egm_minmax_lin1 \rx:=egm_minmax_rot1 \ry:=egm_minmax_rot1
  \rz:=egm_minmax_rot1 \LpFilter:=20;
! Run: keep the end position without returning to the start position
EGMRunPose egmID1,
  EGM_STOP_HOLD\x\y\z\RampInTime:=0.05\PosCorrGain:=1;

! Move to the starting point - fine point is needed.
MoveJ p20, v1000, fine, tFroniusCMT\TLoad:=load1;
! Set the signals
SetAO ao_MoveX, 70;
SetAO ao_MoveY, -5;
SetAO ao_MoveZ, 0;
SetAO ao_RotX, 0;
SetAO ao_RotY, 0;
SetAO ao_RotZ, 0;
! Joint guidance for joints 2-6
EGMActJoint egmID2 \J2:=egm_minmax_joint1 \J3:=egm_minmax_joint1
  \J4:=egm_minmax_joint1 \J5:=egm_minmax_joint1
  \J6:=egm_minmax_joint1 \LpFilter:=20;
! Run: keep the end position without returning to the start position
EGMRunJoint egmID2, EGM_STOP_HOLD \J2 \J3 \J4 \J5 \J6 \CondTime:=0.1
  \RampInTime:=0.05 \PosCorrGain:=1;

EGMReset egmID1;
EGMReset egmID2;
ENDPROC
ENDMODULE

```

Fortsetzung auf nächster Seite

Beispiel 2:

```
MODULE EGM_IRB14000_test
VAR egmident egmID;
CONST egm_minmax egm_minmax_joint1:=[-0.1,0.1];
! For handling if the test is used with left or right arm.
VAR jointtarget jpos10;
CONST jointtarget jpos10_L:=[[0,-130,30,0,40,0],
    [135,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST jointtarget jpos10_R:=[[0,-130,30,0,40,0],
    [-135,9E+09,9E+09,9E+09,9E+09,9E+09]];
PROC main()
    IF GetMecUnitName(ROB_ID) = "ROB_L" THEN
        jpos10 := jpos10_L;
        testAO;
    ELSEIF GetMecUnitName(ROB_ID) = "ROB_R" THEN
        jpos10 := jpos10_R;
        testAO;
    ENDIF
ENDPROC
PROC testAO()
    ! Get an EGM idenity.
    EGMGetId egmID;
    ! Set up the EGM data source:
    ! Analog output signals and configuration "default".
    ! Only the EGM Joint mode support IRB14000.
    ! Notice the joint mapping of the analog output signals.
    EGMSetupAO ROB_ID, egmID, "default" \Joint \aoR1x:=ao_J1
        \aoR2y:=ao_J2 \aoR3z:=ao_J4 \aoR4rx:=ao_J5 \aoR5ry:=ao_J6
        \aoR6rz:=ao_J7 \AoE1:=ao_J3;
    ! Move to the starting point - fine point is needed.
    MoveAbsJ jpos10\NoEOffs, v50, fine, tool0;
    ! Set the signals (using an incrementing offset from the initial
    position).
    ! Another set of analog signals should be created, if running
    this code for both arms at the same time.
    ! Notice the joint mapping from a jointtarget to the analog output
    signals.
    SetAO ao_J1, jpos10.robax.rax_1 + 1;
    SetAO ao_J2, jpos10.robax.rax_2 + 2;
    SetAO ao_J3, jpos10.extax.eax_a + 3;
    SetAO ao_J4, jpos10.robax.rax_3 + 4;
    SetAO ao_J5, jpos10.robax.rax_4 + 5;
    SetAO ao_J6, jpos10.robax.rax_5 + 6;
    SetAO ao_J7, jpos10.robax.rax_6 + 7;
    ! Joint guidance for joints 1-7.
    EGMActJoint egmID \J1:=egm_minmax_joint1 \J2:=egm_minmax_joint1
        \J3:=egm_minmax_joint1 \J4:=egm_minmax_joint1
        \J5:=egm_minmax_joint1 \J6:=egm_minmax_joint1
        \J7:=egm_minmax_joint1 \LpFilter:=20;
    ! Run: keep the end position without returning to the start
    position.
```

5.4.3 Verwendung von EGM Position Guidance mit Signalen als Eingang *Fortsetzung*

```
EGMRunJoint egmID, EGM_STOP_HOLD \J1 \J2 \J3 \J4 \J5 \J6 \J7  
    \CondTime:=1 \RampInTime:=0.05 \PosCorrGain:=1;  
EGMReset egmID;  
ENDPROC ENDMODULE
```

5.4.4 Verwendung von EGM Path Correction mit verschiedenen Protokolltypen

Beschreibung

Dieses Beispiel enthält Beispiele für verschiedene Sensor- und Protokolltypen. Die grundlegende RAPID Programmstruktur ist die gleich für alle davon. Sie verwenden die gleiche externe Bewegungsdatenkonfiguration.

Beispiel

```
MODULE EGM_PATHCORR
! Used tool
PERS tooldata tEGM:=[TRUE,[[148.62,0.25,326.31],
    [0.833900724,0,0.551914471,0]], [1,[0,0,100],
    [1,0,0,0],0,0,0]];
! Sensor tool, has to be calibrated
PERS tooldata
    tLaser:=[TRUE,[[148.619609537,50.250017146,326.310337954],
    [0.390261856,-0.58965743,-0.58965629,0.390263064]],
    [1,[-0.920483747,-0.000000536,-0.390780849],
    [1,0,0,0],0,0,0]];
! Displacement used
VAR pose PP:=[[0,-3,2],[1,0,0,0]];
VAR egmident egmId1;

! Protocol: LTAPP
! Example for a look ahead sensor, e.g. Laser Tracker
PROC Part_2_EGM_OT_Pth_1()
    EGMGetId egmId1;
    ! Set up the EGM data source: LTAPP server using device "Optsim",
    configuration "pathCorr", joint type 1 and look ahead
    sensor.
    EGMSetupLTAPP ROB_1, egmId1, "pathCorr", "OptSim", 1\LATR;
    ! Activate EGM and define the sensor frame. Correction frame is
    always the path frame.
    EGMActMove egmId1, tLaser.tframe\SampleRate:=48;
    ! Move to a suitable approach position.
    MoveJ p100,v1000,z10,tEGM\WObj:=wobj0;
    MoveL p110,v1000,z100,tEGM\WObj:=wobj0;
    MoveL p120,v1000,z100,tEGM\WObj:=wobj0;
    ! Activate displacement (not necessary but possible)
    PDispSet PP;
    ! Move to the start point. Fine point is demanded.
    MoveL p130, v10, fine, tEGM\WObj:=wobj0;
    ! movements with path corrections.
    EGMMoveL egmId1, p140, v10, z5, tEGM\WObj:=wobj0;
    EGMMoveL egmId1, p150, v10, z5, tEGM\WObj:=wobj0;
    EGMMoveC egmId1, p160, p165, v10, z5, tEGM\WObj:=wobj0;
    ! Last path correction movement has to end with a fine point.
    EGMMoveL egmId1, p170, v10, fine, tEGM\WObj:=wobj0;
    ! Move to a safe position after path correction.
    MoveL p180,v1000,z10,tEGM\WObj:=wobj0;
    ! Release the EGM identity for reuse.
```

Fortsetzung auf nächster Seite

5.4.4 Verwendung von EGM Path Correction mit verschiedenen Protokolltypen

Fortsetzung

```

    EGMReset egmId1;
ENDPROC

! Protocol: LTAPP
! Example for an at point sensor, e.g. Weldguide
PROC Part_2_EGM_WG_Pth_1()
    EGMGetId egmId1;
    ! Set up the EGM data source: LTAPP server using device "wglsim",
      configuration "pathCorr", joint type 1 and at point sensor.
    EGMSetupLTAPP ROB_1, egmId1, "pathCorr", "wglsim", 1\APTR;
    ! Activate EGM and define the sensor frame, which is the tool
      frame for at point trackers.
    ! Correction frame is always the path frame.
    EGMActMove egmId1, tEGM.tframe\SampleRate:=48;
    ! Move to a suitable approach position.
    MoveJ p100,v1000,z10,tEGM\WObj:=wobj0;
    MoveL p110,v1000,z100,tEGM\WObj:=wobj0;
    MoveL p120,v1000,fine,tEGM\WObj:=wobj0;
    ! Activate displacement (not necessary but possible)
    PDispSet PP;
    ! Move to the start point. Fine point is demanded.
    MoveL p130, v10, fine, tEGM\WObj:=wobj0;
    ! movements with path corrections.
    EGMMoveL egmId1, p140, v10, z5, tEGM\WObj:=wobj0;
    EGMMoveL egmId1, p150, v10, z5, tEGM\WObj:=wobj0;
    EGMMoveC egmId1, p160, p165, v10, z5, tEGM\WObj:=wobj0;
    ! Last path correction movement has to end with a fine point.
    EGMMoveL egmId1, p170, v10, fine, tEGM\WObj:=wobj0;
    ! Move to a safe position after path correction.
    MoveL p180,v1000,z10,tEGM\WObj:=wobj0;
    ! Release the EGM identity for reuse.
    EGMReset egmId1;
ENDPROC

! Protocol: UdpUc
! Example for an at point sensor, e.g. Weldguide
PROC Part_2_EGM_UDPUC_Pth_1()
    EGMGetId egmId1;
    EGMSetupUC ROB_1, egmId1, "pathCorr", "UCdevice"\PathCorr\APTR;
    EGMActMove egmId1, tEGM.tframe\SampleRate:=48;
    ! Move to a suitable approach position.
    MoveJ p100,v1000,z10,tEGM\WObj:=wobj0;
    MoveL p110,v1000,z100,tEGM\WObj:=wobj0;
    MoveL p120,v1000,fine,tEGM\WObj:=wobj0;
    ! Activate displacement (not necessary but possible)
    PDispSet PP;
    ! Move to the start point. Fine point is demanded.
    MoveL p130, v10, fine, tEGM\WObj:=wobj0;
    ! movements with path corrections.
    EGMMoveL egmId1, p140, v10, z5, tEGM\WObj:=wobj0;
    EGMMoveL egmId1, p150, v10, z5, tEGM\WObj:=wobj0;

```

Fortsetzung auf nächster Seite

5 RAPID Referenzinformation

5.4.4 Verwendung von EGM Path Correction mit verschiedenen Protokolltypen

Fortsetzung

```
EGMMoveC egmId1, p160, p165, v10, z5, tEGM\WObj:=wobj0;  
! Last path correction movement has to end with a fine point.  
EGMMoveL egmId1, p170, v10, fine, tEGM\WObj:=wobj0;  
! Move to a safe position after path correction.  
MoveL p180,v1000,z10,tEGM\WObj:=wobj0;  
! Release the EGM identity for reuse.  
EGMReset egmId1;  
ENDPROC  
ENDMODULE
```

6 UdpUc Codebeispiele

Dateispeicherorte

Die folgenden Codebeispiele sind in der RobotWare-Verteilung verfügbar.

Datei	Beschreibung
<i>egm-sensor.cs</i>	Beispiel zur Verwendung von protobuf-csharp-port
<i>egm-sensor.cpp</i>	Beispiel zur Verwendung von Google protocol buffers C++
<i>egm.proto</i>	Die <i>egm.proto</i> -Datei definiert den Datenvertrag zwischen dem Roboter und dem Sensor.

Die Dateien können über den PC oder die Robotersteuerung bezogen werden.

- **Im RobotWare-Installationsordner in RobotStudio:**
`...\RobotPackages\RobotControl_<version>\utility\Template\EGM\`
- **An der OmniCore-Steuerung:**
`<SystemName>\PRODUCTS\RobotControl_x.x.x-xxx\utility\Template\EGM\`



Hinweis

Navigieren Sie zum RobotWare-Installationsordner in der Registerkarte **Add-Ins** in RobotStudio, indem Sie im **Add-In-Browser** auf die installierte RobotWare-Version rechtsklicken und **Paketordner öffnen** auswählen.

Diese Seite wurde absichtlich leer gelassen

Index

C

C# API, 35

E

EGM, 13

egm_minmax, 105

EGM.proto Datei, 123

EGM Abarbeitungsstatus, 23

EGM Abarbeitungszustände, 24

EGMActJoint, 51

EGMActMove, 56

EGMActPose, 58

egmframetype, 102

EGMGetId, 64

EGMGetState, 101

egmident, 103

EGMMoveC, 65

EGMMoveL, 69

EGM Path Correction, 13

EGM Position Guidance, 13

EGM Position Stream, 13

EGMReset, 73

EGMRunJoint, 74

EGMRunPose, 77

EGM Sensorprotokoll, 33

EGMSetupAI, 80

EGMSetupAO, 83

EGMSetupGI, 86

EGMSetupLTAPP, 89

EGMSetupUC, 91

egmstate, 106–107

EGMStop, 94

egmstopmode, 108

EGMStreamStart, 96

EGMStreamStop, 98

EGMWaitCond, 99

Externally Guided Motion, 13

External Motion Interface Data, 43

G

Google C++, 35

Google C++ API, 35

Google overview, 35

Google Protocol Buffers, 33, 35

N

Nanopb, 35

P

Protobuf, 35

Protobuf-csharp, 35

Protobuf-net, 35

S

Sicherheit, 11

U

UDP, 33

UdpUc, 25, 27

Udp Unicast Communication, 25, 27

**ABB AB****Robotics & Discrete Automation**

S-721 68 VÄSTERÅS, Sweden

Telephone +46 10-732 50 00

ABB AS**Robotics & Discrete Automation**

Nordlysvegen 7, N-4340 BRYNE, Norway

Box 265, N-4349 BRYNE, Norway

Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.

Robotics & Discrete Automation

No. 4528 Kangxin Highway

PuDong New District

SHANGHAI 201319, China

Telephone: +86 21 6105 6666

ABB Inc.**Robotics & Discrete Automation**

1250 Brown Road

Auburn Hills, MI 48326

USA

Telephone: +1 248 391 9000

abb.com/robotics