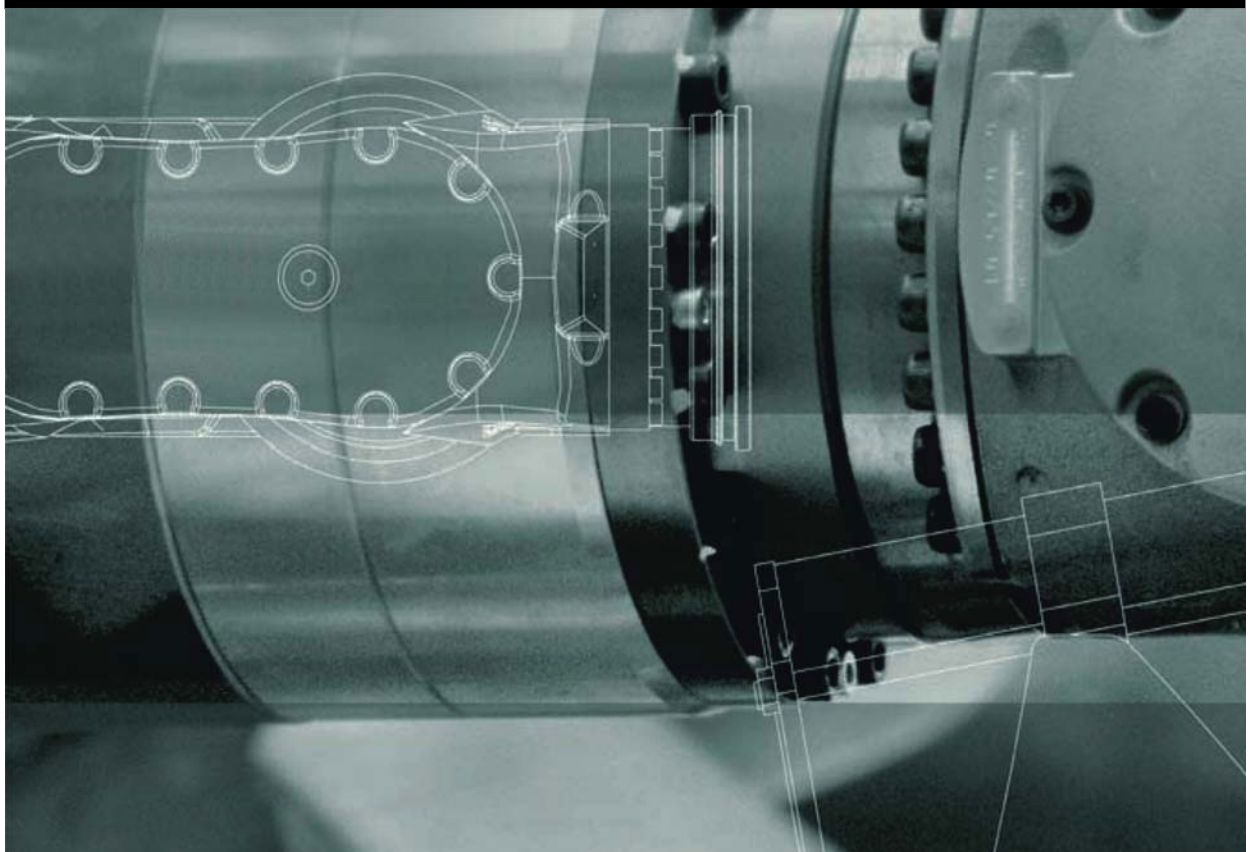


## KUKA.RobotSensorInterface 3.3

Für KUKA System Software 8.3 und 8.4



Stand: 07.11.2014

Version: KST RSI 3.3 V2

© Copyright 2014

KUKA Roboter GmbH  
Zugspitzstraße 140  
D-86165 Augsburg  
Deutschland

Diese Dokumentation darf – auch auszugsweise – nur mit ausdrücklicher Genehmigung der KUKA Roboter GmbH vervielfältigt oder Dritten zugänglich gemacht werden.

Es können weitere, in dieser Dokumentation nicht beschriebene Funktionen in der Steuerung lauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei Neulieferung bzw. im Servicefall.

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden jedoch regelmäßig überprüft und notwendige Korrekturen sind in der nachfolgenden Auflage enthalten.

Technische Änderungen ohne Beeinflussung der Funktion vorbehalten.

Original-Dokumentation

KIM-PS5-DOC

Publikation:	Pub KST RSI 3.3 (PDF) de
Buchstruktur:	KST RSI 3.3 V2.1
Version:	KST RSI 3.3 V2

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Zielgruppe	7
1.2	Dokumentation des Industrieroboters	7
1.3	Darstellung von Hinweisen	7
1.4	Verwendete Begriffe	8
1.5	Marken	9
<b>2</b>	<b>Produktbeschreibung</b>	<b>11</b>
2.1	Übersicht RobotSensorInterface	11
2.2	Funktionsweise Signalverarbeitung	12
2.3	Funktionsweise Datenaustausch	13
2.3.1	Datenaustausch über IO-System	13
2.3.2	Datenaustausch über Ethernet	13
2.4	Funktionsweise Sensorkorrektur	15
<b>3</b>	<b>Sicherheit</b>	<b>21</b>
3.1	Sicherheitshinweise	21
<b>4</b>	<b>Installation</b>	<b>23</b>
4.1	Systemvoraussetzungen	23
4.2	RobotSensorInterface installieren oder updaten	23
4.3	RobotSensorInterface deinstallieren	24
4.4	RSI Visual auf externen PC installieren	24
4.5	RSI Visual deinstallieren	25
<b>5</b>	<b>Konfiguration</b>	<b>27</b>
5.1	Netzwerkverbindung über das KLI der Robotersteuerung	27
5.2	Ethernet Sensor-Netzwerk konfigurieren	27
5.3	Globale Variablen in der RSI.DAT ändern	27
<b>6</b>	<b>Bedienung</b>	<b>29</b>
6.1	RSI Visual Übersicht Bedienoberfläche	29
6.1.1	Signalfluss-Editor öffnen	30
6.1.2	Signal-Ein- und -Ausgänge verknüpfen	30
6.1.3	Kommentar einfügen und verknüpfen	30
6.1.4	RSI Objekt-Parameter einstellen	31
6.1.5	RSI Objekt-Parameter freischalten	31
6.1.6	Signalfluss-Konfiguration speichern	31
6.1.7	Signalfluss-Konfiguration laden	32
6.2	RSI Monitor Übersicht Bedienoberfläche	32
6.2.1	Signaleigenschaften einstellen	33
6.2.2	Signalverlauf anzeigen	34
6.2.3	Signalaufzeichnung speichern	34
6.2.4	Signalaufzeichnung in den Monitor laden	34
<b>7</b>	<b>Programmierung</b>	<b>35</b>
7.1	Übersicht RSI Befehle	35
7.1.1	Zeichen und Schriftarten	35

7.1.2	RSI_CREATE()	35
7.1.3	RSI_DELETE()	36
7.1.4	RSI_ON()	37
7.1.5	RSI_OFF()	37
7.1.6	RSI_MOVECORR()	38
7.1.7	RSI_GETPUBLICPAR()	38
7.1.8	RSI_SETPUBLICPAR()	39
7.1.9	RSI_RESET()	39
7.1.10	RSI_CHECKID()	39
7.1.11	RSI_ENABLE()	40
7.1.12	RSI_DISABLE()	40
7.2	Verhalten der RSI Befehle	40
7.2.1	Verhalten RSI_ENABLE() / RSI_DISABLE()	40
7.2.2	Verhalten RSI_ON() / RSI_OFF()	41
7.3	Signalverarbeitung programmieren	42
7.3.1	Signalfluss in KRL-Programm einbinden	42
7.3.2	Signalfluss-Parameter in KRL ändern	43
7.4	XML-Datei für Ethernet-Verbindung konfigurieren	43
7.4.1	XML-Struktur für Verbindungseigenschaften	44
7.4.2	XML-Struktur für den Datenversand	44
7.4.3	XML-Struktur für den Datenempfang	46
7.4.4	Konfiguration nach XML-Schema	47
7.4.5	Schlüsselwörter - Daten lesen	48
7.4.6	Schlüsselwörter - Daten schreiben	49
<b>8</b>	<b>Beispiele</b>	<b>51</b>
8.1	Beispielkonfigurationen und -programme	51
8.1.1	Beispielapplikation implementieren	52
8.1.2	Bedienoberfläche Server-Programm	52
8.1.3	Kommunikationsparameter im Server-Programm einstellen	54
8.1.4	Beispiel einer kartesischen Korrektur über Ethernet	55
8.1.5	Beispiel einer sensorgeführten Kreisbewegung	56
8.1.6	Beispiel einer Bahnkorrektur zur Abstandssteuerung	59
8.1.7	Beispiel einer Transformation in neues Koordinatensystem	62
<b>9</b>	<b>Diagnose</b>	<b>67</b>
9.1	Diagnosedaten zur RSI anzeigen	67
9.2	Fehlerprotokoll (Logbuch)	67
9.2.1	LOG-Level konfigurieren	67
<b>10</b>	<b>Meldungen</b>	<b>69</b>
10.1	Meldungen während des Betriebs	69
<b>11</b>	<b>Anhang</b>	<b>71</b>
11.1	Speicher erhöhen	71
11.2	RSI Objekt-Bibliothek	71
11.2.1	RSI Objekte zur Korrekturüberwachung	71
11.2.2	RSI Objekte zur Signalübertragung	71
11.2.3	RSI Objekte zur Koordinatentransformation	72
11.2.4	RSI Objekte zur logischen Verknüpfung	72

11.2.5	RSI Objekte zur binären logischen Verknüpfung .....	72
11.2.6	RSI Objekte für mathematische Vergleiche .....	73
11.2.7	RSI Objekte für mathematische Operationen .....	73
11.2.8	RSI Objekte zur Signalsteuerung .....	73
11.2.9	RSI Objekte weitere .....	74
11.2.10	RSI Objekte für Aktionen .....	75
<b>12</b>	<b>KUKA Service .....</b>	<b>77</b>
12.1	Support-Anfrage .....	77
12.2	KUKA Customer Support .....	77
	<b>Index .....</b>	<b>85</b>



# 1 Einleitung

## 1.1 Zielgruppe

Diese Dokumentation richtet sich an Benutzer mit folgenden Kenntnissen:

- Fortgeschrittene KRL-Programmierkenntnisse
- Fortgeschrittene Systemkenntnisse der Robotersteuerung
- Fortgeschrittene Kenntnisse über Bussysteme
- Grundkenntnisse in XML
- Grundkenntnisse in der Digitaltechnik



Für den optimalen Einsatz unserer Produkte empfehlen wir unseren Kunden eine Schulung im KUKA College. Informationen zum Schulungsprogramm sind unter [www.kuka.com](http://www.kuka.com) oder direkt bei den Niederlassungen zu finden.

## 1.2 Dokumentation des Industrieroboters

Die Dokumentation zum Industrieroboter besteht aus folgenden Teilen:

- Dokumentation für die Robotermechanik
- Dokumentation für die Robotersteuerung
- Bedien- und Programmieranleitung für die System Software
- Anleitungen zu Optionen und Zubehör
- Teilekatalog auf Datenträger

Jede Anleitung ist ein eigenes Dokument.

## 1.3 Darstellung von Hinweisen

### Sicherheit

Diese Hinweise dienen der Sicherheit und **müssen** beachtet werden.



Diese Hinweise bedeuten, dass Tod oder schwere Verletzungen sicher oder sehr wahrscheinlich eintreten **werden**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise bedeuten, dass Tod oder schwere Verletzungen eintreten **können**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise bedeuten, dass leichte Verletzungen eintreten **können**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise bedeuten, dass Sachschäden eintreten **können**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise enthalten Verweise auf sicherheitsrelevante Informationen oder allgemeine Sicherheitsmaßnahmen.  
Diese Hinweise beziehen sich nicht auf einzelne Gefahren oder einzelne Vorsichtsmaßnahmen.

Dieser Hinweis macht auf Vorgehensweisen aufmerksam, die der Vorbeugung oder Behebung von Not- oder Störfällen dienen:

**SICHERHEITS-  
ANWEISUNGEN**

Mit diesem Hinweis gekennzeichnete Vorgehensweisen **müssen** genau eingehalten werden.

**Hinweise**

Diese Hinweise dienen der Arbeitserleichterung oder enthalten Verweise auf weiterführende Informationen.



Hinweis zur Arbeitserleichterung oder Verweis auf weiterführende Informationen.

**1.4 Verwendete Begriffe****RSI-Begriffe**

Begriff	Beschreibung
RSI	Robot Sensor Interface  Schnittstelle zur Kommunikation zwischen Industrieroboter und Sensorsystem
RSI Container	Ein RSI Container enthält den mit RSI Visual konfigurierten Signalfuss und muss im KRL-Programm angelegt werden.
RSI Container-ID	Kennung, die beim Anlegen des RSI Containers im KRL-Programm automatisch vergeben wird
RSI Kontext	Der RSI Kontext ist der mit RSI Visual konfigurierte Signalfuss, bestehend aus RSI Objekten und Verknüpfungen zwischen den RSI Objekten.
RSI Monitor	Monitor zur Online-Visualisierung von RSI Signalen
RSI Objekt	Der Signalfuss wird mithilfe von RSI Objekten konfiguriert, die über objektspezifische Signal-Ein- und -Ausgänge verknüpft werden.
RSI Objekt-Bibliothek	Bibliothek mit allen RSI Objekten, die zur Konfiguration des Signalfusses in RSI Visual zur Verfügung stehen
RSI Objekt-Parameter	Die RSI Objekt-Parameter bestimmen die Funktionalität eines RSI Objekts. Die Anzahl der RSI Objekt-Parameter ist spezifisch für jedes RSI Objekt.
RSI Visual	Grafischer Editor zur Konfiguration des Signalfusses (RSI Kontext)

**Allgemeine  
Begriffe**

Begriff	Beschreibung
CCS	Correction Coordinate System  Korrektur-Koordinatensystem im TCP für die kartesische Sensorkorrektur
Ethernet	Ethernet ist eine Datennetztechnologie für lokale Datennetze (LANs). Sie ermöglicht den Datenaustausch in Form von Datenrahmen zwischen den verbundenen Teilnehmern.
KLI	KUKA Line Interface  Linienbus zur Integration der Anlage in das Kundennetz
KR C	KUKA Robot Controller



Begriff	Beschreibung
KUKA smarHMI	KUKA smart Human-Machine Interface Bedienoberfläche der KUKA System Software
Sensormodus	Modus der Signalverarbeitung <ul style="list-style-type: none"> <li>■ IPO: Signalverarbeitung im Sensortakt von 12 ms</li> <li>■ IPO_FAST: Signalverarbeitung im Sensortakt von 4 ms</li> </ul>
Sensortakt	Takt, mit dem die Signalverarbeitung berechnet wird. Je nach Modus ist der Sensortakt 12 ms (Modus IPO) oder 4 ms (Modus IPO_FAST)
TTS	Toolbased Technological System  Das TTS ist ein Koordinatensystem, das auf der Bahn mitfährt. Es wird bei jeder LIN- oder CIRC-Bewegung berechnet. Es wird aus der Bahntangente, der +X-Richtung des TOOL-Koordinatensystems und dem daraus resultierenden Normalenvektor gebildet.  Das bahnbegleitende Koordinatensystem ist folgendermaßen definiert:  <b>X<sub>TTS</sub></b> : Bahntangente  <b>Y<sub>TTS</sub></b> : Normalenvektor zur Ebene gebildet aus Bahntangente und +X-Richtung des TOOL-Koordinatensystems  <b>Z<sub>TTS</sub></b> : Vektor des Rechtssystems gebildet aus <b>X<sub>TTS</sub></b> und <b>Y<sub>TTS</sub></b>  Die Bahntangente und die +X-Richtung des TOOL-Koordinatensystems dürfen nicht parallel sein, sonst kann das TTS nicht berechnet werden.
UDP	User Datagram Protocol  Verbindungsloses Protokoll über den Datenaustausch zwischen den Teilnehmern eines Netzwerks
IP	Internet Protocol  Das Internet-Protokoll hat die Aufgabe, Subnetze über physikalische MAC-Adressen zu definieren.
XML	Extensible Markup Language  Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer vorgegebenen Baumstruktur

## 1.5 Marken

**.NET Framework** ist eine Marke der Microsoft Corporation.

**Visual Studio** ist eine Marke der Microsoft Corporation.

**Windows** ist eine Marke der Microsoft Corporation.



## 2 Produktbeschreibung

### 2.1 Übersicht RobotSensorInterface

- Funktionen** RobotSensorInterface ist ein nachladbares Technologiepaket mit folgenden Funktionen:
- Datenaustausch zwischen Robotersteuerung und Sensorsystem
  - Datenaustausch über Ethernet oder das IO-System der Robotersteuerung
  - Zyklische Signalverarbeitung und -auswertung im Sensortakt
  - Beeinflussen der Roboterbewegung oder des Programmablaufs durch die Verarbeitung von Sensorsignalen
  - Konfigurieren des Signalflusses (RSI Kontext) mit dem grafischen Editor RSI Visual
  - Bibliothek mit RSI Objekten zum Konfigurieren des Signalflusses (RSI Kontext)
  - Online-Visualisierung der RSI Signale (RSI Monitor)

**Kommunikation** Die Robotersteuerung kann über das IO-System oder über Ethernet mit dem Sensorsystem kommunizieren.

Datenaustausch über IO-System:

- Die Daten und Signale des Sensorsystems werden über das IO-System gelesen und geschrieben. RobotSensorInterface greift auf die Daten und Signale zu und verarbeitet sie.



Signale werden über ein Bussystem mit dem IO-System der Robotersteuerung verknüpft:

- Allgemeine Informationen zur Busverwaltung und E/A-Verschaltung sind in der Dokumentation zu WorkVisual zu finden.
- Detaillierte Informationen zur Buskonfiguration sind in der Dokumentation zum Bussystem zu finden.

Datenaustausch über Ethernet:

- Die Robotersteuerung kommuniziert mit dem Sensorsystem über eine echtzeitfähige Netzwerkverbindung. Die Daten werden über das Ethernet UDP/IP-Protokoll übertragen. Es ist kein fester Datenrahmen vorgegeben. Der Benutzer muss den Datensatz in einer XML-Datei konfigurieren.

Eigenschaften:

- Zyklische Datenübertragung von der Robotersteuerung an ein Sensorsystem parallel zum Programmablauf (z. B. Positionsdaten, Achswinkel, Betriebsart etc.)
- Zyklische Datenübertragung von einem Sensorsystem an die Robotersteuerung parallel zum Programmablauf (z. B. Sensor-Messwerte)



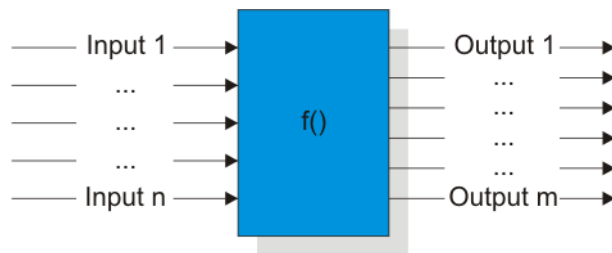
UDP ist ein verbindungsloses Netzwerkprotokoll für den Austausch von Datenpaketen. Der Datenaustausch über UDP ist unzuverlässig und ungesichert. Es kann beispielsweise nicht garantiert werden, dass gesendete Pakete zuverlässig ankommen oder in der Reihenfolge ankommen, in der sie gesendet wurden.

Der Datenaustausch über Ethernet eignet sich für Anwendungen, die unempfindlich auf verlorene oder in falscher Reihenfolge empfangene Pakete reagieren. Toleriert eine Anwendung dies nicht, muss der Programmierer geeignete Maßnahmen ergreifen, z. B. im Programm überprüfen, ob alle Pakete angekommen sind und bei Bedarf die Pakete neu anfordern.

## 2.2 Funktionsweise Signalverarbeitung

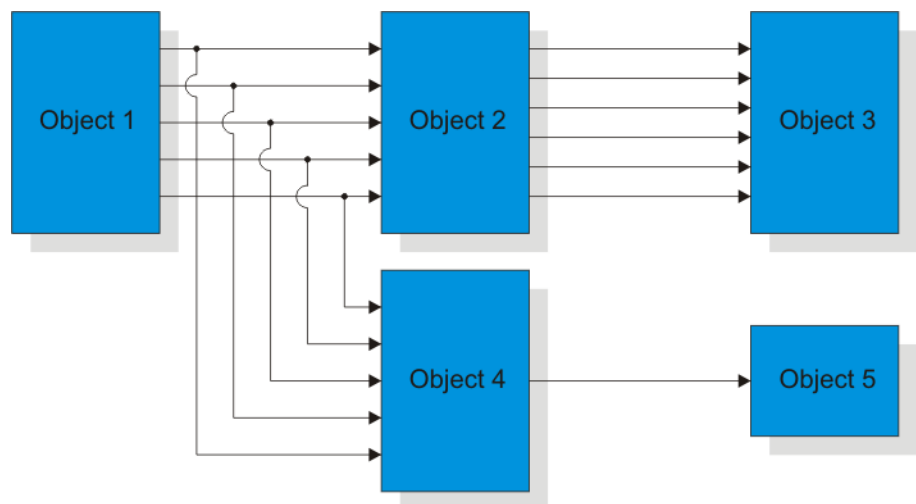
### Beschreibung

Eine Signalverarbeitung wird mit RSI Objekten aufgebaut. Ein RSI Objekt führt mit seinen Signaleingängen eine bestimmte Funktionalität aus und stellt das Ergebnis an den Signalausgängen bereit.



**Abb. 2-1: Schematischer Aufbau eines RSI Objekts**

RobotSensorInterface stellt dem Benutzer in einer Bibliothek einen umfangreichen Satz an RSI Objekten zur Verfügung. Durch die Verknüpfung der Signal-Ein- und Ausgänge mehrerer RSI Objekte entsteht ein Signalfluss. Der gesamte Signalfluss wird als RSI Kontext bezeichnet.



**Abb. 2-2: Schematischer Aufbau eines RSI Kontexts**

Der RSI Kontext wird mit dem grafischen Editor RSI Visual definiert und gespeichert. Im KRL-Programm kann der RSI Kontext geladen und die zum Programmablauf parallele Signalverarbeitung ein- und ausgeschaltet werden. Die Signalverarbeitung wird im Sensortakt berechnet. Je nach Modus ist der Sensortakt 12 ms (Modus IPO) oder 4 ms (Modus IPO\_FAST).

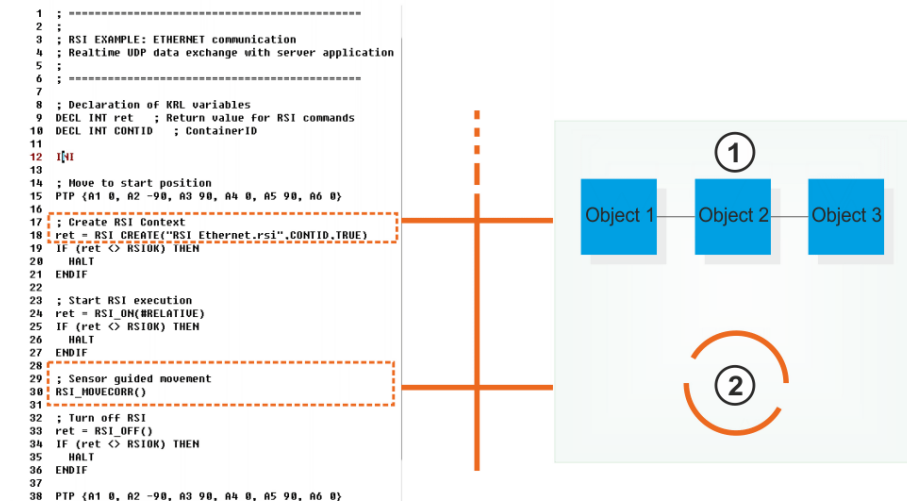


Abb. 2-3: Zusammenspiel KRL-Programm - Signalverarbeitung

1 RSI Kontext

2 Sensortakt

## 2.3 Funktionsweise Datenaustausch

### 2.3.1 Datenaustausch über IO-System

#### Beschreibung

Die Daten und Signale des Sensorsystems werden über das IO-System (\$IN, \$ANIN) der Robotersteuerung eingelesen. Die verarbeiteten Signale werden über das IO-System (\$OUT, \$ANOUT) wieder an das Sensorsystem zurückgegeben. Die Signale werden im Sensortakt gelesen und geschrieben.

Es werden folgende RSI-Objekte verwendet:

- ANIN und DIGIN greifen lesend auf das IO-System zu und übergeben die Daten und Signale des Sensorsystems an die Signalverarbeitung.
- MAP2ANOUT und MAP2DIGOUT greifen auf die verarbeiteten Signale zu und schreiben diese auf das IO-System.

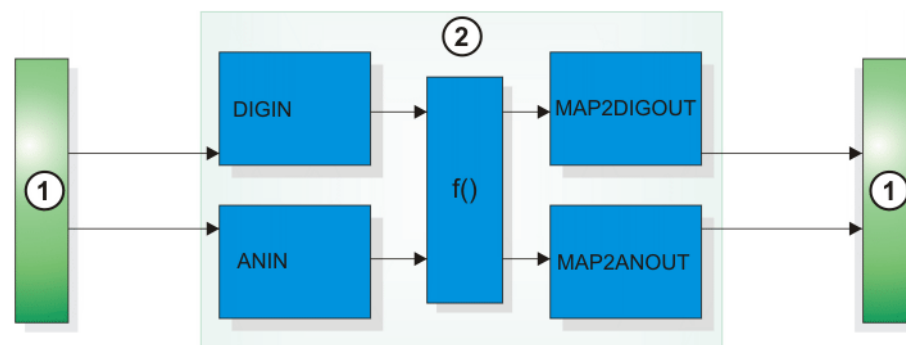


Abb. 2-4: Datenaustausch über IO-System

1 IO-System

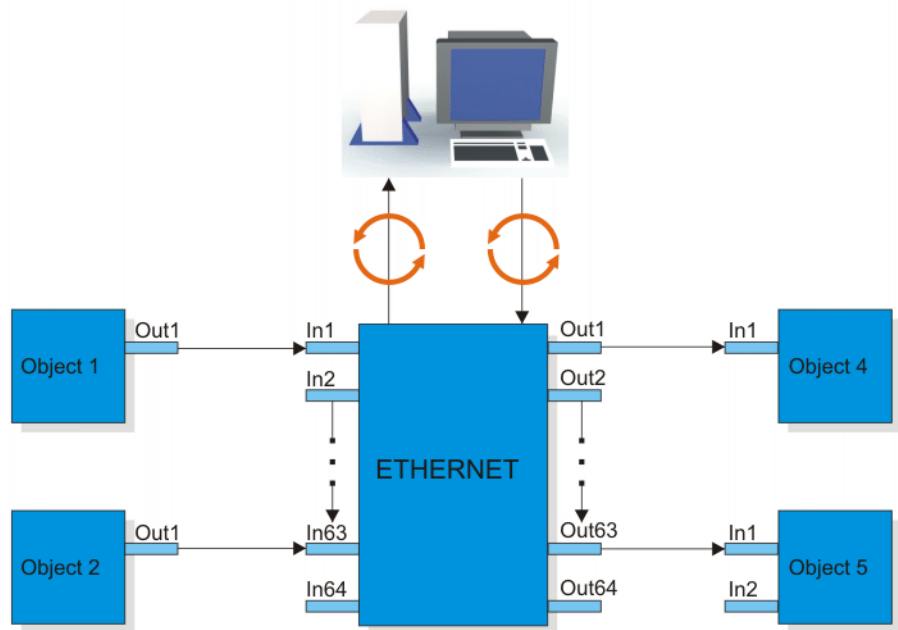
2 RSI Kontext

### 2.3.2 Datenaustausch über Ethernet

#### Beschreibung

Der Datenaustausch über Ethernet wird über das RSI Objekt ETHERNET realisiert.

Für ETHERNET können bis zu 64 Ein- und Ausgänge definiert werden. Die Signale an den Eingängen werden an das Sensorsystem gesendet. An den Ausgängen stehen die vom Sensorsystem empfangenen Daten zur Verfügung.



**Abb. 2-5: Datenaustausch über Ethernet (Funktionsprinzip)**

Mit dem Einschalten der Signalverarbeitung wird ein Kanal vorbereitet, der über das UDP/IP-Protokoll Daten an das Sensorsystem sendet. Die Robotersteuerung leitet den Datenaustausch mit einem Datenpaket ein und sendet im Sensortakt weitere Datenpakete an das Sensorsystem. Das Sensorsystem muss die erhaltenen Datenpakete mit einem eigenen Datenpaket beantworten.

Bei eingeschalteter Signalverarbeitung sendet und empfängt ETHERNET im Sensortakt einen benutzerdefinierten Datensatz im XML-Format. Dieser Datensatz muss in einer XML-Datei konfiguriert werden. Der Name der XML-Datei wird im ETHERNET-Objekt angegeben.

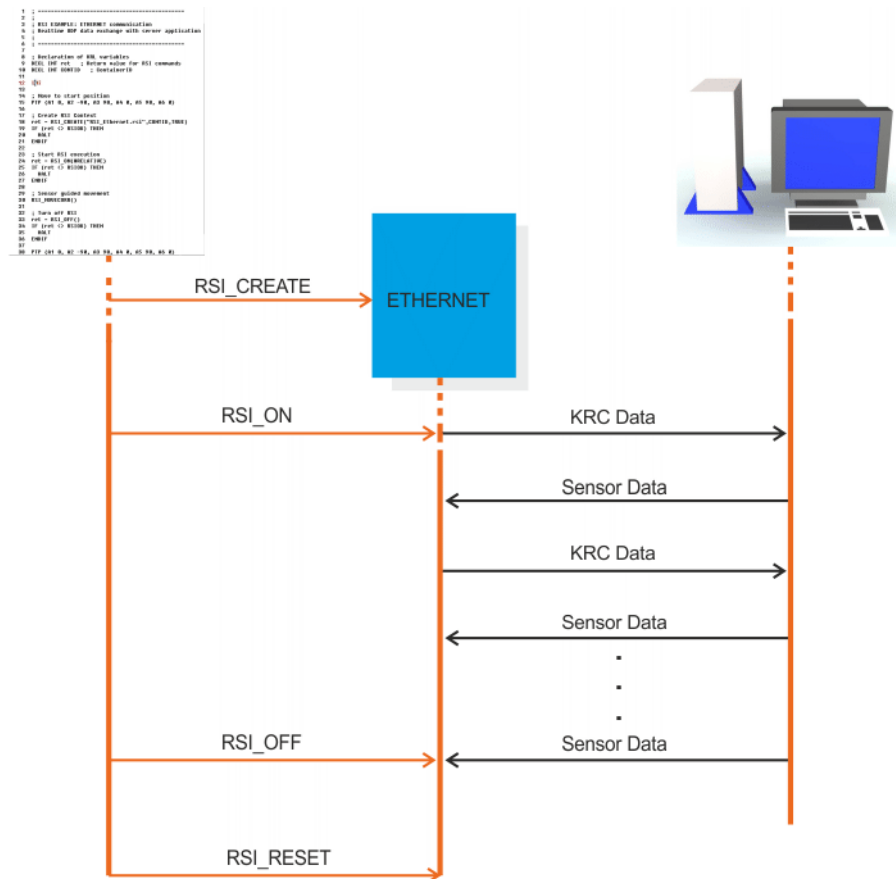


Abb. 2-6: Datenaustausch über Ethernet (Ablauf)

**Echtzeit-Anforderung**

Ein am Sensorsystem eintreffendes Datenpaket muss innerhalb des Sensortakts beantwortet werden. Zu spät eingetroffene Pakete werden verworfen.

Wenn die maximale Anzahl der zu spät beantworteten Datenpakete überschritten ist, stoppt der Roboter. Wenn die Signalverarbeitung ausgeschaltet wird, stoppt auch der Datenaustausch.

**2.4 Funktionsweise Sensorkorrektur**

Die Sensorkorrektur kann für asynchrone Achsen nicht verwendet werden.

**Übersicht**

RobotSensorInterface ermöglicht die kontinuierliche Einflussnahme auf die Roboterbewegung über Sensordaten. Dabei wird im Sensortakt ein Korrekturwert in die aktuelle Sollposition eingerechnet.

Folgende Korrekturtypen sind konfigurierbar:

- Bewegungen mit überlagerter Sensorkorrektur:
  - Achswinkel-Korrektur absolut oder relativ
  - Kartesische Korrektur absolut oder relativ
- Sensorgeführte Bewegung:
  - Achswinkel-Korrektur absolut oder relativ
  - Kartesische Korrektur absolut oder relativ

**HINWEIS**

Sensorkorrekturen beeinflussen die Roboterbewegung direkt. Nicht der Industrieroboter gibt die Bahn vor, sondern der Sensor. Der Benutzer ist dafür verantwortlich die Korrekturvorgaben des Sensors signaltechnisch so aufzubereiten, dass am Robotersystem keine mechanischen Beschädigungen, z. B. durch Schwingungen, auftreten können.

**Achswinkel-Korrektur**

Ein Korrekturwert kann achsweise auf die Roboterachsen A1 ... A6 und die Zusatzachsen E1 ... E6 aufgeschaltet werden.

Verwendete RSI Objekte:

- AXISCORR (Korrektur Roboterachsen)
- AXISCORREXT (Korrektur Zusatzachsen)

Die maximal zulässige Korrektur ist in beide Richtungen begrenzt.

**Kartesische Korrektur**

Ein Korrekturwert (Frame) kann die Roboterposition kartesisch verschieben. Das Korrekturframe bezieht sich auf ein Korrektur-Koordinatensystem (CCS) im TCP.

Für die Orientierung des Korrektur-Koordinatensystems stehen folgende Referenz-Koordinatensysteme zur Verfügung:

- BASE-Koordinatensystem
- ROBROOT-Koordinatensystem
- TOOL-Koordinatensystem
- WORLD-Koordinatensystem
- Technologie-Dreibein (TTS)

Verwendetes RSI Objekt:

- POSCORR

Die maximal zulässige kartesische Korrektur ist begrenzt.



Wenn RobotSensorInterface im RoboTeam verwendet wird, ist zu beachten, dass kartesische Sensorkorrekturen vom Master-Roboter nicht an die Slave-Roboter weitergegeben werden.

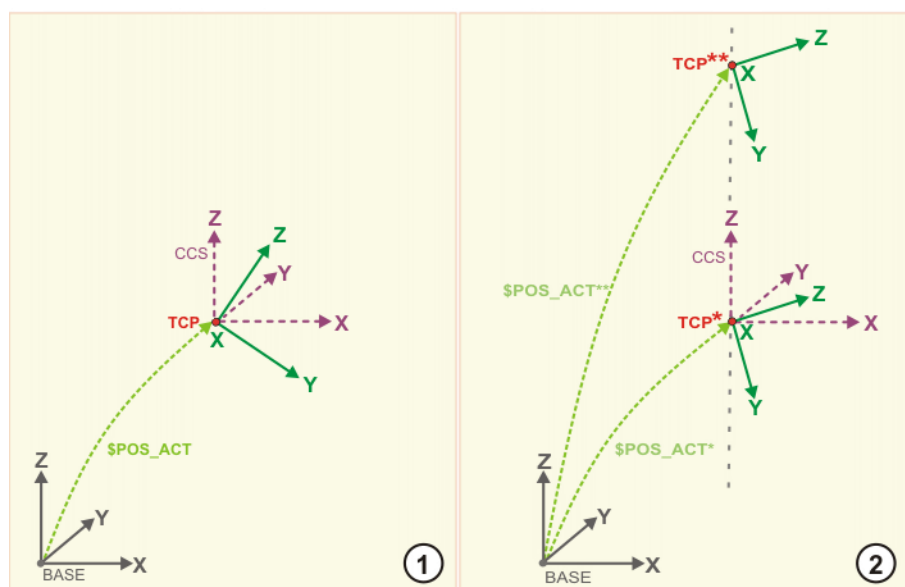


Abb. 2-7: Kartesische Korrektur bezogen auf BASE



Pos.	Beschreibung
1	<p>Ausgangsposition für die kartesische Korrektur</p> <ul style="list-style-type: none"> <li>■ <math>\\$POS\_ACT</math>: Kartesische Roboterposition</li> <li>■ CCS: Korrektur-Koordinatensystem im TCP mit der Orientierung von BASE</li> </ul>
2	<p>Kartesische Korrektur - Korrektur-Koordinatensystem ist das BASE-Koordinatensystem</p> <ul style="list-style-type: none"> <li>■ <math>\\$POS\_ACT^*</math>: Um den Korrekturwert verdrehte kartesische Roboterposition <ul style="list-style-type: none"> <li>■ TCP*: Der TCP ist im Korrektur-Koordinatensystem um +B verdreht</li> </ul> </li> <li>■ <math>\\$POS\_ACT^{**}</math>: Um den Korrekturwert verschobene und verdrehte kartesische Roboterposition <ul style="list-style-type: none"> <li>■ TCP**: Der TCP ist im Korrektur-Koordinatensystem in Richtung +Z verschoben und um +B verdreht</li> </ul> </li> </ul>

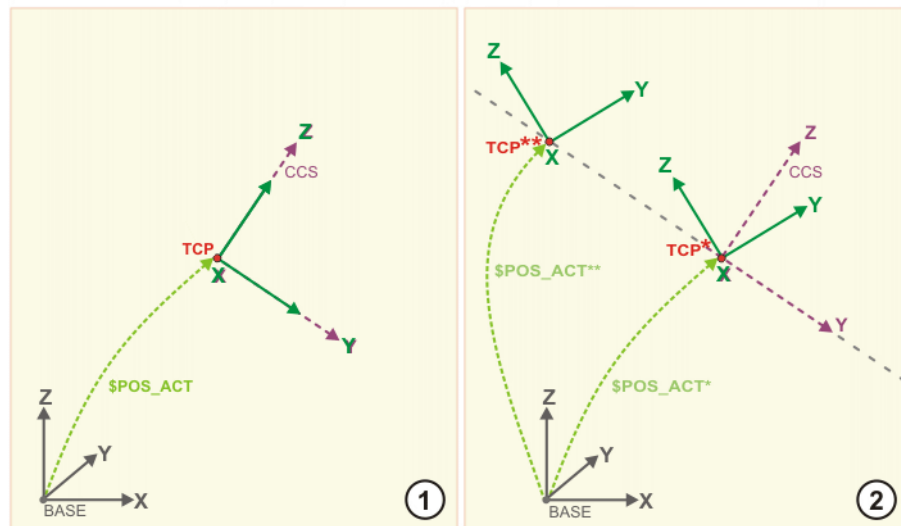



Abb. 2-8: Kartesische Korrektur bezogen auf TOOL

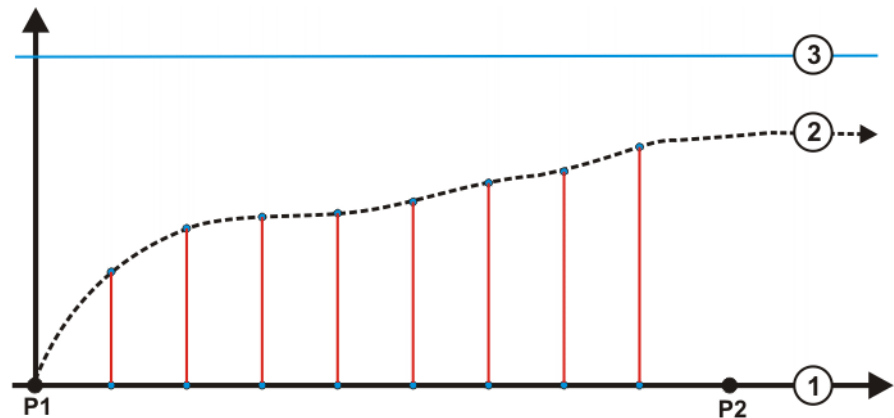
Pos.	Beschreibung
1	<p>Ausgangsposition für die kartesische Korrektur</p> <ul style="list-style-type: none"> <li>■ <math>\\$POS\_ACT</math>: Kartesische Roboterposition</li> <li>■ CCS: Korrektur-Koordinatensystem im TCP mit der Orientierung von TOOL</li> </ul>
2	<p>Kartesische Korrektur - Korrektur-Koordinatensystem ist das TOOL-Koordinatensystem</p> <ul style="list-style-type: none"> <li>■ <math>\\$POS\_ACT^*</math>: Um den Korrekturwert verdrehte kartesische Roboterposition <ul style="list-style-type: none"> <li>■ TCP*: Der TCP ist im Korrektur-Koordinatensystem um +C verdreht.</li> </ul> </li> <li>■ <math>\\$POS\_ACT^{**}</math>: Um den Korrekturwert verschobene und verdrehte kartesische Roboterposition <ul style="list-style-type: none"> <li>■ TCP**: Der TCP ist im Korrektur-Koordinatensystem in Richtung -Y verschoben und um +C verdreht.</li> </ul> </li> </ul>

**Absolutkorrektur**

Die neue Position ergibt sich aus der Verschiebung der Ausgangsposition um den aktuellen Korrekturwert.

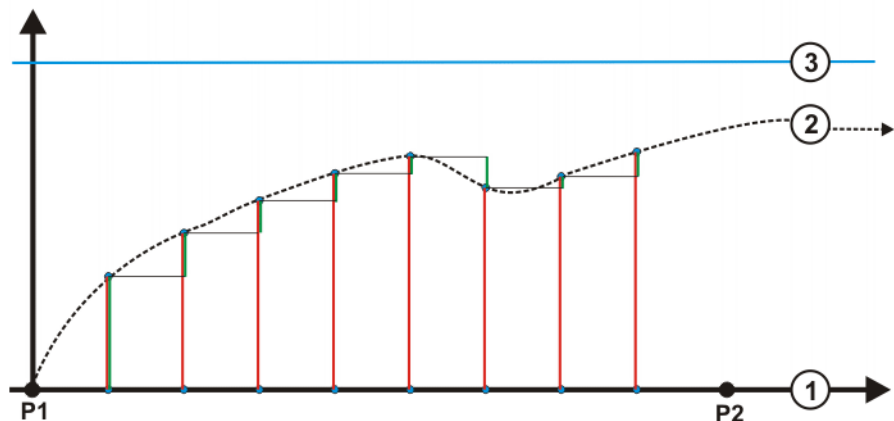
- Relativkorrektur** Die Korrekturwerte werden aufaddiert. Der neue Position ergibt sich aus der Verschiebung der Ausgangsposition um die bisherige Korrektur und den aktuellen Korrekturwert.
- Überlagerte Sensorkorrektur** Die Korrekturwerte werden auf die Stützpunkte einer programmierten Bahn aufgeschaltet. Der Bahnverlauf kann auf der Basis von absoluten oder relativen Korrekturwerten korrigiert werden.

 Wenn die Signale im IPO-Modus verarbeitet werden, kann der Bahnverlauf nur auf der Basis von LIN- und CIRC-Bewegungen korrigiert werden.



**Abb. 2-9: Bahnkorrektur auf Basis von Absolutwerten**

- 1 Programmierte Bahn
- 2 Korrigierte Bahn
- 3 Maximale Gesamtkorrektur
- Rot Absoluter Korrekturwert

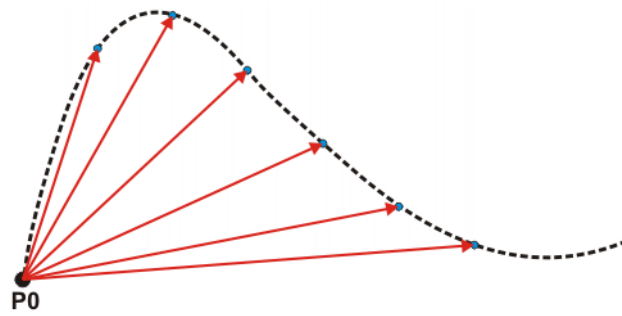


**Abb. 2-10: Bahnkorrektur auf Basis von Relativwerten**

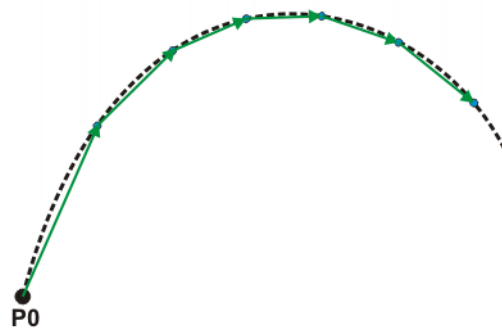
- 1 Programmierte Bahn
- 2 Korrigierte Bahn
- 3 Maximale Gesamtkorrektur
- Rot Gesamtkorrektur
- Grün Relativer Korrekturwert

- Sensorgeführte Bewegung** Mithilfe des Befehls RSI\_MOVECORR() kann eine sensorgeführte Bewegung programmiert werden. Ausgehend von einem Startpunkt fährt der Roboter keinen definierten Zielpunkt an, sondern wird rein korrekturgesteuert auf der Basis von Sensordaten verfahren.

Die sensorgeführte Bewegung kann auf der Basis von absoluten oder relativen Korrekturwerten ausgeführt werden.



**Abb. 2-11: Sensorgeführte Bewegung auf Basis von Absolutwerten**



**Abb. 2-12: Sensorgeführte Bewegung auf Basis von Relativwerten**



Das Fahrverhalten des Roboters bei einer sensorgeführten Bewegung unterscheidet sich von KR C2 und KR C4. An den KUKA-Support wenden, wenn die entsprechenden Anlagen von KR C2 auf KR C4 umgerüstet werden sollen.



## 3 Sicherheit

Diese Dokumentation enthält Sicherheitshinweise, die sich spezifisch auf die hier beschriebene Software beziehen.

Die grundlegenden Sicherheitsinformationen zum Industrieroboter sind im Kapitel "Sicherheit" der Bedien- und Programmieranleitung für Systemintegratoren oder der Bedien- und Programmieranleitung für Endanwender zu finden.



Das Kapitel "Sicherheit" in der Bedien- und Programmieranleitung der KUKA System Software (KSS) muss beachtet werden. Tod von Personen, schwere Verletzungen oder erhebliche Sachschäden können sonst die Folge sein.

### 3.1 Sicherheitshinweise

#### Sensorbetrieb

- Bei falscher Anwendung der RobotSensorInterface kann es zu Personen- und Sachschäden kommen.
- Im Sensorbetrieb kann sich der Roboter in folgenden Fällen unerwartet bewegen:
  - Bei falsch parametrisierten RSI Objekten
  - Bei einem Hardware-Fehler (z. B. falsche Verkabelung, Kabelbruch in der Sensorleitung oder Fehlfunktion des Sensors)
- Bei unerwarteten Bewegungen kann es zu schweren Verletzungen und erheblichen Sachschäden kommen. Der Systemintegrator ist verpflichtet, das Verletzungsrisiko für sich und andere Personen sowie Sachschaden durch den Einsatz geeigneter Sicherheitsmaßnahmen zu minimieren, z. B. durch die Begrenzung des Arbeitsraumes.
- Beim Start der Signalverarbeitung mit RobotSensorInterface gibt die Sicherheitssteuerung in der Betriebsart T1 oder T2 eine Quittiermeldung aus:

**!!! Achtung – Sensorkorrektur wird aktiviert !!!**

#### Arbeitsraum-Begrenzung

- Die Achsbereiche aller Roboterachsen sind über einstellbare Software-Endschalter begrenzt. Diese Software-Endschalter sind so einzustellen, dass der Arbeitsraum des Roboters auf den für den Prozess minimal notwendigen Bereich begrenzt ist.
- Mit der System Software können maximal 8 kartesische und 8 achsspezifische Arbeitsräume konfiguriert werden. Der Systemintegrator muss die Arbeitsräume so konfigurieren, dass sie auf den für den Prozess minimal notwendigen Bereich begrenzt sind. Damit wird die Gefahr von Schäden durch unerwartete Bewegungen im Sensorbetrieb auf ein Minimum reduziert.



Weitere Informationen zur Konfiguration von Arbeitsräumen sind in der Bedien- und Programmieranleitung für Systemintegratoren zu finden.

#### Sensorkorrektur

RobotSensorInterface überwacht und begrenzt die maximale Sensorkorrektur. Es kann jedes einzelne Korrekturobjekt überwacht werden sowie die Gesamtkorrektur über alle Korrekturobjekte.

Objektbezogene Sensorkorrekturen sind defaultmäßig auf maximal +/- 5 mm oder 5° begrenzt, die Gesamtkorrektur auf maximal +/- 6 mm oder 6°.

Bei Überschreiten einer objektbezogenen Korrektur läuft die Signalverarbeitung weiter und die Korrektur wird automatisch auf die zulässige Maximalkorrektur begrenzt. Bei Überschreiten der zulässigen Gesamtkorrektur wird die Signalverarbeitung gestoppt.




## 4 Installation

### 4.1 Systemvoraussetzungen

- Hardware**
- Robotersteuerung KR C4
  - Für den Datenaustausch über Ethernet:
    - Prozessorgestütztes externes System mit echtzeitfähigem Betriebssystem und echtzeitfähiger Netzwerkkarte mit 100 MBit im Voll-Duplex-Modus
    - Mikroprozessorgestützter Sensor mit echtzeitfähiger Netzwerkkarte bei Sensorik-Applikationen
    - Netzkabel für Switch, Hub oder gekreuztes Netzkabel zur direkten Verbindung
  - Für den Datenaustausch über IO-System: Bussystem, z. B. Profinet
  - Externer PC für die Signalfluss-Konfiguration mit RSI Visual
- Empfohlene Roboter**
- RobotSensorInterface sollte nur in Kombination mit KUKA 6-Achs-Robotern verwendet werden. Den Einsatz anderer Roboter nur nach Rücksprache mit der KUKA Roboter GmbH planen.
- (>>> 12 "KUKA Service" Seite 77)
- Software**
- Robotersteuerung:
- KUKA System Software 8.3 oder 8.4
- Externer PC:
- Betriebssystem Windows mit .Net Framework 3.5 inkl. Service Pack 1
- KRL-Ressourcen**
- Für RSI Korrekturen im IPO-Modus müssen folgende KRL-Ressourcen frei sein:
- | KRL-Ressource      | Nummer |
|--------------------|--------|
| Funktionsgenerator | 1      |
- Kompatibilität**
- RobotSensorInterface darf nicht mit folgenden Technologiepaketen auf der gleichen Robotersteuerung installiert sein:
    - KUKA.ConveyorTech
    - KUKA.ServoGun TC
    - KUKA.ServoGun FC
    - KUKA.EqualizingTech
  - Wenn RobotSensorInterface und KUKA.RoboTeam auf der gleichen Robotersteuerung installiert sind, ist zu beachten, dass kartesische Sensor-korrekturen vom Master-Roboter nicht an die Slave-Roboter weitergegeben werden.

### 4.2 RobotSensorInterface installieren oder updaten



Es wird empfohlen, vor dem Update einer Software alle zugehörigen Daten zu archivieren.

- Voraussetzung**
- Software auf KUKA.USBData-Stick
  - Es ist kein Programm angewählt.
  - Betriebsart T1 oder T2
  - Benutzergruppe Experte

**HINWEIS**

Es darf ausschließlich der KUKA.USBData-Stick verwendet werden. Wenn ein anderer USB-Stick verwendet wird, können Daten verloren gehen oder verändert werden.

**Vorgehensweise**

1. USB-Stick anstecken.
2. Im Hauptmenü **Inbetriebnahme > Zusatzsoftware installieren** wählen.
3. Auf **Neue Software** drücken. Wenn eine Software, die sich auf dem USB-Stick befindet, nicht angezeigt wird, auf **Aktualisieren** drücken.
4. Den Eintrag **RSI** markieren und auf **Installieren** drücken. Sicherheitsabfrage mit **Ja** beantworten. Die Dateien werden auf die Festplatte kopiert.
5. Wenn eine weitere Software von diesem Stick installiert werden soll, Schritt 4 wiederholen.
6. USB-Stick entfernen.
7. Abhängig von der Zusatzsoftware kann ein Neustart notwendig sein. In diesem Fall wird eine Aufforderung zum Neustart angezeigt. Mit **OK** bestätigen und die Robotersteuerung neu starten. Die Installation wird fortgesetzt und abgeschlossen.

**LOG-Datei**

Es wird eine LOG-Datei unter C:\KRC\ROBOTER\LOG erstellt.

**4.3 RobotSensorInterface deinstallieren**

Es wird empfohlen, vor der Deinstallation einer Software alle zugehörigen Daten zu archivieren.

**Voraussetzung**

- Benutzergruppe Experte

**Vorgehensweise**

1. Im Hauptmenü **Inbetriebnahme > Zusatzsoftware installieren** wählen. Alle installierten Zusatzprogramme werden angezeigt.
2. Den Eintrag **RSI** markieren und auf **Deinstallieren** drücken. Sicherheitsabfrage mit **Ja** beantworten. Die Deinstallation wird vorbereitet.
3. Die Robotersteuerung neu starten. Die Deinstallation wird fortgesetzt und abgeschlossen.

**LOG-Datei**

Es wird eine LOG-Datei unter C:\KRC\ROBOTER\LOG erstellt.

**4.4 RSI Visual auf externen PC installieren****Vorbereitung**

- Den Ordner **RSIVisual** auf externen PC kopieren:
  - Von KUKA.USBData-Stick
  - Oder von der Robotersteuerung im Verzeichnis D:\KUKA\_OPT\RSI, wenn die Software vorinstalliert ist

**Voraussetzung**

- Lokale Administratorrechte

**Vorgehensweise**

1. Programm **setup.exe** im Ordner **RSIVisual** starten.
2. Es öffnet sich ein Installations-Assistent für RSI Visual. Den Anweisungen des Installations-Assistenten folgen.
3. RSI Visual wird defaultmäßig in den Ordner C:\Programme\KUKA Roboter GmbH\RSIVisual installiert.  
Wenn gewünscht, ein anderes Verzeichnis auswählen.
4. Wenn die Installation abgeschlossen ist, auf **Close** klicken, um den Installations-Assistenten zu schließen.



## 4.5 RSI Visual deinstallieren

**Voraussetzung** ■ Lokale Administratorrechte

**Vorgehensweise**

1. Im Windows-Start-Menü unter **Einstellungen** > **Systemsteuerung** > **Software** den Eintrag **RSIVisual** entfernen.
2. Im Verzeichnis C:\Programme\KUKA Roboter GmbH den Ordner **RSIVisual** löschen.



## 5 Konfiguration

### 5.1 Netzwerkverbindung über das KLI der Robotersteuerung

**Beschreibung** Für den Datenaustausch über Ethernet muss eine Netzwerkverbindung über das KLI der Robotersteuerung hergestellt werden. RSI benötigt hierzu ein eigenes, von anderen KLI-Subnetzwerken unabhängiges Ethernet Sensor-Netzwerk.

Je nach Spezifikation stehen an der Kundenschnittstelle der Robotersteuerung folgende Ethernet-Schnittstellen als Option zur Verfügung:

- Schnittstelle X66 (1 Steckplatz)
- Schnittstelle X67.1-3 (3 Steckplätze)



Weitere Informationen zu den Ethernet-Schnittstellen sind in der Betriebs- oder Montageanleitung der Robotersteuerung zu finden.

### 5.2 Ethernet Sensor-Netzwerk konfigurieren

**Voraussetzung**

- Benutzergruppe Experte
- Netzwerkverbindung über das KLI der Robotersteuerung

**Vorgehensweise**

1. Im Hauptmenü **Inbetriebnahme** > **Service** > **HMI minimieren** wählen.
2. Im Windows-Start-Menü unter **Alle Programme** > **RSI-Network** wählen.  
Das Fenster **Network Setup** öffnet sich. Bereits eingerichtete Netzwerkverbindungen werden in der Baumstruktur unter **Other Installed Interfaces** angezeigt.
3. In der Baumstruktur unter **RSI Ethernet** den Eintrag **New** markieren und auf **Edit** drücken.
4. Die IP-Adresse eintragen und mit **Ok** bestätigen.



Der IP-Adressenbereich 192.168.0.x ist für die Konfiguration der Netzwerkverbindung gesperrt. Die eingetragene IP-Adresse muss in einem eigenen Subnetz liegen, die Adresse darf nicht im Adressbereich eines anderen KLI-Subnetz liegen.

5. Robotersteuerung mit einem Kaltstart neu starten.

### 5.3 Globale Variablen in der RSI.DAT ändern

In der Datei KRC:\R1\TP\RSI\RSI.DAT sind globale Variablen definiert. Nur die hier beschriebenen Variablen können geändert werden.

**Voraussetzung**

- Benutzergruppe Experte

**Beschreibung**

```
DEFDAT RSI PUBLIC
...
RSI global Variables:
GLOBAL BOOL RSIERRMSG=TRUE
...
; Flag for writing context information
GLOBAL INT RSITECHIDX=1
; Tech Channel used for RSI corrections

ENDDAT
```

Variable	Beschreibung
RSIERRMSG	<p>TRUE = Fehler beim Ausführen von RSI Befehlen werden mit einer Quittiermeldung auf der smartHMI angezeigt.</p> <p>FALSE = Keine Quittiermeldung. Zur Fehlerbehandlung müssen die Rückgabewerte der RSI Befehle im KRL-Programm ausgewertet werden.</p> <p>Default: TRUE</p>
RSITECHIDX	<p>Funktionsgenerator für RSI Korrekturen im IPO-Modus</p> <p>Default-Wert: <b>1</b></p> <p>Die maximale Anzahl der Funktionsgeneratoren ist in den Maschinendaten definiert (\$TECH_MAX).</p>

## 6 Bedienung

### 6.1 RSI Visual Übersicht Bedienoberfläche

Die Bedienoberfläche steht je nach Auswahl bei der Installation in folgenden Sprachen zur Verfügung:

- Deutsch
- Englisch

Auf der Bedienoberfläche sind nicht alle Elemente defaultmäßig sichtbar, sondern sie können nach Bedarf ein- oder ausgeblendet werden.

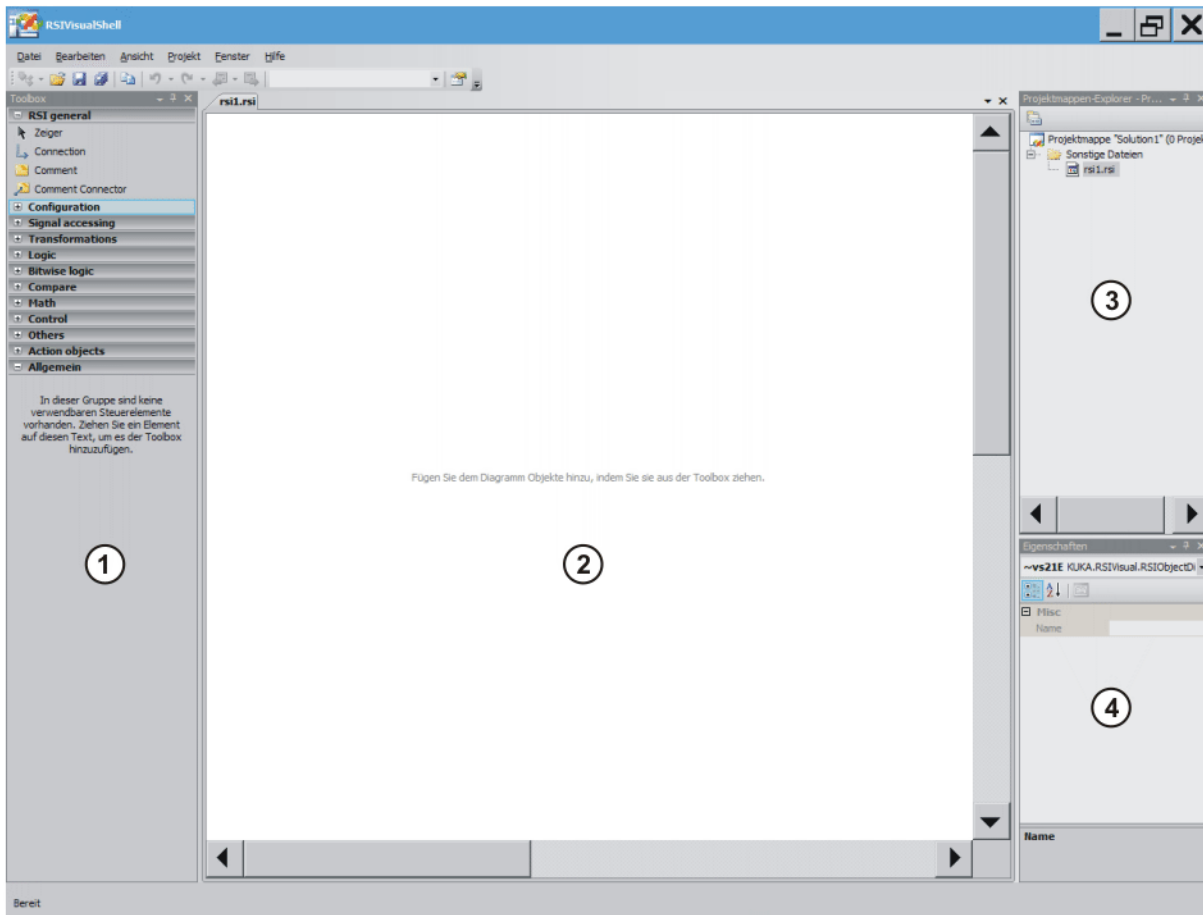


Abb. 6-1: Übersicht Bedienoberfläche

Pos.	Beschreibung
1	<b>Fenster <b>Toolbox</b></b> Enthält alle notwendigen Werkzeuge und RSI Objekte zur Konfiguration des RSI Kontextes. Die RSI Objekte können per Drag&Drop in den Signalfuss-Editor eingefügt werden. Werkzeuge unter <b>RSI general</b> : <ul style="list-style-type: none"> <li>■ <b>Comment</b>: Ein Kommentar-Objekt kann per Drag&amp;Drop in den Editor eingefügt werden.</li> <li>■ <b>Comment Connector</b>: Ein Kommentar-Objekt kann mit dem zugehörigen RSI Objekt verknüpft werden.</li> </ul> Die Beschreibung der RSI Objekte ist im Anhang zu finden. ( <a href="#">&gt;&gt;&gt; 11.2 "RSI Objekt-Bibliothek" Seite 71</a> )
2	<b>Signalfuss-Editor</b> Hier wird die Signalfuss-Konfiguration erstellt.
3	<b>Fenster <b>Projektmappen-Explorer</b></b> In diesem Fenster werden alle geladenen Dateien in einer Baumstruktur angezeigt.
4	<b>Fenster <b>Eigenschaften</b></b> Wird ein RSI Objekt, ein RSI Objekt-Parameter oder ein Signal-Ein oder -Ausgang im Signalfuss-Editor markiert, werden dessen Eigenschaften angezeigt. Einzelne Eigenschaften oder Parameter können geändert werden.

### 6.1.1 Signalfuss-Editor öffnen



- Vorgehensweise**
1. Menüfolge **Datei > Neu > Datei...** wählen.
  2. Die Vorlage **rsi** mit **Öffnen** laden.  
Ein leeres Dokument steht für die Signalfuss-Konfiguration zur Verfügung.

### 6.1.2 Signal-Ein- und -Ausgänge verknüpfen

**Beschreibung** Der Signalfuss wird mithilfe der RSI Objekte konfiguriert, die per Drag&Drop in den Signalfuss-Editor eingefügt und über die objektspezifischen Signal-Ein- und -Ausgänge miteinander verknüpft werden. Ein Signal-Ausgang kann mit mehreren Signal-Eingängen verknüpft sein.

- Vorgehensweise**
1. Mit dem Maus-Zeiger auf den gewünschten Objekt-Ausgang zeigen.
  2. Wenn das Verknüpfungssymbol am Ausgang angezeigt wird, klicken und mit dem Maus-Zeiger auf den gewünschten Objekt-Eingang zeigen.
  3. Wenn das Verknüpfungssymbol am Eingang angezeigt wird, erneut klicken.

#### Symbole



Symbol	Beschreibung
	Verknüpfungssymbol am Signal-Ausgang
	Verknüpfungssymbol am Signal-Eingang

### 6.1.3 Kommentar einfügen und verknüpfen

- Vorgehensweise**
1. Kommentar-Objekt per Drag&Drop in den Editor einfügen.

2. Textfeld markieren und Kommentar eingeben.
3. In der **Toolbox** das Werkzeug **Comment Connector** auswählen.
4. Mit dem Maus-Zeiger auf den Kommentar zeigen.
5. Wenn das Verknüpfungssymbol am Kommentar angezeigt wird, klicken und mit dem Maus-Zeiger auf das gewünschte RSI Objekt zeigen.
6. Wenn das Verknüpfungssymbol am RSI Objekt angezeigt wird, erneut klicken.

### Symbole

Symbol	Beschreibung
	Verknüpfungssymbol am Kommentar
	Verknüpfungssymbol am RSI Objekt

### 6.1.4 RSI Objekt-Parameter einstellen

- Vorgehensweise**
1. RSI Objekt-Parameter im Signalfluss-Editor markieren.  
Im Fenster **Eigenschaften** werden die Eigenschaften des Parameters angezeigt.
  2. Im Feld **Value** den gewünschten Wert eingeben oder auswählen.

### 6.1.5 RSI Objekt-Parameter freischalten

**Beschreibung** Es ist möglich, den Wert eines RSI Objekt-Parameters im KRL-Programm auszulesen und dem Objekt-Parameter nachträglich einen neuen Wert zuzuweisen.

(>>> 7.3.2 "Signalfluss-Parameter in KRL ändern" Seite 43)

Voraussetzung dafür ist, dass der Parameter in der Signalfluss-Konfiguration freigeschaltet ist.

- Vorgehensweise**
1. RSI Objekt-Parameter im Signalfluss-Editor markieren.  
Im Fenster **Eigenschaften** werden die Eigenschaften des Parameters angezeigt.
  2. Feld **IsPublic** auf **True** setzen.



Bei bestimmten Objekten ist das Setzen von Parameter zur Laufzeit vom RSI unzulässig. In diesem Fall kann das Feld IsPublic nicht auf True gesetzt werden.

### 6.1.6 Signalfluss-Konfiguration speichern

**Beschreibung** Beim Speichern der Signalfluss-Konfiguration werden folgende Dateien erzeugt:

- **<Dateiname>.rsi**: Signalfluss-Konfiguration von RSI Visual
- **<Dateiname>.rsi.diagram**: Signalfluss-Layout von RSI Visual nach XML-Schema
- **<Dateiname>.rsi.xml**: XML-Datei für die Signalverarbeitung auf der Robotersteuerung



RSI-, DIAGRAM- und XML-Datei sind eine Einheit und müssen zusammen auf die Robotersteuerung übertragen werden.

Zielverzeichnis:

C:\KRC\Roboter\Config\User\Common\SensorInterface

- Vorgehensweise**
1. Menüfolge **Datei** > **<Dateiname>.rsi speichern** oder **<Dateiname>.rsi speichern unter...** wählen.
  2. Einen Dateinamen für die Konfiguration vergeben und mit **Speichern** im gewünschten Verzeichnis speichern.

### 6.1.7 Signalfluss-Konfiguration laden

- Vorgehensweise**
1. Menüfolge **Datei** > **Öffnen** > **Datei...** wählen.
  2. Gewünschte RSI-Datei mit **Öffnen** laden.

## 6.2 RSI Monitor Übersicht Bedienoberfläche

- Aufruf**
- Im Hauptmenü **Anzeige** > **RSI Monitor** wählen.

**Beschreibung** RSI Monitor kann bis zu 24 Signale aus dem RSI Kontext anzeigen und aufzeichnen. Im RSI Kontext wird hierzu das RSI Objekt MONITOR verwendet. In der Signalfluss-Konfiguration müssen die anzuzeigenden Signale mit den Eingängen des MONITOR-Objekts verknüpft sein.



**Abb. 6-2: Übersicht Bedienoberfläche**

Folgende Schaltflächen stehen zur Verfügung:

Schaltfläche	Beschreibung
<b>Setup</b>	Die Signaleigenschaften für die Signalaufzeichnung können definiert werden.
<b>Datei</b>	Der aufgezeichnete Signalverlauf kann in einer Datei gespeichert oder eine Datei kann geladen werden.



Schaltfläche	Beschreibung
<b>Einstellung</b>	<p>Die Kanalnummer des MONITOR RSI-Objekts kann eingestellt werden. (Relevant, wenn im RSI Kontext mehrere MONITOR-Objekte verwendet werden.)</p> <p>■ 1 ... 8</p> <p>Default: 1</p> <p>Diese Schaltfläche steht in der Benutzergruppe Anwender nicht zur Verfügung.</p>
<b>Zoom</b>	<p>Über einen Regler kann das angezeigte Zeitfenster vergrößert oder verkleinert werden.</p> <p>Das sichtbare Zeitfenster kann durch horizontales Schieben auf dem Monitor-Anzeigefenster verschoben werden.</p>

### 6.2.1 Signaleigenschaften einstellen

#### Beschreibung



Abb. 6-3: RSI Monitor – Signaleigenschaften

Pos.	Beschreibung
1	Listefeld mit den Signalen 1 ... 24
2	<p>Linienstärke des Signals</p> <p>■ <b>Linienstärke1 ... 4</b></p> <p>Default: <b>Linienstärke1</b></p>
3	Signalfarbe des Signals
4	<p>Ordinate, auf die das Signal bezogen wird</p> <p>■ <b>Links</b>: Darstellung bezüglich linker Ordinate</p> <p>■ <b>Rechts</b>: Darstellung bezüglich rechter Ordinate</p> <p>Default: <b>Links</b></p> <p>Die Skalierung der Ordinate richtet sich automatisch nach dem größten, ihr zugeordnetem Signal.</p>
5	<p>Die Checkbox muss aktiviert sein (Häkchen gesetzt), um ein Signal im Monitor anzuzeigen.</p> <p>Default: Checkbox für <b>Signal 1 ... 6</b> aktiviert.</p>

Folgende Schaltflächen stehen zur Verfügung:

Schaltfläche	Beschreibung
<b>Alle aktivieren</b>	Aktiviert alle Signale.
<b>Alle deaktivieren</b>	Deaktiviert alle Signale.
<b>Stärke zurücksetzen</b>	Setzt die Linienstärken der Signale auf die voreingestellte Linienstärke 1 zurück.
<b>Farben zurücksetzen</b>	Setzt die Signalfarben auf die voreingestellten Farben zurück.
<b>Zeichnungsfläche zurücksetzen</b>	Setzt das Zeitfenster auf die voreingestellte Größe zurück (Zoom aufheben)
<b>Alles Zurücksetzen</b>	Setzt alle Signaleigenschaften auf die Voreinstellung zurück.

### 6.2.2 Signalverlauf anzeigen

**Beschreibung** Jedes MONITOR-Objekt verwendet einen eigenen Kanal zum RSI Monitor. Wenn im RSI Kontext mehrere MONITOR-Objekte verwendet werden, muss die Kanalnummer des gewünschten MONITOR-Objekts für die Signalaufzeichnung eingestellt werden. RSI Monitor zeigt nur die über den eingestellten Kanal empfangenen Signale an.

**Voraussetzung** ■ IP-Adresse im RSI Objekt MONITOR: 192.168.0.1

**Vorgehensweise**

1. RSI Monitor aufrufen und auf **Setup** drücken.
2. Signaleigenschaften für die Aufzeichnung einstellen.
3. Bei Bedarf zur Benutzergruppe Experte wechseln und auf **Einstellung** drücken, um die Kanalnummer des MONITOR-Objekts einzustellen.
4. Programm anwählen und abfahren.

Die Aufzeichnung startet mit dem Einschalten der Signalverarbeitung und endet mit den Ausschalten der Signalverarbeitung.



Eine Signalaufzeichnung wird im RSI Monitor erst gelöscht, wenn ein MONITOR-Objekt im KRL-Programm neu angelegt wird. Nach dem Zurücksetzen des Programms oder dem Löschen der Signalverarbeitung bleibt die Signalaufzeichnung im RSI Monitor stehen.

### 6.2.3 Signalaufzeichnung speichern

**Vorgehensweise**

1. Checkbox **Datei** aktivieren.
2. Im Feld **Datei speichern** einen Dateinamen für die Aufzeichnung vergeben und auf **Speichern** drücken.

Die Aufzeichnung wird als DAT-Datei im Verzeichnis C:\KRC\ROBOTER\LOG\SensorInterface\MONITOR gesichert.

### 6.2.4 Signalaufzeichnung in den Monitor laden

**Vorgehensweise**

1. Checkbox **Datei** aktivieren.
2. Im Feld **Datei laden** die gewünschte Datei auswählen und auf **Laden** drücken.

Es stehen alle Aufzeichnungen zur Auswahl, die im Verzeichnis C:\KRC\ROBOTER\LOG\SensorInterface\MONITOR gesichert wurden.

## 7 Programmierung

### 7.1 Übersicht RSI Befehle

Zur Programmierung der Signalverarbeitung stellt RobotSensorInterface Funktionen zur Verfügung. Jede dieser Funktionen, ausgenommen RSI\_MOVECORR(), besitzt einen Rückgabewert. Der Rückgabewert kann im KRL-Programm abgefragt und ausgewertet werden.

Als Fehlercodes sind Konstanten in der Datenliste RSI.DAT im Verzeichnis KRC:\R1\TP\RSI deklariert. Um zu überprüfen, ob ein RSI Befehl korrekt ausgeführt wurde, können die in den Funktionsbeschreibungen angegebenen Konstanten verwendet werden.

Funktion	Beschreibung
RSI_CREATE()	(>>> 7.1.2 "RSI_CREATE()" Seite 35)
RSI_DELETE()	(>>> 7.1.3 "RSI_DELETE()" Seite 36)
RSI_ON()	(>>> 7.1.4 "RSI_ON()" Seite 37)
RSI_OFF()	(>>> 7.1.5 "RSI_OFF()" Seite 37)
RSI_MOVECORR()	(>>> 7.1.6 "RSI_MOVECORR()" Seite 38)
RSI_GETPUBLICPAR()	(>>> 7.1.7 "RSI_GETPUBLICPAR()" Seite 38)
RSI_SETPUBLICPAR()	(>>> 7.1.8 "RSI_SETPUBLICPAR()" Seite 39)
RSI_RESET()	(>>> 7.1.9 "RSI_RESET()" Seite 39)
RSI_CHECKID()	(>>> 7.1.10 "RSI_CHECKID()" Seite 39)
RSI_ENABLE()	(>>> 7.1.11 "RSI_ENABLE()" Seite 40)
RSI_DISABLE()	(>>> 7.1.12 "RSI_DISABLE()" Seite 40)

#### 7.1.1 Zeichen und Schriftarten

In den Syntaxbeschreibungen werden folgende Zeichen und Schriftarten verwendet:

Syntax-Element	Darstellung
KRL-Code	<ul style="list-style-type: none"> <li>■ Schriftart Courier</li> <li>■ Großschreibung</li> </ul> Beispiele: GLOBAL; ANIN ON; OFFSET
Elemente, die durch programmspezifische Angaben ersetzt werden müssen	<ul style="list-style-type: none"> <li>■ Kursiv</li> <li>■ Groß-/Kleinschreibung</li> </ul> Beispiele: <i>Strecke</i> ; <i>Zeit</i> ; <i>Format</i>
Optionale Elemente	<ul style="list-style-type: none"> <li>■ In spitzen Klammern</li> </ul> Beispiel: <STEP <i>Schrittweite</i> >
Elemente, die sich gegenseitig ausschließen	<ul style="list-style-type: none"> <li>■ Getrennt durch das Zeichen " "</li> </ul> Beispiel: IN  OUT

#### 7.1.2 RSI\_CREATE()

##### Beschreibung

RSI\_CREATE() erstellt einen RSI Container und lädt den mit RSI Visual konfigurierten Signalfluss in den Container. Über die Container-ID kann auf den erstellten Container zugegriffen werden.

Der mit RSI\_CREATE() erstellte Container ist defaultmäßig aktiviert. Wenn der Container deaktiviert wird (Element *Status:IN*), muss dieser vor dem Ein-

schalten der Signalverarbeitung mit RSI\_ON() wieder aktiviert werden.  
RSI\_ENABLE() aktiviert einen deaktivierten Container.

**Syntax**

*RET*=RSI\_CREATE(*Datei:IN*<,*ContainerID:OUT*><,*Status:IN*>)

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIFILENOTFOUND: Datei mit der Signalkonfiguration nicht gefunden</li> <li>■ RSIINVFILE: Ungültige Datei, z. B. ungültiges Dateiformat oder Fehler in der Konfiguration</li> <li>■ RSINOMEMORY: Kein freier RSI Speicher mehr verfügbar</li> <li>■ RSIINVOBJTYPE: Unbekanntes Objekt im RSI Kontext</li> <li>■ RSIEXTLIBNOTFOUND: Externe RSI Objekt-Bibliothek nicht gefunden</li> <li>■ RSINOTLINKED: RSI Objekt mit fehlendem Eingangssignal</li> <li>■ RSILNKCIRCLE: Fehler in der Signalfluss-Verknüpfung</li> </ul>
<i>Datei:IN</i>	Typ: CHAR-Feld Name der Signalkonfiguration: <Dateiname>.rsi
<i>ContainerID:OUT</i>	Typ: INT ID des RSI Containers
<i>Status:IN</i>	Typ: BOOL TRUE = Aktiviert den RSI Container FALSE = Deaktiviert den RSI Container Default: TRUE

**7.1.3 RSI\_DELETE()****Beschreibung**

RSI\_DELETE() löscht einen RSI Container und die darin enthaltenen RSI Objekte.

**Syntax**

*RET*=RSI\_DELETE(*ContainerID:IN*)

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIINVOBJID: Ungültige Container-ID</li> </ul>
<i>ContainerID:IN</i>	Typ: INT ID des RSI Containers

### 7.1.4 RSI\_ON()

**Beschreibung** Mit RSI\_ON() wird die Signalverarbeitung eingeschaltet sowie der Korrekturmodus und der Sensormodus definiert.

Die Signalverarbeitung wird defaultmäßig im Modus IPO\_FAST ausgeführt. In diesem Fall muss das Referenz-Koordinatensystem für die Sensorkorrektur im RSI Objekt POSCORR konfiguriert werden. Wenn die Signalverarbeitung im IPO-Modus eingeschaltet wird, muss das Referenz-Koordinatensystem mit RSI\_ON() definiert werden.



Wenn die Signale im IPO-Modus verarbeitet werden, kann der Bahnverlauf nur auf der Basis von LIN- und CIRC-Bewegungen korrigiert werden.

#### Syntax

`RET=RSI_ON(<Korrekturmodus:IN><,Sensormodus:IN><,Koordinatensystem:IN>)`

#### Erläuterung der Syntax

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIALREADYON: Signalverarbeitung ist bereits eingeschaltet.</li> </ul>
<i>Korrekturmodus:IN</i>	Typ: ENUM Korrekturmodus: <ul style="list-style-type: none"> <li>■ #ABSOLUTE: Absolutkorrektur</li> <li>■ #RELATIVE: Relativkorrektur</li> </ul> Default: #ABSOLUTE
<i>Sensormodus:IN</i>	Typ: ENUM Modus der Signalverarbeitung: <ul style="list-style-type: none"> <li>■ #IPO_FAST: 4 ms</li> <li>■ #IPO: 12 ms mit Filterung (\$FILTER)</li> </ul> Default: #IPO_FAST
<i>Koordinatensystem:IN</i>	Typ: ENUM Referenz-Koordinatensystem für die Sensorkorrektur (nur relevant, wenn Sensormodus = #IPO) <ul style="list-style-type: none"> <li>■ #BASE</li> <li>■ #TCP</li> <li>■ #TTS</li> <li>■ #WORLD</li> </ul> Default: #BASE

### 7.1.5 RSI\_OFF()

**Beschreibung** RSI\_OFF() schaltet die Signalverarbeitung aus.

**Syntax** `RET=RSI_OFF()`

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSINOTRUNNING: Keine laufende Signalverarbeitung</li> </ul>

**7.1.6 RSI\_MOVECORR()****Beschreibung**

RSI\_MOVECORR() schaltet die sensorgeführte Bewegung ein. Der Roboter wird rein korrekturgesteuert auf der Basis von Sensordaten verfahren, d. h. mit den Korrekturwerten der RSI Objekte POSCORR oder AXISCORR.

Eine sensorgeführte Bewegung kann mithilfe des RSI Objekts STOP abgebrochen werden.

**Syntax**

RSI\_MOVECORR(<Stopp-Modus:IN>)

**Erläuterung der Syntax**

Element	Beschreibung
<i>Stopp-Modus</i>	Typ: ENUM Verhalten nach Abbruch der Bewegung: <ul style="list-style-type: none"> <li>■ #RSIBRAKE: Roboter fährt direkt vom Haltepunkt aus weiter.</li> <li>■ #RSIBRAKERET: Roboter fährt zurück zu dem Punkt auf der Bahn, an dem das Stopp-Signal eingetroffen ist.</li> </ul> Default: RSIBRAKE

**7.1.7 RSI\_GETPUBLICPAR()****Beschreibung**

Mit RSI\_GETPUBLICPAR() kann der Parameterwert eines RSI Objekts gelesen werden. Voraussetzung ist, dass der Objekt-Parameter im RSI Kontext freigeschaltet ist.

**Syntax**

RET=RSI\_GETPUBLICPAR(ContainerID:IN, Objekt:IN, Parameter:IN, Wert:OUT)

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIINVCONT: Ungültige Container-ID</li> <li>■ RSIINPARAMID: Ungültiger RSI Objekt- oder Parametername oder RSI Objekt-Parameter ist nicht freigeschaltet.</li> </ul>
<i>ContainerID:IN</i>	Typ: INT ID des RSI Containers
<i>Objekt:IN</i>	Typ: CHAR-Feld Name des RSI Objekts

Element	Beschreibung
<i>Parameter:IN</i>	Typ: CHAR-Feld Name des RSI Objekt-Parameters
<i>Wert:OUT</i>	Typ: REAL Wert des RSI Objekt-Parameters

### 7.1.8 RSI\_SETPUBLICPAR()

**Beschreibung** Mit RSI\_SETPUBLICPAR() kann dem Parameter eines RSI Objekts einen neuer Wert zugewiesen werden. Voraussetzung ist, dass der Objekt-Parameter im RSI Kontext freigeschaltet ist.

**Syntax** *RET=RSI\_SETPUBLICPAR(ContainerID:IN,Objekt:IN,Parameter:IN,Wert:IN)*

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIINVCONT: Ungültige Container-ID</li> <li>■ RSIINPARAMID: Ungültiger RSI Objekt- oder Parametername oder RSI Objekt-Parameter ist nicht freigeschaltet.</li> <li>■ RSIINPARAM: Ungültiger RSI Objekt-Parameterwert</li> </ul>
<i>ContainerID:IN</i>	Typ: INT ID des RSI Containers
<i>Objekt:IN</i>	Typ: CHAR-Feld Name des RSI Objekts
<i>Parameter:IN</i>	Typ: CHAR-Feld Name des RSI Objekt-Parameters
<i>Wert:IN</i>	Typ: REAL Neuer Wert des RSI Objekt-Parameters

### 7.1.9 RSI\_RESET()

**Beschreibung** RSI\_RESET() löscht die Signalverarbeitung und alle RSI Objekte.

**Syntax** *RET=RSI\_RESET()*

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewert: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> </ul>

### 7.1.10 RSI\_CHECKID()

**Beschreibung** Mit RSI\_CHECKID() kann geprüft werden, ob eine gültige RSI Container-ID verwendet wird.

**Syntax** *RET=RSI\_CHECKID(ContainerID:IN)*

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: BOOL Rückgabewerte: <ul style="list-style-type: none"> <li>■ TRUE = RSI Container zur ID vorhanden</li> <li>■ FALSE = Kein RSI Container zur ID vorhanden</li> </ul>
<i>ContainerID:IN</i>	Typ: INT ID des RSI Containers

**7.1.11 RSI\_ENABLE()**

**Beschreibung** RSI\_ENABLE() aktiviert einen deaktivierten RSI Container.

**Syntax** *RET*=RSI\_ENABLE(*ContainerID:IN*)

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIINVOBJID: Ungültige Container-ID</li> </ul>
<i>ContainerID:IN</i>	Typ: INT ID des RSI Containers

**7.1.12 RSI\_DISABLE()**

**Beschreibung** RSI\_DISABLE() deaktiviert einen RSI Container.

Ein deaktivierter Container muss vor dem Einschalten der Signalverarbeitung mit RSI\_ON() wieder aktiviert werden. RSI\_ENABLE() aktiviert einen deaktivierten Container.

**Syntax** *RET*=RSI\_DISABLE(*ContainerID:IN*)

**Erläuterung der Syntax**

Element	Beschreibung
<i>RET</i>	Typ: INT Rückgabewerte: <ul style="list-style-type: none"> <li>■ RSIOK: Funktion erfolgreich ausgeführt</li> <li>■ RSIINVOBJID: Ungültige Container-ID</li> </ul>
<i>ContainerID:IN</i>	Typ: INT ID des RSI Containers

**7.2 Verhalten der RSI Befehle****7.2.1 Verhalten RSI\_ENABLE() / RSI\_DISABLE()**

**Beschreibung** Mit den Befehlen RSI\_ENABLE(ContId) und RSI\_DISABLE(ContId) kann ein bestimmter RSI Kontext pausieren. Die aktuellen Werte innerhalb des Kontext bleiben nach dem RSI\_DISABLE erhalten. Der RSI Kontext läuft nach dem Pausieren wie gewohnt weiter.



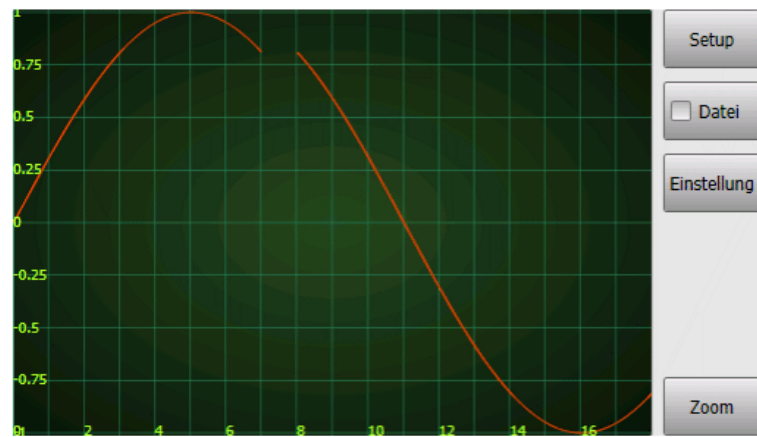


Abb. 7-1: RSI Kontext pausiert

**Beispiel**

Pausieren des RSI Kontexts durch RSI\_DISABLE() beim Generieren eines Sinussignals durch ein SOURCE-Objekt.

```
DEF RSI_DISABLEENABLE()
DECL INT ret,cont
ret=RSI_CREATE("Signals.rsi",cont())
ret=RSI_ON()
wait sec 7
ret=RSI_DISABLE(cont)
wait sec 1
ret=RSI_ENABLE(cont)
wait sec 10
ret=RSI_OFF()
END
```



Sensoren können während der Pause ihre Werte verlieren, dies kann zu einem unerwarteten Verhalten beim Neustart des Kontext mit einem RSI\_ENABLE(contId) führen.

**7.2.2 Verhalten RSI\_ON() / RSI\_OFF()****Beschreibung**

Mit den Befehlen RSI\_OFF und RSI\_ON wird der gesamte RSI Kontext neu gestartet. Mit diesen Befehlen kann kein einzelner RSI Kontext angesprochen werden, sondern nur alle durch RSI\_CREATE erzeugten Kontexte gemeinsam.

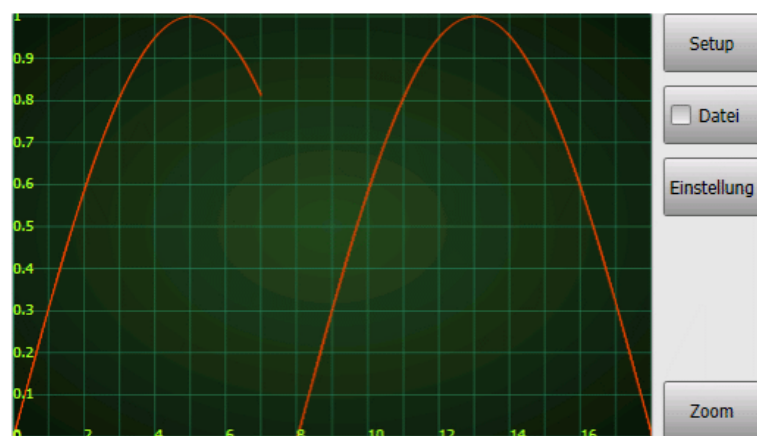


Abb. 7-2: Kontext neu starten

**Beispiel**

Zurücksetzen des RSI Kontext durch RSI\_OFF() beim Generieren eines Sinussignals durch ein SOURCE-Objekt.

```
DEF RSI_ONOFF()
DECL INT ret,cont
ret=RSI_CREATE("Signals.rsi",cont)
ret=RSI_ON()
wait sec 7
ret=RSI_OFF()
wait sec 1
ret=RSI_ON()
Wait sec 10
ret=RSI_OFF()
END
```

### 7.3 Signalverarbeitung programmieren

**Übersicht**

Schritt	Beschreibung
1	Signalfluss mit RSI Visual konfigurieren.
2	Signalfluss-Konfiguration (3 Dateien) auf die Robotersteuerung übertragen.  Zielverzeichnis: C:\KRC\Roboter\Config\User\Common\SensorInterface
3	Signalfluss in KRL-Programm einbinden.  (>>> 7.3.1 "Signalfluss in KRL-Programm einbinden" Seite 42)

#### 7.3.1 Signalfluss in KRL-Programm einbinden

**Beschreibung**

Die Signalverarbeitung muss im KRL-Programm initialisiert, eingeschaltet und wieder ausgeschaltet werden.

Aufbau eines Signalverarbeitungsprogramms:

```
1 DEF signal_processing()
2
3 DECL INT ret
4
5 INI
6 ...
6 ret=RSI_CREATE("test.rsi")
7 ret=RSI_ON()
8 ...
10 movements
11 ...
15 ret=RSI_OFF()
16 ...
20 END
```

Zeile	Beschreibung
3	Deklaration der KRL-Variablen (hier nur Variable "ret" für den Rückgabewert)
6	RSI_CREATE() initialisiert die Signalverarbeitung. Die Signalfluss-Konfiguration wird in einen RSI Container geladen.
7	RSI_ON() schaltet die Signalverarbeitung ein.

Zeile	Beschreibung
10	Bewegungsanweisungen oder RSI_MOVECORR() für eine sensorgeführte Bewegung
15	RSI_OFF() schaltet die Signalverarbeitung aus.

### 7.3.2 Signalfluss-Parameter in KRL ändern

- Beschreibung** Signalfluss-Parameter können nachträglich über folgende Funktionen im KRL-Programm geändert werden:
- RSI\_GETPUBLICPAR(): Liest den konfigurierten Wert eines RSI Objekt-Parameters
  - RSI\_SETPUBLICPAR(): Weist dem RSI Objekt-Parameter einen neuen Wert zu
- Voraussetzung** ■ RSI Objekt-Parameter ist freigeschaltet.
- Beispiel** (>>> 8.1.5 "Beispiel einer sensorgeführten Kreisbewegung" Seite 56)

## 7.4 XML-Datei für Ethernet-Verbindung konfigurieren

**Übersicht** RobotSensorInterface verwendet das XML-Format, um Daten über Ethernet auszutauschen. Für eine Ethernet-Verbindung muss im Verzeichnis C:\KRC\ROBOTER\Config\User\Common\SensorInterface eine Konfigurationsdatei definiert sein.



RSI Visual enthält die Vorlage **RSIEthernet** (Menüfolge **Datei > Neu > Datei...**). Die Vorlage kann verwendet werden, um die Ethernet-Verbindung zu konfigurieren.

Der Name der Konfigurationsdatei wird im ETHERNET-Objekt der Signalfluss-Konfiguration angegeben und beim Initialisieren der Signalverarbeitung im KRL-Programm eingelesen.

```
<ROOT>
  <CONFIG></CONFIG>
  <SEND>
    <ELEMENTS></ELEMENTS>
  </SEND>
  <RECEIVE>
    <ELEMENTS></ELEMENTS>
  </RECEIVE>
</ROOT>
```

Abschnitt	Beschreibung
<CONFIG ... </CONFIG>	Konfiguration der Verbindungsparameter zwischen Sensorsystem und Schnittstelle  (>>> 7.4.1 "XML-Struktur für Verbindungseigenschaften" Seite 44)
<SEND> ... </SEND>	Konfiguration der Sendestruktur  (>>> 7.4.2 "XML-Struktur für den Datenversand" Seite 44)
<RECEIVE> ... </RECEIVE>	Konfiguration der Empfangsstruktur  (>>> 7.4.3 "XML-Struktur für den Datenempfang" Seite 46)

### 7.4.1 XML-Struktur für Verbindungseigenschaften

#### Beschreibung

Elemente der XML-Struktur:

Element	Beschreibung
IP_NUMBER	IP-Adresse des Sensorsystems
PORT	Port-Nummer des Sensorsystems ■ 1 ... 65 534
SENTYPE	Kennung des Sensorsystems (Name frei wählbar) Die Robotersteuerung überprüft die Kennung bei jedem Datenpaket, das sie empfängt.
ONLYSEND	Richtung des Datenaustauschs ■ TRUE = Die Robotersteuerung sendet Daten und erwartet keine Daten vom Sensorsystem zurück. ■ FALSE = Die Robotersteuerung sendet und empfängt Daten. Default: FALSE

#### Beispiel

```
<CONFIG>
  <IP_NUMBER>172.1.10.5</IP_NUMBER>
  <PORT>49152</PORT>
  <SENTYPE>ImFree</SENTYPE>
  <ONLYSEND>FALSE</ONLYSEND>
</CONFIG>
```

### 7.4.2 XML-Struktur für den Datenversand

#### Beschreibung

Hier werden die Signale aus dem RSI Kontext definiert, die an den Eingängen des ETHERNET Objekts eintreffen und an das Sensorsystem gesendet werden.

Das ETHERNET Objekt verfügt außerdem über eine Lesefunktion, mit der Systeminformationen der Robotersteuerung eingelesen und an das Sensorsystem gesendet werden können. Die Lesefunktion wird über Schlüsselwörter aktiviert.

RobotSensorInterface erstellt aus der konfigurierten XML-Struktur automatisch das XML-Dokument, das die Robotersteuerung sendet.

#### Signaleingänge

Definition der Signaleingänge in der XML-Struktur:

Attribut	Beschreibung
TAG	Name des Elements  Hier wird die XML-Struktur für den Datenversand definiert (XML-Schema).  (>>> 7.4.4 "Konfiguration nach XML-Schema" Seite 47)

Attribut	Beschreibung
TYPE	Datentyp des Elements <ul style="list-style-type: none"> <li>■ BOOL</li> <li>■ DOUBLE</li> <li>■ LONG</li> </ul>
INDX	Nummer des ETHERNET Objektingangs <ul style="list-style-type: none"> <li>■ 1 ... 64</li> </ul> <p><b>Hinweis:</b> Die Objektingänge müssen fortlaufend nummeriert sein.</p>

### Beispiel Signaleingänge

- Konfigurierte XML-Struktur für den Datenversand:

```
<SEND>
<ELEMENTS>
  <ELEMENT TAG="Out.01" TYPE="BOOL" INDX="1" />
  <ELEMENT TAG="Out.02" TYPE="BOOL" INDX="2" />
  <ELEMENT TAG="Out.03" TYPE="BOOL" INDX="3" />
  <ELEMENT TAG="Out.04" TYPE="BOOL" INDX="4" />
  <ELEMENT TAG="Out.05" TYPE="BOOL" INDX="5" />
  <ELEMENT TAG="FTC.Fx" TYPE="DOUBLE" INDX="6" />
  <ELEMENT TAG="FTC.Fy" TYPE="DOUBLE" INDX="7" />
  <ELEMENT TAG="FTC.Fz" TYPE="DOUBLE" INDX="8" />
  <ELEMENT TAG="FTC.Mx" TYPE="DOUBLE" INDX="9" />
  <ELEMENT TAG="FTC.My" TYPE="DOUBLE" INDX="10" />
  <ELEMENT TAG="FTC.Mz" TYPE="DOUBLE" INDX="11" />
  <ELEMENT TAG="Override" TYPE="LONG" INDX="12" />
</ELEMENTS>
</SEND>
```

- Von der Robotersteuerung gesendetes XML-Dokument:

```
<Rob TYPE="KUKA">
  <Out 01="0" 02="1" 03="1" 04="0" 05="0" />
  <FTC Fx="1.234" Fy="54.75" Fz="345.7
    Mx="2346.6" My="12.0" Mz="3546" />
  <Override>90</Override>
  <IPOC>123645634563</IPOC>
</Rob>
```



Das Schlüsselwort IPOC sendet einen Zeitstempel und wird automatisch erstellt.

### Lesefunktion

Aktivieren der Lesefunktion in der XML-Struktur:

Attribut	Beschreibung
TAG	Name des Elements  Ein Schlüsselwort gibt an, welche Systeminformation eingelesen wird.  (>>> 7.4.5 "Schlüsselwörter - Daten lesen" Seite 48)
TYPE	Datentyp des Elements <ul style="list-style-type: none"> <li>■ DOUBLE</li> <li>■ LONG</li> </ul>
INDX	Schlüsselwort zum Einlesen der Systeminformation <ul style="list-style-type: none"> <li>■ INTERNAL</li> </ul>

## Beispiel Lesefunktion

(>>> "Beispiel Lesefunktion" Seite 49)

### 7.4.3 XML-Struktur für den Datenempfang

#### Beschreibung

Hier werden die Signale definiert, die an den Ausgängen des ETHERNET Objekts vom Sensorsystem empfangen und im RSI Kontext an die Robotersteuerung weitergeleitet werden.

Das ETHERNET Objekt verfügt außerdem über eine Schreibfunktion, mit der Informationen in die Robotersteuerung geschrieben oder Meldungen auf der smartHMI ausgegeben werden können. Die Schreibfunktion wird über Schlüsselwörter aktiviert.

RobotSensorInterface erstellt aus der konfigurierten XML-Struktur automatisch das XML-Dokument, das die Robotersteuerung erwartet.

#### Signalausgänge

Definition der Signalausgänge in der XML-Struktur:

Attribut	Beschreibung
TAG	Name des Elements  Hier wird die XML-Struktur für den Datenempfang definiert (XML-Schema).  (>>> 7.4.4 "Konfiguration nach XML-Schema" Seite 47)
TYPE	Datentyp des Elements  <ul style="list-style-type: none"> <li>■ BOOL</li> <li>■ DOUBLE</li> <li>■ LONG</li> </ul>
INDX	Nummer des ETHERNET Objektausgangs  <ul style="list-style-type: none"> <li>■ 1 ... 64</li> </ul> <b>Hinweis:</b> Die Objektausgänge müssen fortlaufend nummeriert sein.
HOLDON	Verhalten des Objektausgangs bei zu spät eintreffenden Datenpaketen  <ul style="list-style-type: none"> <li>■ 0: Der Ausgang wird zurückgesetzt.</li> <li>■ 1: Der letzte gültige Wert, der eingetroffen ist, bleibt am Ausgang stehen.</li> </ul>

#### Beispiel Signalausgänge

##### ■ Konfigurierte XML-Struktur für den Datenempfang:

```
<RECEIVE>
<ELEMENTS>
  <ELEMENT TAG="RKorr.X" TYPE="DOUBLE" INDX="1" HOLDON="1" />
  <ELEMENT TAG="RKorr.Y" TYPE="DOUBLE" INDX="2" HOLDON="1" />
  <ELEMENT TAG="RKorr.Z" TYPE="DOUBLE" INDX="3" HOLDON="1" />
  <ELEMENT TAG="RKorr.A" TYPE="DOUBLE" INDX="4" HOLDON="1" />
  <ELEMENT TAG="RKorr.B" TYPE="DOUBLE" INDX="5" HOLDON="1" />
  <ELEMENT TAG="RKorr.C" TYPE="DOUBLE" INDX="6" HOLDON="1" />
  <ELEMENT TAG="AK.A1" TYPE="DOUBLE" INDX="7" HOLDON="0" />
  <ELEMENT TAG="AK.A2" TYPE="DOUBLE" INDX="8" HOLDON="0" />
  <ELEMENT TAG="AK.A3" TYPE="DOUBLE" INDX="9" HOLDON="0" />
  <ELEMENT TAG="AK.A4" TYPE="DOUBLE" INDX="10" HOLDON="0" />
  <ELEMENT TAG="AK.A5" TYPE="DOUBLE" INDX="11" HOLDON="0" />
  <ELEMENT TAG="AK.A6" TYPE="DOUBLE" INDX="12" HOLDON="0" />
  <ELEMENT TAG="EK.E1" TYPE="DOUBLE" INDX="13" HOLDON="0" />
  <ELEMENT TAG="EK.E2" TYPE="DOUBLE" INDX="14" HOLDON="0" />
```

```

<ELEMENT TAG="EK.E3" TYPE="DOUBLE" INDX="15" HOLDON="0" />
<ELEMENT TAG="EK.E4" TYPE="DOUBLE" INDX="16" HOLDON="0" />
<ELEMENT TAG="EK.E5" TYPE="DOUBLE" INDX="17" HOLDON="0" />
<ELEMENT TAG="EK.E6" TYPE="DOUBLE" INDX="18" HOLDON="0" />
<ELEMENT TAG="DiO" TYPE="LONG" INDX="19" HOLDON="1" />
</ELEMENTS>
</RECEIVE>

```

■ Vom Sensorsystem empfangenes XML-Dokument:

```

<Sen Type="ImFree">
  <RKorr X="4" Y="7" Z="32" A="6" B="" C="6" />
  <AK A1="2" A2="54" A3="35" A4="76" A5="567" A6="785" />
  <EK E1="67" E2="67" E3="678" E4="3" E5="3" E6="7" />
  <DiO>123</DiO>
  <IPOC>123645634563</IPOC>
</Sen>

```



Der Zeitstempel mit dem Schlüsselwort IPOC wird überprüft. Das Datenpaket ist nur dann gültig, wenn der Zeitstempel mit dem zuvor gesendeten Zeitstempel übereinstimmt.

## Schreibfunktion

Aktivieren der Schreibfunktion in der XML-Struktur:

Attribut	Beschreibung
TAG	Name des Elements Ein Schlüsselwort gibt an, welche Information in die Robotersteuerung geschrieben wird oder ob eine Meldung auf der smartHMI ausgegeben wird.
TYPE	Datentyp des Elements <ul style="list-style-type: none"> <li>■ DOUBLE</li> <li>■ STRING</li> </ul>
INDX	Schlüsselwort zum Schreiben der Information <ul style="list-style-type: none"> <li>■ INTERNAL</li> </ul>
HOLDON	Verhalten des Objektausgangs bei zu spät eintreffenden Datenpaketen <ul style="list-style-type: none"> <li>■ 0: Der Ausgang wird zurückgesetzt.</li> <li>■ 1: Der letzte gültige Wert, der eingetroffen ist, bleibt am Ausgang stehen.</li> </ul>

## Beispiel Schreibfunktion

### 7.4.4 Konfiguration nach XML-Schema

#### Beschreibung

RobotSensorInterface erstellt aus der konfigurierten XML-Struktur automatisch die XML-Dokumente für den Datenaustausch.

Nach XML-Schema sind folgende Schreibweisen zu unterscheiden:

- Element-Schreibweise
- Attribut-Schreibweise

#### Element-Schreibweise

- TAGs in der konfigurierten XML-Struktur:

```

...
<ELEMENTS>

```

```

<ELEMENT TAG="Out1" ... />
<ELEMENT TAG="Out2" ... />
<ELEMENT TAG="Out3" ... />
</ELEMENTS>
...

```

■ TAGs im erstellten XML-Dokument:

```

...
<Out1>...</Out1>
<Out2>...</Out2>
<Out3>...</Out3>
...

```

**Attribut-Schreibweise**

■ TAGs in der konfigurierten XML-Struktur:

```

...
<ELEMENTS>
  <ELEMENT TAG="Out.01" ... />
  <ELEMENT TAG="Out.02" ... />
  <ELEMENT TAG="Out.03" ... />
</ELEMENTS>
...

```

■ TAG mit Attributen im erstellten XML-Dokument:

```

...
<Out 01="..." 02="..." 03="..." />
...

```

#### 7.4.5 Schlüsselwörter - Daten lesen

Schlüsselwörter sind Buchstabenfolgen mit einer fest zugeordneten Bedeutung. Sie dürfen in der XML-Struktur nicht anders als in dieser Bedeutung verwendet werden. Die Groß- und Kleinschreibung ist nicht relevant. Ein Schlüsselwort gilt in jeder Schreibweise als Schlüsselwort.

**Schlüsselwörter** Folgende Informationen der Robotersteuerung können über Schlüsselwörter im Attribut TAG eingelesen werden:

Information	Schlüsselwort	Datentyp
Kartesische Istposition	DEF_RIst	DOUBLE
Kartesische Sollposition	DEF_RSol	DOUBLE
Achsspezifische Istposition der Roboterachsen A1 ... A6	DEF_AIPos	DOUBLE
Achsspezifische Sollposition der Roboterachsen A1 ... A6	DEF_ASPos	DOUBLE
Achsspezifische Istposition der Zusatzachsen E1 ... E6	DEF_EIPos	DOUBLE
Achsspezifische Sollposition der Zusatzachsen E1 ... E6	DEF_ESPos	DOUBLE
Motorströme der Roboterachsen A1 ... A6	DEF_MACur	DOUBLE
Motorströme der Zusatzachsen E1 ... E6	DEF_MECur	DOUBLE
Anzahl der zu spät eingetroffenen Datenpakete	DEF_Delay	LONG



Information	Schlüsselwort	Datentyp
Technologie-Parameter im Hauptlauf (Funktionsgeneratoren 1 ... 6)	DEF_Tech.C1 ... DEF_Tech.C6	DOUBLE
Technologie-Parameter im Vorlauf (Funktionsgeneratoren 1 ... 6)	DEF_Tech.T1 ... DEF_Tech.T6	DOUBLE

### Beispiel Lesefunktion

- Konfigurierte XML-Struktur für den Datenversand:

```
<SEND>
<ELEMENTS>
  <ELEMENT TAG="DEF_RIst" TYPE="DOUBLE" INDX="INTERNAL" />
  <ELEMENT TAG="DEF_AIPos" TYPE="DOUBLE" INDX="INTERNAL" />
  <ELEMENT TAG="DEF_MACur" TYPE="DOUBLE" INDX="INTERNAL" />
  <ELEMENT TAG="DEF_Delay" TYPE="LONG" INDX="INTERNAL" />
  <ELEMENT TAG="DEF_Tech.C1" TYPE="DOUBLE" INDX="INTERNAL" />
</ELEMENTS>
</SEND>
```

- Von der Robotersteuerung gesendetes XML-Dokument:

```
<Rob TYPE="KUKA">
  <RIst X="0.0" Y="0.0" Z="0.0" A="0.0" B="0.0" C="0.0" />
  <AIPos A1="0.0" A2="0.0" A3="0.0" A4="0.0" A5="0.0" A6="0.0" />
  <MACur A1="1.0" A2="1.0" A3="1.0" A4="1.0" A5="1.0" A6="1.0" />
  <Delay D="" />
  <Tech T11="0.0" T12="0.0" T13="0.0" T14="0.0" T15="0.0" T16="0.0"
    T17="0.0" T18="0.0" T19="0.0" T110="0.0" />
  <IPOC>123645634563</IPOC>
</Rob>
```



Das Schlüsselwort IPOC sendet einen Zeitstempel und wird automatisch erstellt.

## 7.4.6 Schlüsselwörter - Daten schreiben

Schlüsselwörter sind Buchstabenfolgen mit einer fest zugeordneten Bedeutung. Sie dürfen in der XML-Struktur nicht anders als in dieser Bedeutung verwendet werden. Die Groß- und Kleinschreibung ist nicht relevant. Ein Schlüsselwort gilt in jeder Schreibweise als Schlüsselwort.

### Schlüsselwörter

Folgende Informationen können über Schlüsselwörter im Attribut TAG in die Robotersteuerung geschrieben werden:

Information	Schlüsselwort	Datentyp
Technologie-Parameter im Hauptlauf (Funktionsgeneratoren 1 ... 6)	DEF_Tech.C1 ... DEF_Tech.C6	DOUBLE
Technologie-Parameter im Vorlauf (Funktionsgeneratoren 1 ... 6)	DEF_Tech.T1 ... DEF_Tech.T6	DOUBLE

Schlüsselwort im Attribut TAG um Meldungen auf der smartHMI auszugeben:

Information	Schlüsselwort	Datentyp
Hinweis- oder Fehlermeldung	DEF_EStr	STRING

### Meldungstypen

Im XML-Dokument, das vom Sensorsystem beschrieben und gesendet wird, können folgende Meldungstypen auftreten:

- **<EStr> xxx </EStr>**: Hinweismeldung
- **<EStr>Error: xxx </EStr>**: Quittiermeldung (Stopp des Roboters)

### Beispiel Schreibfunktion

- **<EStr/>**: Keine Meldung bei einem leeren Tag
- Konfigurierte XML-Struktur für den Datenempfang:

```
<RECEIVE>
<ELEMENTS>
  <ELEMENT TAG="DEF_EStr" TYPE="STRING" INDX="INTERNAL" />
  <ELEMENT TAG="DEF_Tech.T2" TYPE="DOUBLE" INDX="INTERNAL" HOLDON="0" />
</ELEMENTS>
</RECEIVE>
```

- Vom Robotersystem empfangenes XML-Dokument:

```
<Sen Type="ImFree">
  <EStr>Message!</EStr>
  <Tech T21="0.0" T22="0.0" T23="0.0" T24="0.0" T25="0.0" T26="0.0"
    T27="0.0" T28="0.0" T29="0.0" T210="0.0" />
  <IPOC>123645634563</IPOC>
</Sen>
```



Der Zeitstempel mit dem Schlüsselwort IPOC wird überprüft. Das Datenpaket ist nur dann gültig, wenn der Zeitstempel mit dem zuvor gesendeten Zeitstempel übereinstimmt.

## 8 Beispiele

### 8.1 Beispielkonfigurationen und -programme



RSI-, DIAGRAM- und XML-Datei sind eine Einheit und müssen zusammen auf die Robotersteuerung übertragen werden.

Zielverzeichnis:

C:\KRC\Roboter\Config\User\Common\SensorInterface

#### Übersicht

RobotSensorInterface beinhaltet eine Beispielapplikation, mit der eine Ethernet-Kommunikation zwischen einem Server-Programm und der Robotersteuerung hergestellt und getestet werden kann. Die Beispielapplikation sowie weitere Beispiel-Konfigurationen und -Programme befinden sich Verzeichnis DOC\Examples der Software.

Die Beispielapplikation für die Ethernet-Kommunikation besteht aus folgenden Komponenten:

Komponenten	Ordner
Server-Programm TestServer.exe	...Ethernet\Server
Beispielprogramm in KRL: ■ RSI_Ethernet.src	...Ethernet
Beispielkonfiguration für den Signalfluss: ■ RSI_Ethernet.rsi ■ RSI_Ethernet.rsi.xml ■ RSI_Ethernet.rsi.diagram XML-Datei für die Ethernet-Verbindung: ■ RSI_EthernetConfig.xml	...Ethernet\Config

Weitere Beispielkonfigurationen und -programme:

Komponenten	Ordner
Beispielprogramm in KRL: ■ RSI_CircleCorr.src	...CircleCorr
Beispielkonfiguration für den Signalfluss: ■ RSI_CircleCorr.rsi ■ RSI_CircleCorr.rsi.xml ■ RSI_CircleCorr.rsi.diagram	...CircleCorr\Config
Beispielprogramm in KRL: ■ RSI_DistanceCtrl.src	...DistanceCtrl
Beispielkonfiguration für den Signalfluss: ■ RSI_DistanceCtrl.rsi ■ RSI_DistanceCtrl.rsi.xml ■ RSI_DistanceCtrl.rsi.diagram	...DistanceCtrl\Config
Beispielprogramm in KRL: ■ RSI_SigTransformation.src	...Transformations
Beispielkonfiguration für den Signalfluss: ■ RSI_SigTransformation.rsi ■ RSI_SigTransformation.rsi.xml ■ RSI_SigTransformation.rsi.diagram	...Transformations\Config

### 8.1.1 Beispielapplikation implementieren

#### Voraussetzung

Externes System:

- Windows-Betriebssystem mit installiertem .NET-Framework 3.5 oder höher

Robotersteuerung:

- Benutzergruppe Experte
- Betriebsart T1 oder T2

#### Vorgehensweise

1. Server-Programm auf externes System kopieren.
2. KRL-Programme in das Verzeichnis C:\KRC\ROBOTER\KRC\R1\Program der Robotersteuerung kopieren.
3. Die Beispielkonfigurationen und die XML-Datei für die Ethernet-Verbindung in das Verzeichnis C:\KRC\ROBOTER\Config\User\Common\SensorInterface der Robotersteuerung kopieren.
4. Server-Programm auf dem externen System starten.
5. Menü-Button drücken. Das Fenster **Server Properties** öffnet sich.
6. Unter Available Network Interfaces werden die verfügbaren IP-Adressen des Server-PCs angezeigt.
7. Die IP-Adresse der Verbindung zwischen Roboter und PC, in der XML-Datei für die Ethernet-Verbindung einstellen.

### 8.1.2 Bedienoberfläche Server-Programm

Das Server-Programm ermöglicht es, die Verbindung zwischen einem externen System und der Robotersteuerung zu testen, indem eine stabile Kommunikation zur Robotersteuerung hergestellt wird.

Dazu werden die empfangenen Daten ausgewertet und der aktuelle Zeitstempel des Paketes in das zu sendende XML-Dokument kopiert. Das XML-Dokument kann mit Korrekturdaten oder mit Nullwerten gesendet werden.

Das Server-Programm enthält folgende Funktionalitäten:

- Senden und Empfangen von Daten im Sensortakt
- Freie Bewegungskorrektur, kartesisch über Bedienelemente
- Anzeige der empfangenen Daten
- Anzeige der gesendeten Daten



Das Testserver-Programm und das Betriebssystem Windows sind nicht echtzeitfähig. Es können keine Rückschlüsse auf das zeitliche Verhalten oder die Stabilität eines Gesamtsystems gezogen werden.

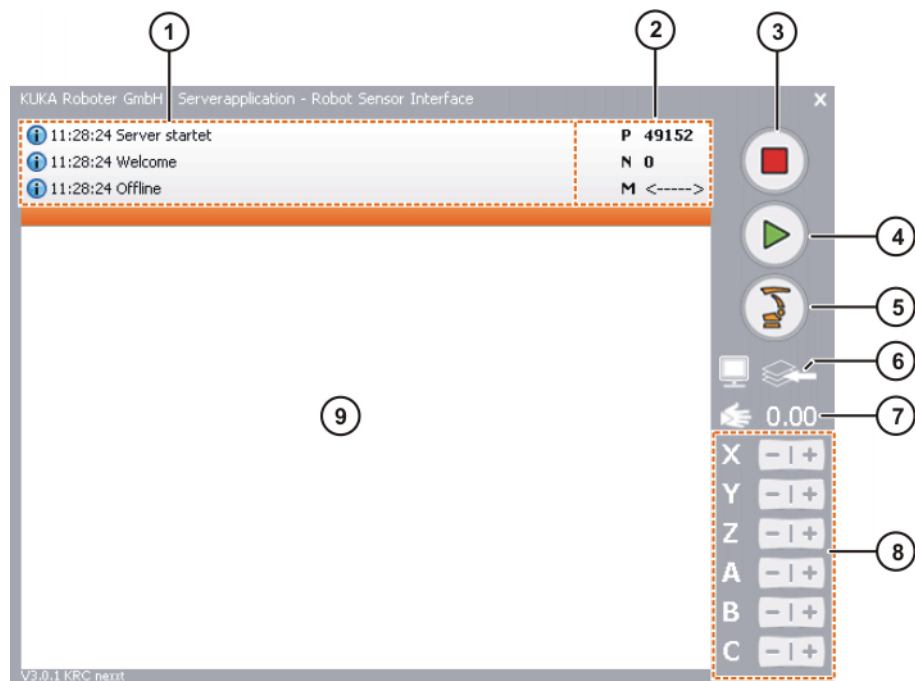


Abb. 8-1: Serverprogramm

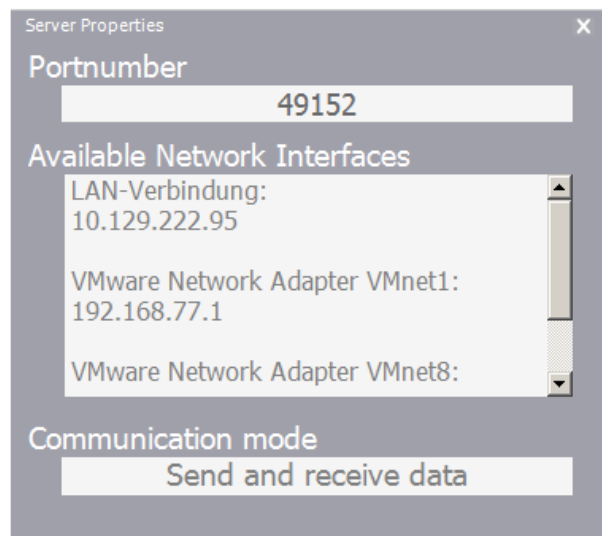
Pos.	Beschreibung
1	Meldungsfenster
2	Anzeige der eingestellten Kommunikationsparameter <ul style="list-style-type: none"> <li>■ <b>P</b>: Port-Nummer</li> <li>■ <b>N</b>: Netzwerkkarten-Index</li> <li>■ <b>M</b>: Kommunikationsmodus <ul style="list-style-type: none"> <li>■ &lt;----&gt;: Der Server kann Daten empfangen und senden.</li> <li>■ &lt;-----: Der Server kann nur Daten empfangen.</li> </ul> </li> </ul>
3	Stopp-Button Die Kommunikation mit der Robotersteuerung wird beendet und der Server wird zurückgesetzt.
4	Start-Button Der Datenaustausch zwischen Server-Programm und Robotersteuerung wird ausgewertet. Die erste eingehende Verbindungsanfrage wird gebunden und als Kommunikationsadapter benutzt.
5	Menü-Button zum Einstellen der Kommunikationsparameter
6	Anzeigeoptionen <ul style="list-style-type: none"> <li>■ Pfeil zeigt nach links: Die empfangenen Daten werden angezeigt. (Default)</li> <li>■ Pfeil zeigt nach rechts: Die gesendeten Daten werden angezeigt.</li> </ul>
7	Hand-Symbol Über einen Regler kann die Schrittweite für die Bewegungskorrektur pro Sensortakt eingestellt werden. <ul style="list-style-type: none"> <li>■ <b>0.00 ... 3.33</b></li> </ul>

Pos.	Beschreibung
8	Buttons für die inkrementelle Bewegungskorrektur pro Sensortakt Die Schrittweite wird über das Hand-Symbol eingestellt.
9	Anzeigefenster Je nach eingestellter Anzeigeeoption werden die gesendeten oder die empfangenen Daten angezeigt. Die angezeigten Daten werden im Sensortakt aktualisiert.

### 8.1.3 Kommunikationsparameter im Server-Programm einstellen

- Vorgehensweise**
1. Im Server-Programm auf den Menü-Button klicken.  
Das Fenster **Server Properties** öffnet sich.
  2. Kommunikationsparameter einstellen.
  3. Fenster schließen.

#### Beschreibung



**Abb. 8-2: Fenster Server Properties**

Element	Beschreibung
Portnumber	Port-Nummer der Socket-Verbindung eingeben.  An diesem Port erwartet das externe System die Verbindungsanfrage der Robotersteuerung. Es muss eine freie Nummer gewählt werden, die nicht als Standarddienst belegt ist.  Default-Wert: <b>49152</b>
Available Network Interfaces	Zeigt die verfügbaren IP-Adressen an, die auf dem verwendeten PC definiert sind.
Communication mode	Kommunikationsmodus auswählen. <ul style="list-style-type: none"> <li>■ <b>Send and receive data:</b> Der Server kann Daten empfangen und senden.</li> <li>■ <b>Only receive data:</b> Der Server kann nur Daten empfangen.</li> </ul> Default-Wert: <b>Send and receive data</b>

### 8.1.4 Beispiel einer kartesischen Korrektur über Ethernet

Die Robotersteuerung empfängt kartesische Korrekturdaten von einem Sensor und schickt diese an den Roboter. Der Roboter wird rein korrekturgesteuert auf Basis von relativen Korrekturwerten verfahren. Referenz-Koordinatensystem für die Korrekturaufschaltung ist das BASE-Koordinatensystem.

#### Programm

```

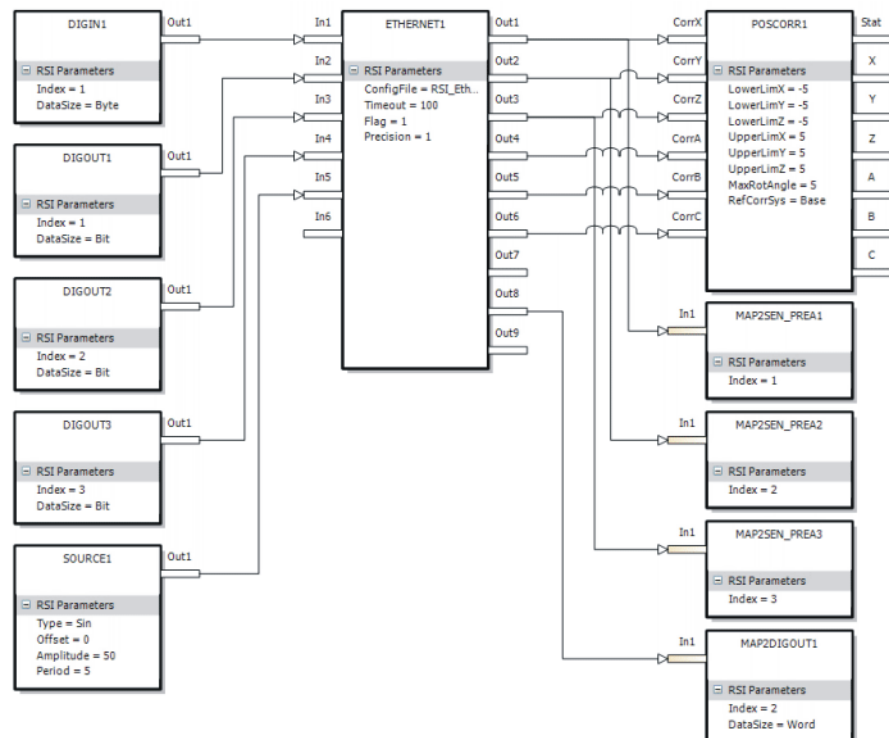
1  DEF RSI_Ethernet( )
2  ; =====
3  ;
4  ; RSI EXAMPLE: ETHERNET communication
5  ; Realtime UDP data exchange with server application
6  ;
7  ; =====
8
9  ; Declaration of KRL variables
10 DECL INT ret; Return value for RSI commands
11 DECL INT CONTID; ContainerID
12
13 INI
14
15 ; Move to start position
16 PTP {A1 0, A2 -90, A3 90, A4 0, A5 90, A6 0}
17
18 ; Create RSI Context
19 ret = RSI_CREATE("RSI_Ethernet.rsi",CONTID,TRUE)
20 IF (ret <> RSIOK) THEN
21   HALT
22 ENDIF
23
24 ; Start RSI execution
25 ret = RSI_ON(#RELATIVE)
26 IF (ret <> RSIOK) THEN
27   HALT
28 ENDIF
29
30 ; Sensor guided movement
31 RSI_MOVECORR()
32
33 ; Turn off RSI
34 ret = RSI_OFF()
35 IF (ret <> RSIOK) THEN
36   HALT
37 ENDIF
38
39 PTP {A1 0, A2 -90, A3 90, A4 0, A5 90, A6 0}
40
41 END

```

Zeile	Beschreibung
16	Startposition der sensorgeführten Bewegung
19	RSI_CREATE() lädt die Signalfuss-Konfiguration in einen RSI Container.
25	RSI_ON() schaltet die Signalverarbeitung ein. ■ Korrekturmodus: Relativkorrektur
31	RSI_MOVECORR() schaltet die sensorgeführte Bewegung ein.

Zeile	Beschreibung
34	RSI_OFF() schaltet die Signalverarbeitung aus.
39	Zurück zur Startposition

### Signalfluss-Konfiguration



**Abb. 8-3: Signalfluss - Kartesische Korrektur über Ethernet**

RSI Objekt	Beschreibung
DIGIN1	Liest über 8 digitale Eingänge die Sensordaten ein und übergibt diese an die Ethernet-Schnittstelle (Eingang 1).
DIGOUT1 ... DIGOUT3	Liest über 3 digitale Ausgänge Roboterdaten ein und übergibt diese an die Ethernet-Schnittstelle (Eingang 2 ... 4).
SOURCE1	Liefert ein sinusförmiges Signal mit der Amplitude 50, periodisch alle 5 s.
ETHERNET1	Sendet die über Eingang 2 ... 5 eingehenden Signale an das Sensorsystem und erhält Sensordaten über Eingang 1 zurück. Die Sensordaten stehen an Ausgang 1... 6 zur weiteren Verarbeitung zur Verfügung.
POSCORR1	Liest die Sensordaten ein, die an Ausgang 1... 6 der Ethernet-Schnittstelle zur Verfügung stehen und ermittelt die kartesischen Korrekturdaten.
MAP2SEN_PREA1 ... MAP2SEN_PREA3	Schreiben die kartesischen Korrekturdaten in die Systemvariable \$SEN_PREA.
MAP2DIGOUT1	Greift auf die verarbeiteten Signale zu und setzt 16 digitale Ausgänge.

#### 8.1.5 Beispiel einer sensorgeführten Kreisbewegung

Es wird eine sensorgeführte Kreisbewegung konfiguriert. Dazu wird ein sinusförmiges Signal erzeugt, das als Sinus in Z-Richtung und noch einmal zeitverzögert als Sinus in Y-Richtung in das Korrekturobjekt POSCORR eingelesen wird. Nach dem ersten Durchlauf der Signalverarbeitung wird die Amplitude des Signals im KRL-Programm nachträglich geändert. Bei erneutem Start der



Signalverarbeitung mit halbiert Amplitude, erhält man eine kleinere Kreisbewegung.

Der Roboter wird rein korrekturgesteuert auf Basis der absoluten Korrekturwerte in Y- und Z-Richtung verfahren. Referenz-Koordinatensystem für die Korrekturaufschaltung ist das BASE-Koordinatensystem. Über einen Timer wird die sensorgeführte Bewegung nach einer definierten Zeit abgebrochen.

## Programm

```

1  DEF RSI_CircleCorr( )
2  ; =====
3  ;
4  ; RSI EXAMPLE: Lissajous circle
5  ; Create a circle movement with two sine corrections
6  ;
7  ; =====
8
9  ; Declaration of KRL variables
10 DECL INT ret; Return value for RSI commands
11 DECL INT CONTID; ContainerID
12
13 INI
14
15 ; Move to start position
16 PTP {A1 0, A2 -90, A3 90, A4 0, A5 90, A6 0}
17
18 ; Base in actual position
19 $BASE.X=$POS_ACT.X
20 $BASE.Y=$POS_ACT.Y
21 $BASE.Z=$POS_ACT.z
22
23 ; Create RSI Context
24 ret=RSI_CREATE("RSI_CircleCorr.rsi",CONTID)
25 IF (ret <> RSIOK) THEN
26   HALT
27 ENDIF
28
29 ; Start RSI execution
30 ret=RSI_ON(#ABSOLUTE)
31 IF (ret <> RSIOK) THEN
32   HALT
33 ENDIF
34
35 ; Sensor guided movement
36 RSI_MOVECORR()
37
38 ; Turn off RSI
39 ret=RSI_OFF()
40 IF (ret <> RSIOK) THEN
41   HALT
42 ENDIF
43
44 ; Modify RSI parameter
45 ret=RSI_GETPUBLICPAR(CONTID,"SOURCE1","Amplitude", fVar)
46 ...
49 ret=RSI_SETPUBLICPAR(CONTID,"SOURCE1","Amplitude", fVar/2)
50 ...
54 ; Start RSI execution
55 ret=RSI_ON(#ABSOLUTE)
56 ...
59
60 ; Sensor guided movement

```

```

61 RSI_MOVECORR()
62
63 ; Turn off RSI
64 ret=RSI_OFF()
...
68
69 PTP {A1 0, A2 -90, A3 90, A4 0, A5 90, A6 0}
70
71 END

```

Zeile	Beschreibung
16	Startpunkt der sensorgeführten Bewegung
19 ... 21	Aktuelle Roboterposition bezogen auf die Basis
24	RSI_CREATE() lädt die Signalfluss-Konfiguration in einen RSI Container.
30	RSI_ON() schaltet die Signalverarbeitung ein. ■ Korrekturmodus: Absolutkorrektur
36	RSI_MOVECORR() schaltet die sensorgeführte Bewegung ein.
39	RSI_OFF() schaltet die Signalverarbeitung aus.
45	RSI_GETPUBLICPAR() liest die aktuell eingestellte Amplitude des Signals (SOURCE1).
49	RSI_SETPUBLICPAR() weist der Amplitude des Signals (SOURCE1) einen neuen Wert zu. Die Amplitude wird halbiert.
55	RSI_ON() schaltet die Signalverarbeitung ein. ■ Korrekturmodus: Absolutkorrektur
61	RSI_MOVECORR() schaltet die sensorgeführte Bewegung ein.
64	RSI_OFF() schaltet die Signalverarbeitung aus.

## Signalfluss- Konfiguration

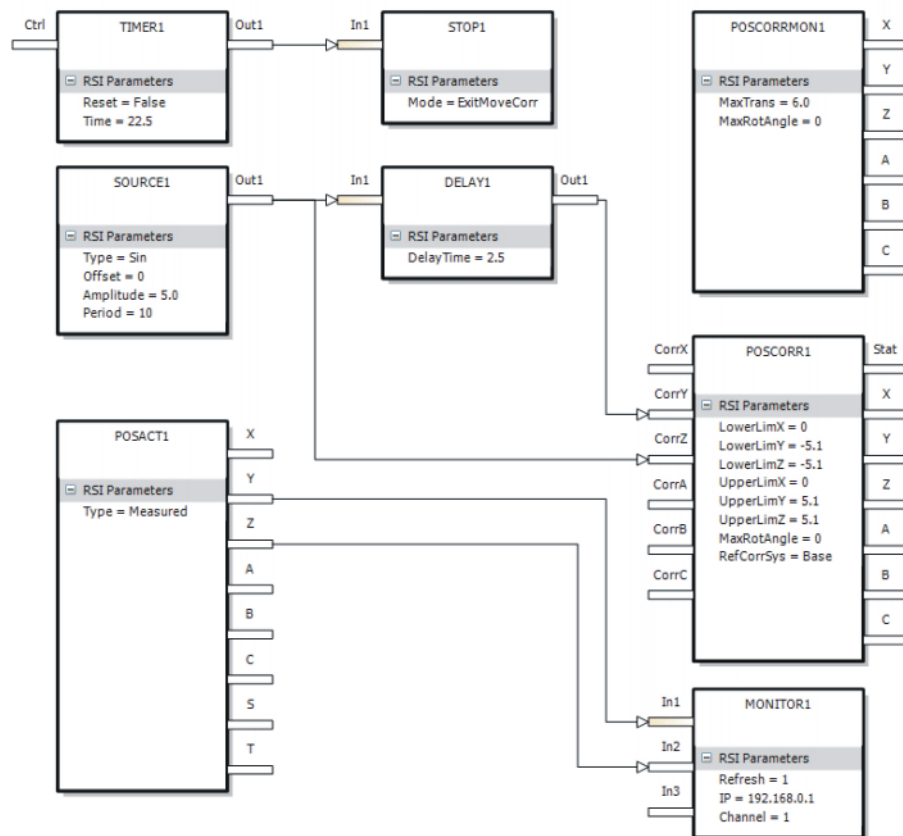
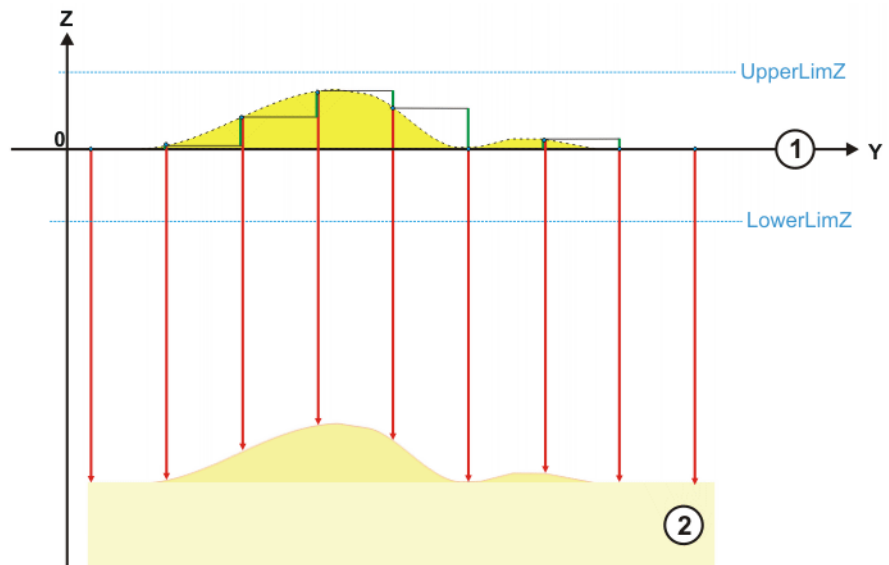


Abb. 8-4: Signalfluss - Sensorgeführte Kreisbewegung

RSI Objekt	Beschreibung
TIMER1	Nach Ablauf der im Timer eingestellten Zeit wird die sensorgeführte Bewegung abgebrochen.
STOP1	
POSCORRMON1	Begrenzt die maximale kartesische Gesamtkorrektur. <ul style="list-style-type: none"> <li>Maximale translatorische Auslenkung in X, Y, Z: 6 mm</li> </ul>
SOURCE1	Liefert ein sinusförmiges Signal mit der Amplitude 5.0, periodisch alle 10 s.
DELAY1	Das Signal wird um 2.5 s verzögert.
POSCORR1	Liest den Sinus-Korrekturwert in Z-Richtung und zeitverzögert den Sinus-Korrekturwert in Y-Richtung ein.
POSACT1	Liest die kartesische Istposition des Roboters in Y- und Z-Richtung ein. <ul style="list-style-type: none"> <li>Referenz-Koordinatensystem für die Korrekturaufschaltung: BASE</li> </ul>
MONITOR1	Folgende Signale sind mit dem MONITOR-Objekt verknüpft und können mit RSI Monitor auf der Robotersteuerung angezeigt werden: <ul style="list-style-type: none"> <li>Kartesische Istposition des Roboters in Y- und Z-Richtung [mm]</li> </ul>

### 8.1.6 Beispiel einer Bahnkorrektur zur Abstandssteuerung

Es soll ein definierter Abstand zu einem Bauteil eingehalten werden. Bei eingeschalteter Signalverarbeitung misst ein Sensor den Abstand zum Bauteil und verfährt mit einer LIN-Bewegung 100 mm in Y-Richtung. Parallel wird ein relativer Korrekturwert ermittelt und der Bahnverlauf der LIN-Bewegung in Z-Richtung korrigiert.



**Abb. 8-5: Bahnkorrektur zur Abstandssteuerung**

1 Bauteil

2 Programmierte Bahn

#### Programm

```

1  DEF RSI_DistanceCtrl( )
2  ; =====
3  ;
4  ; RSI EXAMPLE: Distance Ctrl
5  ; Move on a LIN path with superimposed corrections
6  ; Deviation from programmed path is controlled with
7  ; a analog input $ANIN[1]
8  ;
9  ; =====
10
11 ; Declaration of KRL variables
12 DECL INT ret; Return value for RSI commands
13 DECL INT CONTID; ContainerID
14
15 INI
16
17 ; Move to start position
18 PTP {A1 0, A2 -90, A3 90, A4 0, A5 90, A6 0}
19 $BASE=$POS_ACT
20
21 ; Create signal processing
22 ret=RSI_CREATE("RSI_DistanceCtrl.rsi")
23 IF (ret <> RSIOK) THEN
24   HALT
25 ENDIF
26
27 ; Start signal processing in relative correction mode
28 ret=RSI_ON(#RELATIVE)
29 IF (ret <> RSIOK) THEN
30   HALT
31 ENDIF
32
33 LIN_REL {Y 100}
34
35 ; Turn off RSI
36 ret=RSI_OFF()
37 IF (ret <> RSIOK) THEN

```

```

38   HALT
39   ENDIF
40
41   END

```

Zeile	Beschreibung
18	Startpunkt der Sensorkorrektur
19	Position des BASE-Koordinatensystems im aktuellen TCP
22	RSI_CREATE() lädt die Signalfluss-Konfiguration in einen RSI Container.
28	RSI_ON() schaltet die Signalverarbeitung ein. ■ Korrekturmodus: Relativkorrektur
33	Relative LIN-Bewegung in Y-Richtung (100 mm)
36	RSI_OFF() schaltet die Signalverarbeitung aus.

### Signalfluss-Konfiguration

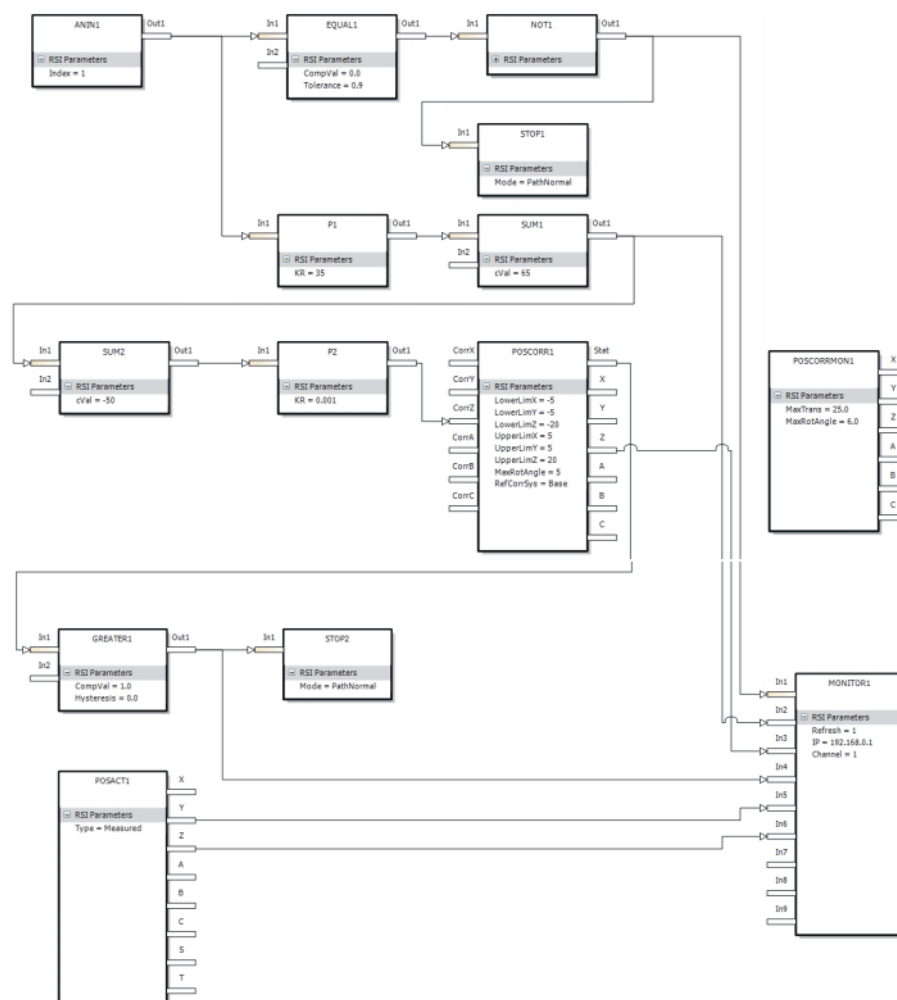


Abb. 8-6: Signalfluss - Bahnkorrektur zur Abstandssteuerung

RSI Objekt	Beschreibung
ANIN1	Liest über einen analogen Eingang das Sensorsignal ein.
EQUAL1	Mit EQUAL wird überprüft, ob das Sensorsignal innerhalb einer Toleranzgrenze liegt. Wenn dies nicht der Fall ist (NOT), bremst der Roboter bahntreu.
NOT1	
STOP1	

RSI Objekt	Beschreibung
P1	Mit P1 wird das Sensorsignal umgerechnet, z. B. ergeben 5 V einen Abstand von 10 cm (= Istabstand). Der Istabstand (SUM1) wird mit dem Sollabstand (SUM2) addiert. Das Ergebnis ist der Korrekturwert in Z-Richtung in cm. Mit P2 wird der Korrekturwert in mm umgerechnet.
SUM1	
SUM2	
P2	
POSCORR1	Liest den berechneten Korrekturwert in Z-Richtung ein, der am Ausgang des Objekts P2 zur Verfügung steht. <ul style="list-style-type: none"> <li>Referenz-Koordinatensystem für die Korrekturaufschaltung: BASE</li> </ul>
POSCORRMON1	Begrenzt die maximale kartesische Gesamtkorrektur. <ul style="list-style-type: none"> <li>Maximale translatorische Auslenkung in X, Y, Z: 25 mm</li> <li>Maximale rotatorische Auslenkung der Drehwinkel: 6°</li> </ul>
GREATER1	Der Korrekturstatus, der am Ausgang "Stat" des Korrekturobjekts POSCORR zur Verfügung steht, wird überprüft. Wenn der Korrekturstatus >1 ist, d. h. die zulässige Korrektur wurde überschritten und automatisch auf die Maximalkorrektur ±20 mm begrenzt, bremst der Roboter bahntreu.
STOP2	
POSACT1	Liest die aktuelle kartesische Istposition des Roboters in Y- und Z-Richtung ein.
MONITOR1	Folgende Signale sind mit dem MONITOR-Objekt verknüpft und können mit RSI Monitor auf der Robotersteuerung angezeigt werden: <ul style="list-style-type: none"> <li>Analoges eingelesenes Sensorsignal [V]</li> <li>Berechneter Istabstand [cm]</li> <li>Korrekturwert in Z-Richtung [mm]</li> <li>Korrekturbegrenzung (true, false)</li> <li>Kartesische Istposition des Roboters in Y- und Z-Richtung [mm]</li> </ul>

### 8.1.7 Beispiel einer Transformation in neues Koordinatensystem

Hier wird die Programmierung einer Transformation von Positionsdaten beschrieben, die von einem Sensor erfasst werden.

Am Anbaufansch des Roboters ist neben dem Werkzeug ein Sensor angebracht, der die Position eines Werkstücks erfasst, z. B. eine Kamera. Die Sensordaten müssen ausgehend vom Sensor-Koordinatensystem in das BASE-Koordinatensystem der Robotersteuerung transformiert werden.

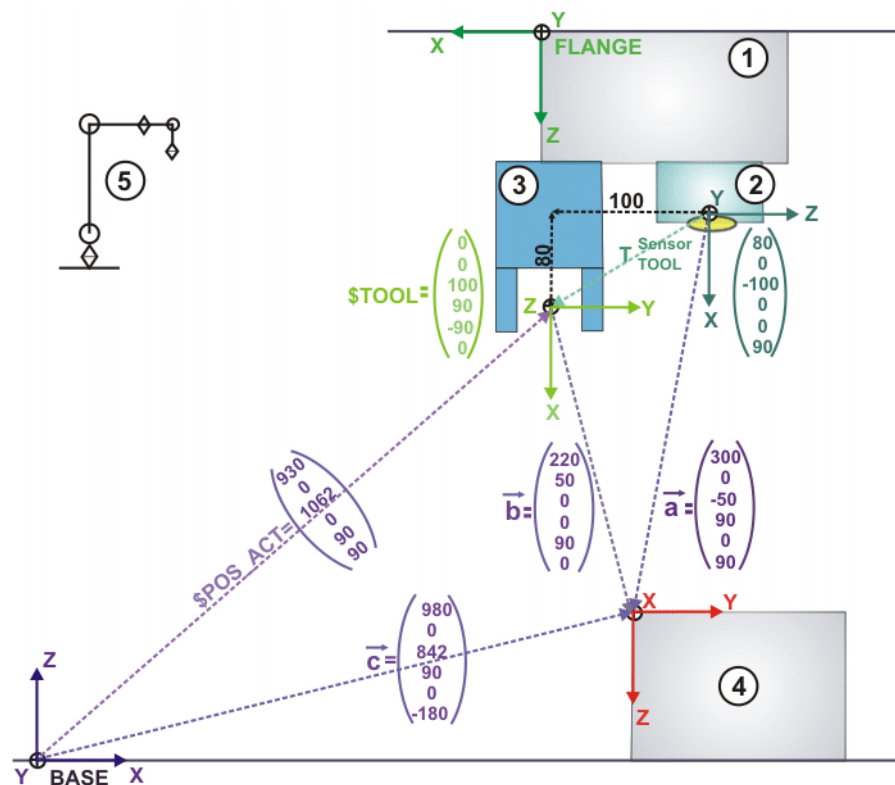


Abb. 8-7: Beispiel einer Transformation

- |                |                   |
|----------------|-------------------|
| 1 Anbauflansch | 4 Werkstück       |
| 2 Sensor       | 5 Roboterstellung |
| 3 Werkzeug     |                   |

Der Sensor erfasst die Position und die Orientierung eines Werkstücks im Sensor-Koordinatensystem (Vektor a). Im RSI Objekt TRAFO\_USERFRAME wird die Verschiebung und Verdrehung des Sensors zum Tool angegeben (T\_Sensor/Tool). TRAFO\_USERFRAME transformiert die Sensordaten in das TOOL-Koordinatensystem (Vektor b). Um die Sensordaten im BASE-Koordinatensystem zu erhalten, wird das RSI Objekt TRAFO\_ROBFRAME verwendet. TRAFO\_ROBFRAME transformiert die Tool-Koordinaten in das BASE-Koordinatensystem (Vektor c).

Mit dem Beispielprogramm kann das in der Abbildung angegebene Zahlenbeispiel überprüft werden. Dazu muss ein KR 16 eingestellt sein. Wenn die mit dem MONITOR-Objekt verknüpften Signale mit RSI Monitor auf der Robotersteuerung angezeigt werden, erhält man Position und Orientierung des Werkstücks im BASE-Koordinatensystem der Robotersteuerung (Vektor c).

## Programm

```

1 DEF RSI_SigTransformation( )
2 ; =====
3 ;
4 ; RSI EXAMPLE: Transformation of coordinates
5 ; Simulate a sensorsignal in relationship to
6 ; a flange mounted sensor. Transform the SIGNAL
7 ; to $BASE coordinates. Show the transformed
8 ; position in RSIMONITOR
9 ; =====
10
11 ; Declaration of KRL variables
12 DECL INT CONTID; ContainerID
13
14 INI

```

```

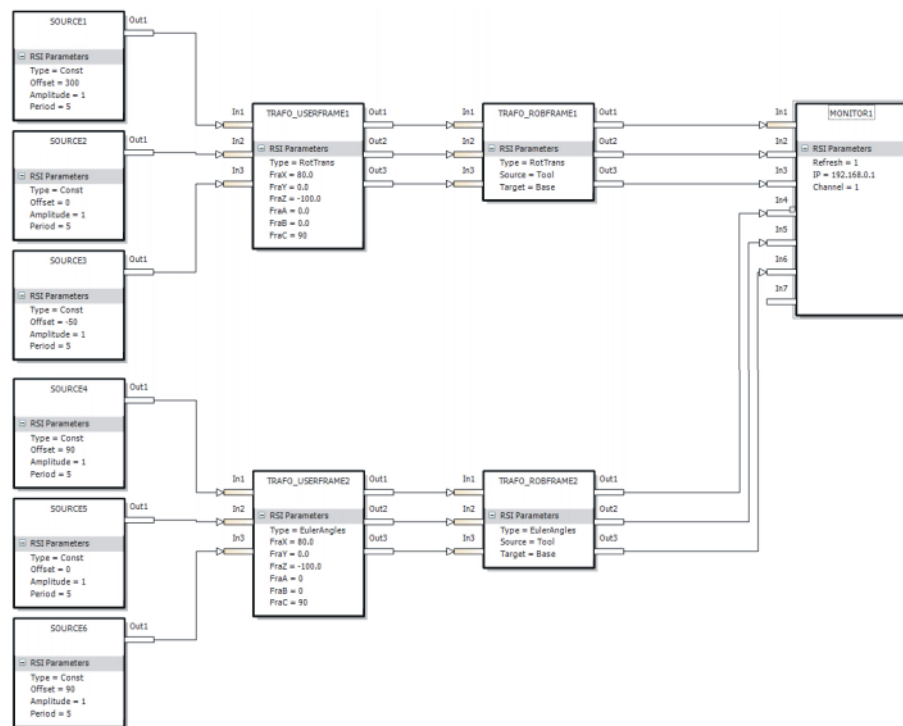
15
16 ; Move to start position
17 PTP {A1 0, A2 -90, A3 90, A4 0, A5 90, A6 0}
18 $TOOL = {X 0, Y 0, Z 100, A 90 ,B -90, C 0}
19 $BASE = $NULLFRAME
20
21 ; Create signal processing
22 IF (RSI_CREATE("RSI_SigTransformation.rsi") <> RSIOK) THEN
23     HALT
24 ENDIF
25
26 ; Start signal processing
27 IF (RSI_ON() <> RSIOK) THEN
28     HALT
29 ENDIF
30
31 wait sec 0.012
32
33 ; Turn off RSI
34 IF (RSI_OFF() <> RSIOK) THEN
35     HALT
36 ENDIF
37
38 END

```

Zeile	Beschreibung
17	Startposition der Transformation
18	Position des TOOL-Koordinatensystems
19	Position des BASE-Koordinatensystems (NULLFRAME)
22	RSI_CREATE() lädt die Signalfluss-Konfiguration in einen RSI Container.
27	RSI_ON() schaltet die Signalverarbeitung ein. <div> <div></div> Korrekturmodus: Relativkorrektur </div>
31	In der Wartezeit werden die Transformationsdaten berechnet.
34	RSI_OFF() schaltet die Signalverarbeitung aus.



## Signalfluss- Konfiguration



**Abb. 8-8: Signalfluss - Transformation**

RSI Objekt	Beschreibung
SOURCE1 ... SOURCE3	Liefern die Position des Werkstücks im Sensor-Koordinatensystem (Vektor a) und übergeben die Daten an TRAFO_USERFRAME1.
TRAFO_USERFRAME1	Transformiert die Positionsdaten des Werkstücks im Sensor-Koordinatensystem in das TOOL-Koordinatensystem der Robotersteuerung (Vektor b). Die Daten stehen an den Ausgängen des Objekts zur Verfügung.
TRAFO_ROBFRAME1	Transformiert die Positionsdaten des Werkstücks im TOOL-Koordinatensystem in das BASE-Koordinatensystem der Robotersteuerung (Vektor c). Die Daten stehen an den Ausgängen des Objekts zur Verfügung.
SOURCE4 ... SOURCE6	Liefern die Orientierung des Werkstücks im Sensor-Koordinatensystem (Vektor a) und übergeben die Daten an TRAFO_USERFRAME2.
TRAFO_USERFRAME2	Transformiert die Orientierungswinkel des Werkstücks im Sensor-Koordinatensystem in das TOOL-Koordinatensystem der Robotersteuerung (Vektor b). Die Daten stehen an den Ausgängen des Objekts zur Verfügung.
TRAFO_ROBFRAME2	Transformiert die Orientierungswinkel des Werkstücks im TOOL-Koordinatensystem in das BASE-Koordinatensystem der Robotersteuerung (Vektor c). Die Daten stehen an den Ausgängen des Objekts zur Verfügung.
MONITOR1	Folgende Signale sind mit dem MONITOR-Objekt verknüpft und können mit RSI Monitor auf der Robotersteuerung angezeigt werden: <ul style="list-style-type: none"> <li>■ Ergebnis der Transformation (Vektor c): Position und Orientierung des Werkstücks im BASE-Koordinatensystem der Robotersteuerung</li> </ul>



## 9 Diagnose

### 9.1 Diagnosedaten zur RSI anzeigen

- Vorgehensweise**
1. Im Hauptmenü **Diagnose** > **Diagnosemonitor** wählen.
  2. Im Feld **Modul** das Modul **RSI Diagnose** auswählen.

**Beschreibung** Diagnosedaten zur RSI:

Name	Beschreibung
Status	Status der Signalverarbeitung <ul style="list-style-type: none"> <li>■ <b>Aktiv IPO</b>: Signalverarbeitung im IPO-Modus</li> <li>■ <b>Aktiv IPO_FAST</b>: Signalverarbeitung im Modus IPO_FAST</li> <li>■ <b>Inaktiv</b>: Keine Signalverarbeitung</li> </ul>
Sensor Takt	Zykluszeit der Signalverarbeitung
Zähler	Anzahl der Berechnungszyklen seit dem Start der Signalverarbeitung
Ausführungszeit	Zeit, die zur Berechnung des aktuellen RSI Kontextes benötigt wird
Ausführungszeit (Min)	Minimale Zeit zur Berechnung des aktuellen RSI Kontextes
Ausführungszeit (Max)	Maximale Zeit zur Berechnung des aktuellen RSI Kontextes
Ausführungszeit (Mittel)	Durchschnittliche Zeit zur Berechnung des aktuellen RSI Kontextes
Objekt Zähler	Anzahl der angelegten RSI Objekte
Speicher	Ingesamt verfügbarer Speicher für RSI (Byte)
Benutzter Speicher	Benutzer Speicher (Byte)
Kommunikationszyklen	Erfolgreiche Kommunikationszyklen seit dem Start der Signalverarbeitung
Gesamtverlust	Anzahl der Paketverluste seit dem Start der Signalverarbeitung
Verbindungsqualität	Qualität der Signalverarbeitung <ul style="list-style-type: none"> <li>■ <b>0 ... 100 %</b></li> </ul> 100 % = Alle Pakete sind erfolgreich eingetroffen. 0 % = Kein Paket ist erfolgreich eingetroffen.
Maximal zusammenhängender Verlust	Größter zusammenhängender Verlust von Paketen seit Beginn der Signalverarbeitung

### 9.2 Fehlerprotokoll (Logbuch)

Defaultmäßig werden die Fehlermeldungen der Schnittstelle in einer LOG-Datei unter C:\KRC\ROBOTER\LOG\SensorInterface protokolliert.

Der LOG-Level kann so geändert werden, dass zusätzlich Hinweismeldungen protokolliert werden.

#### 9.2.1 LOG-Level konfigurieren

In der Datei C:\KRC\Roboter\Config\User\Common\Logging\_RSI.xml kann der LOG-Level geändert werden.

**Voraussetzung**

- Benutzergruppe Experte
- Betriebsart T1 oder T2
- Es ist kein Programm angewählt.

**Vorgehensweise**

1. Datei öffnen.
2. In dieser Zeile den LOG-Level ändern:

```
<Class Name="RSILogger1" LogLevel="error" />
```

3. Änderung speichern.

**Beschreibung**

LogLevel	Beschreibung
error	Fehlermeldungen der Schnittstelle werden protokolliert.
info	Fehlermeldungen und Hinweismeldungen der Schnittstelle werden protokolliert.

## 10 Meldungen

### 10.1 Meldungen während des Betriebs

Nr.	Meldung	Ursache	Abhilfe
29000	<i>{Type} Zulässige Gesamtkorrektur überschritten: RSI wird angehalten</i>	Die Zulässige Gesamtkorrektur wurde überschritten.	Meldung quittieren
29001	<i>{Typ} Korrektur außerhalb des zulässigen Bereichs: {Wert}</i>	Die kommandierte Korrektur überschreitet den für das Korrekturobjekt definierten zulässigen Bereich.	<ul style="list-style-type: none"> <li>■ Die zulässige Korrektur vergrößern</li> <li>■ Die Signalverarbeitung überprüfen</li> </ul>
29002	<i>Signalfluss ({Mode}): Objekt {ObjName} liefert Fehler {ErrorCode}</i>	Das genannte RSI-Objekt kann in RSI-Visual nicht berechnet werden.	Überprüfen, ob das RSI-Objekt in RSI-Visual richtig konfiguriert wurde und ob die benötigte Konfigurationsdateien vorhanden und korrekt sind.
29004	<i>Interner RSI Fehler</i>	Unerwarteter Rückgabewert wurde von einer Funktion geliefert.	Besteht das Problem dauerhaft, den KUKA Service kontaktieren.
29005	<i>RSI kann keine Ausgänge setzen wegen Bediener-schutz</i>	RSI kann keine Ausgänge setzen wegen dem Bediener-schutz.	Signal Bediener-schutz darf nicht gesetzt sein.
29006	<i>RSI: Zeitüberschreitung der Signalberechnung {CalcTime} usec</i>	Der konfigurierte RSI Kontext ist zu umfangreich um in der zur Verfügung stehenden Zeit berechnet zu werden.	RSI Kontext verkleinern



## 11 Anhang

### 11.1 Speicher erhöhen



Der Speicher darf nur nach Rücksprache mit der KUKA Roboter GmbH erhöht werden. (>>> 12 "KUKA Service" Seite 77)

**Beschreibung** Wenn der zur Verfügung stehende Speicher nicht ausreicht, wird empfohlen die Programmierweise in KRL und die Signalfluss-Konfiguration zu überprüfen.

**Voraussetzung** ■ Windows-Ebene

**Vorgehensweise**

1. Datei C:\KRC\ROBOTER\Config\User\Common\RSI.XML öffnen.
2. Im Abschnitt <Interface> im Element <MemSize> die gewünschte Speicherkapazität in Byte eintragen.

```
<Interface>
  <MemSize>500000</MemSize>
</Interface>
```

3. Änderung speichern und Datei schließen.

### 11.2 RSI Objekt-Bibliothek

#### 11.2.1 RSI Objekte zur Korrekturüberwachung

Name	Beschreibung
POSCORRMON	Begrenzung für die kartesische Gesamtkorrektur Bei Überschreitung muss das Roboterprogramm zurückgesetzt werden. Die Ausgänge des Objekts liefern die aktuelle Gesamtkorrektur.
AXISCORRMON	Begrenzung für die achsspezifische Gesamtkorrektur Bei Überschreitung muss das Roboterprogramm zurückgesetzt werden. Die Ausgänge des Objekts liefern die aktuelle Gesamtkorrektur.

#### 11.2.2 RSI Objekte zur Signalübertragung

Name	Beschreibung
DIGIN	Liefert den Wert eines Bereichs von digitalen Eingängen \$IN.
DIGOUT	Liefert den Wert eines Bereichs von digitalen Ausgängen \$OUT
ANIN	Liefert den Wert eines analogen Eingangs \$ANIN.
ANOUT	Liefert den Wert eines analogen Ausganges \$ANOUT.
SEN_PINT	Liefert den Wert der Systemvariablen \$SEN_PINT.
SEN_PREA	Liefert den Wert der Systemvariablen \$SEN_PREA.
POSACT	Liefert die aktuelle kartesische Roboterposition.
AXISACT	Liefert die aktuellen Achswinkel der Roboterachsen A1... A6.
AXISACTEXT	Liefert die aktuellen Positionen der Zusatzachsen E1... E6.
SOURCE	Signalgenerator Erzeugt einen definierten Signalverlauf, z. B. über ein konstantes Signal, ein sinus- oder cosinusförmiges Signal etc.
GEARTORQUE	Liefert die Getriebemomente der Roboterachsen A1... A6.

Name	Beschreibung
GEARTORQUEEXT	Liefert die Getriebemomente der Zusatzachsen E1... E6.
MOTORCURRENT	Liefert die Motorströme der Roboterachsen A1... A6.
MOTORCURREN- TEXT	Liefert die Motorströme der Zusatzachsen E1... E6.
STATUS	Liefert Statusinformationen der Robotersteuerung, z. B. aktueller Status von Submit- oder Roboter-Interpreter, aktuelle Betriebsart, etc.
OV_PRO	Liefert den aktuellen Programm-Override \$OV_PRO.

### 11.2.3 RSI Objekte zur Koordinatentransformation

Name	Beschreibung
TRAFO_ USERFRAME	Transformiert einen Vektor aus den Eingängen 1 ... 3 in ein neues Referenz-Koordinatensystem mit definierter Verschiebung und Verdrehung.
TRAFO_ ROBFRAME	Transformiert einen Vektor aus den Eingängen 1 ... 3 von einem Roboter Referenz-Koordinatensystem in ein anderes.

### 11.2.4 RSI Objekte zur logischen Verknüpfung

Name	Beschreibung
AND	UND-Verknüpfung Es können bis zu 10 Eingangssignale angeschlossen werden.
OR	ODER-Verknüpfung Es können bis zu 10 Eingangssignale angeschlossen werden.
XOR	ENTWEDER-ODER-Verknüpfung Es können bis zu 10 Eingangssignale angeschlossen werden.
NOT	Logische Negierung

### 11.2.5 RSI Objekte zur binären logischen Verknüpfung

Name	Beschreibung
BAND	Binäre UND-Verknüpfung Verknüpft den Signaleingang 1 mit einem konstanten Wert. Sind mehrere Signaleingänge verlinkt, werden diese miteinander verknüpft. Es können bis zu 10 Eingangssignale angeschlossen werden.
BOR	Binäre ODER-Verknüpfung Verknüpft den Signaleingang 1 mit einem konstanten Wert. Sind mehrere Signaleingänge verlinkt, werden diese miteinander verknüpft. Es können bis zu 10 Eingangssignale angeschlossen werden.
BCOMPL	Binäres Komplement



### 11.2.6 RSI Objekte für mathematische Vergleiche

Name	Beschreibung
EQUAL	Vergleich auf Gleichheit Vergleicht Signaleingang 1 mit einen konstanten Wert oder vergleicht Signaleingang 1 und 2 miteinander.
GREATER	Vergleich auf Größer Vergleicht Signaleingang 1 mit einen konstanten Wert oder vergleicht Signaleingang 1 und 2 miteinander.
LESS	Vergleich auf Kleiner Vergleicht Signaleingang 1 mit einen konstanten Wert oder vergleicht Signaleingang 1 und 2 miteinander.

### 11.2.7 RSI Objekte für mathematische Operationen

Name	Beschreibung
SUM	Addition von Signalen Es können bis zu 10 Eingangssignale angeschlossen werden. Mit dem RSI Objekt-Parameter <b>cVal</b> kann ein konstanter Wert addiert werden.
MULTI	Multiplikation von Signalen
ABS	Betrag-Funktion
POW	Potenz-Funktion
SIN	Sinus-Funktion
COS	Kosinus-Funktion
TAN	Tangens-Funktion
ASIN	Arkussinus-Funktion
ACOS	Arkuskosinus-Funktion
ATAN	Arkustangens-Funktion
EXP	Exponential-Funktion
LOG	Logarithmus-Funktion
CEIL	Kleinste Integer-Zahl größer oder gleich Eingangssignal
FLOOR	Größte Integer-Zahl größer oder gleich Eingangssignal
ROUND	Rundungs-Funktion
ATAN2	Arkustangens des Quotienten von Eingang 1 und 2 Der Quadrant des Ergebnisses wird aus den Vorzeichen der Eingangssignale ermittelt.

### 11.2.8 RSI Objekte zur Signalsteuerung

Name	Beschreibung
P	Verstärkung des Signals
PD	Proportional-Differential-Objekt $y(k) = B0 * x(k) + B1 * x(k-1)$ $B0 = KR * (1 + (TV / <Sensortakt>))$ $B1 = -KR * (TV / <Sensortakt>)$

Name	Beschreibung
I	Integrationsobjekt (Trapez-Algorithmus) $y(k) = B0 * (x(k) + x(k-1)) + y(k-1)$ $B0 = \langle \text{Sensortakt} \rangle / (2 * T1)$
D	Differenzierungsobjekt $y(k) = B0 * (x(k) - x(k-1))$ $B0 = KD / \langle \text{Sensortakt} \rangle$
PID	PID-Objekt $y(k) = y(k-1) + B0 * x(k) + B1 * x(k-1) + B2 * x(k-2)$ $B0 = KR * (1 + TV / \langle \text{Sensortakt} \rangle)$ $B1 = -KR * (1 - \langle \text{Sensortakt} \rangle / TN + 2 * TV / \langle \text{Sensortakt} \rangle)$ $B2 = KR * TV / \langle \text{Sensortakt} \rangle$
PT1	Verzögerungsobjekt 1. Ordnung $y(k) = -A0 * y(k-1) + B0 * x(k)$ $A0 = -\exp(-\langle \text{Sensortakt} \rangle / T1)$ $B0 = KR * (1 - \exp(-\langle \text{Sensortakt} \rangle / T1))$
PT2	Verzögerungsobjekt 2. Ordnung $y(k) = -A0 * y(k-1) - A1 * y(k-2) + B0 * x(k) + B1 * x(k-1)$ Fall 1: $T1 \neq T2$ <ul style="list-style-type: none"> <li>■ <math>Z1 = \exp(-\langle \text{Sensortakt} \rangle / T1)</math></li> <li>■ <math>Z2 = \exp(-\langle \text{Sensortakt} \rangle / T2)</math></li> <li>■ <math>A0 = -Z1 - Z2</math> <math>A1 = Z1 * Z2</math></li> <li>■ <math>B0 = (KP / (T1 - T2)) / (T1 * (1 - Z1) - T2 * (1 - Z2))</math></li> <li>■ <math>B1 = (KP / (T1 - T2)) / (T2 * Z1 * (1 - Z2) - T1 * Z2 * (1 - Z1))</math></li> </ul> Fall 2: $T1 == T2$ <ul style="list-style-type: none"> <li>■ <math>Z0 = \exp(-\langle \text{Sensortakt} \rangle / T1)</math></li> <li>■ <math>B0 = KP * (1 - Z0 * (\langle \text{Sensortakt} \rangle / T1 + 1))</math></li> <li>■ <math>B1 = KP * Z0 * (Z0 + (\langle \text{Sensortakt} \rangle / T1) - 1)</math></li> </ul>
GENCTRL	Generisches Signalverarbeitungsobjekt bis zur 8. Ordnung $y(z) = B0*u(z) + B1*u(z-1) + B2*u(z-2) + \dots + B8*u(z-8) - A1*y(z-1) - A2*y(z-2) - \dots - A8*y(z-8)$
IIRFILTER	IIR-FILTER

### 11.2.9 RSI Objekte weitere

Name	Beschreibung
TIMER	Nach Ablauf der eingestellten Zeit wird am Signalausgang "Out1" eine positive Flanke ausgegeben.
LIMIT	Beschränkt ein Signal auf Werte innerhalb einer unteren und oberen Grenze (LowerLimit, UpperLimit).
MINMAX	Liefert das aktuell kleinste und größte Signal über alle Eingangssignale. Es können bis zu 10 Eingangssignale angeschlossen werden.
DELAY	Verzögert das Eingangssignal um eine bestimmte Zeit.

Name	Beschreibung
SIGNALSWITCH	Umschalten zwischen 2 Signalpfaden mittels Steuersignal
ETHERNET	UDP Ethernet-Kommunikation im XML Datenformat  Bis zu 64 Ein- und Ausgänge können in der Konfigurationsdatei definiert werden. Signale an den Eingängen werden an den Kommunikationspartner gesendet. An den Ausgängen stehen die vom Kommunikationspartner empfangenen Daten zur Verfügung.

### 11.2.10 RSI Objekte für Aktionen

Name	Beschreibung
MAP2OV_PRO	Ändert den Programm-Override (\$OV_PRO).
STOP	Anhalten einer Bewegung bei positiver Signalfanke  Eine rein sensorgeführte Bewegung mit RSI_MOVECORR kann mit dem Modus <b>ExitMoveCorr</b> abgebrochen werden.
MAP2SEN_PINT	Ändert den Wert der Systemvariablen \$SEN_PINT.
MAP2SEN_PREA	Ändert den Wert der Systemvariablen \$SEN_PREA.
MAP2DIGOUT	Beschreibt einen digitalen Ausgang \$OUT oder einen Bereich von digitalen Ausgängen.
MAP2ANOUT	Beschreibt einen analogen Ausgang \$ANOUT.
SETDIGOUT	Setzt bei positiver Flanke einen digitalen Ausgang \$OUT.  Der gesetzte Ausgang wird auch bei negativer Flanke gehalten.
RESETDIGOUT	Setzt bei positiver Flanke einen digitalen Ausgang \$OUT zurück.  Der zurückgesetzte Ausgang wird auch bei negativer Flanke gehalten.
POSCORR	Kartesische Korrekturaufschaltung mit Begrenzung
AXISCORR	Achswise Korrekturaufschaltung mit Begrenzung Roboterachsen A1 ... A6
AXISCORREXT	Achswise Korrekturaufschaltung mit Begrenzung Zusatzachsen E1 ... E6
MONITOR	RSI Monitor  Visualisierung von bis zu 24 RSI Signalen



## 12 KUKA Service

### 12.1 Support-Anfrage

<b>Einleitung</b>	Diese Dokumentation bietet Informationen zu Betrieb und Bedienung und unterstützt Sie bei der Behebung von Störungen. Für weitere Anfragen steht Ihnen die lokale Niederlassung zur Verfügung.
<b>Informationen</b>	<p>Zur Abwicklung einer Anfrage werden folgende Informationen benötigt:</p> <ul style="list-style-type: none"> <li>■ Typ und Seriennummer des Manipulators</li> <li>■ Typ und Seriennummer der Steuerung</li> <li>■ Typ und Seriennummer der Lineareinheit (wenn vorhanden)</li> <li>■ Typ und Seriennummer der Energiezuführung (wenn vorhanden)</li> <li>■ Version der System Software</li> <li>■ Optionale Software oder Modifikationen</li> <li>■ Diagnosepaket <b>KrcDiag</b></li> </ul> <p>Für KUKA Sunrise zusätzlich: Vorhandene Projekte inklusive Applikationen</p> <p>Für Versionen der KUKA System Software älter als V8: Archiv der Software (<b>KrcDiag</b> steht hier noch nicht zur Verfügung.)</p> <ul style="list-style-type: none"> <li>■ Vorhandene Applikation</li> <li>■ Vorhandene Zusatzachsen</li> <li>■ Problembeschreibung, Dauer und Häufigkeit der Störung</li> </ul>

### 12.2 KUKA Customer Support

<b>Verfügbarkeit</b>	Der KUKA Customer Support ist in vielen Ländern verfügbar. Bei Fragen stehen wir gerne zur Verfügung!
<b>Argentinien</b>	<p>Ruben Costantini S.A. (Agentur)</p> <p>Luis Angel Huergo 13 20</p> <p>Parque Industrial</p> <p>2400 San Francisco (CBA)</p> <p>Argentinien</p> <p>Tel. +54 3564 421033</p> <p>Fax +54 3564 428877</p> <p>ventas@costantini-sa.com</p>
<b>Australien</b>	<p>KUKA Robotics Australia Pty Ltd</p> <p>45 Fennell Street</p> <p>Port Melbourne VIC 3207</p> <p>Australien</p> <p>Tel. +61 3 9939 9656</p> <p>info@kuka-robotics.com.au</p> <p>www.kuka-robotics.com.au</p>

<b>Belgien</b>	<p>KUKA Automatisering + Robots N.V.  Centrum Zuid 1031  3530 Houthalen  Belgien  Tel. +32 11 516160  Fax +32 11 526794  <a href="mailto:info@kuka.be">info@kuka.be</a>  <a href="http://www.kuka.be">www.kuka.be</a></p>
<b>Brasilien</b>	<p>KUKA Roboter do Brasil Ltda.  Travessa Claudio Armando, nº 171  Bloco 5 - Galpões 51/52  Bairro Assunção  CEP 09861-7630 São Bernardo do Campo - SP  Brasilien  Tel. +55 11 4942-8299  Fax +55 11 2201-7883  <a href="mailto:info@kuka-roboter.com.br">info@kuka-roboter.com.br</a>  <a href="http://www.kuka-roboter.com.br">www.kuka-roboter.com.br</a></p>
<b>Chile</b>	<p>Robotec S.A. (Agency)  Santiago de Chile  Chile  Tel. +56 2 331-5951  Fax +56 2 331-5952  <a href="mailto:robotec@robotec.cl">robotec@robotec.cl</a>  <a href="http://www.robotec.cl">www.robotec.cl</a></p>
<b>China</b>	<p>KUKA Robotics China Co., Ltd.  No. 889 Kungang Road  Xiaokunshan Town  Songjiang District  201614 Shanghai  P. R. China  Tel. +86 21 5707 2688  Fax +86 21 5707 2603  <a href="mailto:info@kuka-robotics.cn">info@kuka-robotics.cn</a>  <a href="http://www.kuka-robotics.com">www.kuka-robotics.com</a></p>
<b>Deutschland</b>	<p>KUKA Roboter GmbH  Zugspitzstr. 140  86165 Augsburg  Deutschland  Tel. +49 821 797-4000  Fax +49 821 797-1616  <a href="mailto:info@kuka-roboter.de">info@kuka-roboter.de</a>  <a href="http://www.kuka-roboter.de">www.kuka-roboter.de</a></p>

**Frankreich** KUKA Automatismes + Robotique SAS  
 Techvallée  
 6, Avenue du Parc  
 91140 Villebon S/Yvette  
 Frankreich  
 Tel. +33 1 6931660-0  
 Fax +33 1 6931660-1  
[commercial@kuka.fr](mailto:commercial@kuka.fr)  
[www.kuka.fr](http://www.kuka.fr)

**Indien** KUKA Robotics India Pvt. Ltd.  
 Office Number-7, German Centre,  
 Level 12, Building No. - 9B  
 DLF Cyber City Phase III  
 122 002 Gurgaon  
 Haryana  
 Indien  
 Tel. +91 124 4635774  
 Fax +91 124 4635773  
[info@kuka.in](mailto:info@kuka.in)  
[www.kuka.in](http://www.kuka.in)

**Italien** KUKA Roboter Italia S.p.A.  
 Via Pavia 9/a - int.6  
 10098 Rivoli (TO)  
 Italien  
 Tel. +39 011 959-5013  
 Fax +39 011 959-5141  
[kuka@kuka.it](mailto:kuka@kuka.it)  
[www.kuka.it](http://www.kuka.it)

**Japan** KUKA Robotics Japan K.K.  
 YBP Technical Center  
 134 Godo-cho, Hodogaya-ku  
 Yokohama, Kanagawa  
 240 0005  
 Japan  
 Tel. +81 45 744 7691  
 Fax +81 45 744 7696  
[info@kuka.co.jp](mailto:info@kuka.co.jp)

**Kanada** KUKA Robotics Canada Ltd.  
 6710 Maritz Drive - Unit 4  
 Mississauga  
 L5W 0A1  
 Ontario  
 Kanada  
 Tel. +1 905 670-8600  
 Fax +1 905 670-8604  
[info@kukarobotics.com](mailto:info@kukarobotics.com)  
[www.kuka-robotics.com/canada](http://www.kuka-robotics.com/canada)

<b>Korea</b>	<p>KUKA Robotics Korea Co. Ltd.  RIT Center 306, Gyeonggi Technopark  1271-11 Sa 3-dong, Sangnok-gu  Ansan City, Gyeonggi Do  426-901  Korea  Tel. +82 31 501-1451  Fax +82 31 501-1461  <a href="mailto:info@kukakorea.com">info@kukakorea.com</a></p>
<b>Malaysia</b>	<p>KUKA Robot Automation (M) Sdn Bhd  South East Asia Regional Office  No. 7, Jalan TPP 6/6  Taman Perindustrian Puchong  47100 Puchong  Selangor  Malaysia  Tel. +60 (03) 8063-1792  Fax +60 (03) 8060-7386  <a href="mailto:info@kuka.com.my">info@kuka.com.my</a></p>
<b>Mexiko</b>	<p>KUKA de México S. de R.L. de C.V.  Progreso #8  Col. Centro Industrial Puente de Vigas  Tlalnepantla de Baz  54020 Estado de México  Mexiko  Tel. +52 55 5203-8407  Fax +52 55 5203-8148  <a href="mailto:info@kuka.com.mx">info@kuka.com.mx</a>  <a href="http://www.kuka-robotics.com/mexico">www.kuka-robotics.com/mexico</a></p>
<b>Norwegen</b>	<p>KUKA Sveiseanlegg + Roboter  Sentrumsvegen 5  2867 Hov  Norwegen  Tel. +47 61 18 91 30  Fax +47 61 18 62 00  <a href="mailto:info@kuka.no">info@kuka.no</a></p>
<b>Österreich</b>	<p>KUKA Roboter CEE GmbH  Gruberstraße 2-4  4020 Linz  Österreich  Tel. +43 7 32 78 47 52  Fax +43 7 32 79 38 80  <a href="mailto:office@kuka-roboter.at">office@kuka-roboter.at</a>  <a href="http://www.kuka.at">www.kuka.at</a></p>



**Polen**  
KUKA Roboter Austria GmbH  
Spółka z ograniczoną odpowiedzialnością  
Oddział w Polsce  
Ul. Porcelanowa 10  
40-246 Katowice  
Polen  
Tel. +48 327 30 32 13 or -14  
Fax +48 327 30 32 26  
ServicePL@kuka-roboter.de

**Portugal**  
KUKA Sistemas de Automatización S.A.  
Rua do Alto da Guerra n° 50  
Armazém 04  
2910 011 Setúbal  
Portugal  
Tel. +351 265 729780  
Fax +351 265 729782  
kuka@mail.telepac.pt

**Russland**  
KUKA Robotics RUS  
Werbnaja ul. 8A  
107143 Moskau  
Russland  
Tel. +7 495 781-31-20  
Fax +7 495 781-31-19  
info@kuka-robotics.ru  
www.kuka-robotics.ru

**Schweden**  
KUKA Svetsanläggningar + Robotar AB  
A. Odhners gata 15  
421 30 Västra Frölunda  
Schweden  
Tel. +46 31 7266-200  
Fax +46 31 7266-201  
info@kuka.se

**Schweiz**  
KUKA Roboter Schweiz AG  
Industriestr. 9  
5432 Neuenhof  
Schweiz  
Tel. +41 44 74490-90  
Fax +41 44 74490-91  
info@kuka-roboter.ch  
www.kuka-roboter.ch

<b>Spanien</b>	<p>KUKA Robots IBÉRICA, S.A.  Pol. Industrial  Torrent de la Pastera  Carrer del Bages s/n  08800 Vilanova i la Geltrú (Barcelona)  Spanien  Tel. +34 93 8142-353  Fax +34 93 8142-950  Comercial@kuka-e.com  www.kuka-e.com</p>
<b>Südafrika</b>	<p>Jendamark Automation LTD (Agentur)  76a York Road  North End  6000 Port Elizabeth  Südafrika  Tel. +27 41 391 4700  Fax +27 41 373 3869  www.jendamark.co.za</p>
<b>Taiwan</b>	<p>KUKA Robot Automation Taiwan Co., Ltd.  No. 249 Pujong Road  Jungli City, Taoyuan County 320  Taiwan, R. O. C.  Tel. +886 3 4331988  Fax +886 3 4331948  info@kuka.com.tw  www.kuka.com.tw</p>
<b>Thailand</b>	<p>KUKA Robot Automation (M)SdnBhd  Thailand Office  c/o Maccall System Co. Ltd.  49/9-10 Soi Kingkaew 30 Kingkaew Road  Tt. Rachatheva, A. Bangpli  Samutprakarn  10540 Thailand  Tel. +66 2 7502737  Fax +66 2 6612355  atika@ji-net.com  www.kuka-roboter.de</p>
<b>Tschechien</b>	<p>KUKA Roboter Austria GmbH  Organisation Tschechien und Slowakei  Sezemická 2757/2  193 00 Praha  Horní Počernice  Tschechische Republik  Tel. +420 22 62 12 27 2  Fax +420 22 62 12 27 0  support@kuka.cz</p>

**Ungarn** KUKA Robotics Hungaria Kft.  
Fő út 140  
2335 Taksony  
Ungarn  
Tel. +36 24 501609  
Fax +36 24 477031  
info@kuka-robotics.hu

**USA** KUKA Robotics Corporation  
51870 Shelby Parkway  
Shelby Township  
48315-1787  
Michigan  
USA  
Tel. +1 866 873-5852  
Fax +1 866 329-5852  
info@kukarobotics.com  
www.kukarobotics.com

**Vereinigtes Königreich** KUKA Automation + Robotics  
Hereward Rise  
Halesowen  
B62 8AN  
Vereinigtes Königreich  
Tel. +44 121 585-0800  
Fax +44 121 585-0900  
sales@kuka.co.uk



## Index

### A

ABS (RSI Objekt) 73  
 ACOS (RSI Objekt) 73  
 AND (RSI Objekt) 72  
 Anhang 71  
 ANIN (RSI Objekt) 71  
 ANOUT (RSI Objekt) 71  
 ASIN (RSI Objekt) 73  
 ATAN (RSI Objekt) 73  
 ATAN2 (RSI Objekt) 73  
 AXISACT (RSI Objekt) 71  
 AXISACTEXT (RSI Objekt) 71  
 AXISCORR (RSI Objekt) 75  
 AXISCORR (RSI Objekt) 16  
 AXISCORREXT (RSI Objekt) 75  
 AXISCORREXT (RSI Objekt) 16  
 AXISCORRMON (RSI Objekt) 71

### B

BAND (RSI Objekt) 72  
 BCOMPL (RSI Objekt) 72  
 Bedienoberfläche, RSI Monitor 32  
 Bedienoberfläche, RSI Visual 29  
 Bedienung 29  
 Begriffe, verwendete 8  
 Beispielapplikation, implementieren 52  
 Beispiele 51  
 Beispielkonfigurationen 51  
 Beispielprogramme 51  
 BOR (RSI Objekt) 72

### C

CCS 8  
 CEIL (RSI Objekt) 73  
 COS (RSI Objekt) 73

### D

D (RSI Objekt) 74  
 Datenaustausch, Funktionsweise 13  
 Deinstallieren, RobotSensorInterface 24  
 Deinstallieren, RSI Visual 25  
 DELAY (RSI Objekt) 74  
 Diagnose 67  
 Diagnosemonitor (Menüpunkt) 67  
 DIGIN (RSI Objekt) 71  
 DIGOUT (RSI Objekt) 71  
 Dokumentation, Industrieroboter 7

### E

Einleitung 7  
 EQUAL (RSI Objekt) 73  
 Ethernet 8  
 ETHERNET (RSI Objekt) 75  
 Ethernet-Verbindung, XML-Datei 43  
 Ethernet, Schnittstellen 27  
 Ethernet, Sensor-Netzwerk 27  
 EXP (RSI Objekt) 73

### F

Fehlerbehandlung 28  
 Fehlermeldungen 69  
 Fehlerprotokoll 67  
 FLOOR (RSI Objekt) 73  
 Funktionen 11  
 Funktionsgenerator 23, 28  
 Funktionsweise, Datenaustausch 13  
 Funktionsweise, Sensorkorrektur 15  
 Funktionsweise, Signalverarbeitung 12

### G

GEARTORQUE (RSI Objekt) 71  
 GEARTORQUEEXT (RSI Objekt) 72  
 GENCTRL (RSI Objekt) 74  
 GREATER (RSI Objekt) 73

### H

Hardware 23  
 Hinweise 7

### I

I (RSI Objekt) 74  
 IIRFILTER (RSI Objekt) 74  
 Installation 23  
 Installieren, RobotSensorInterface 23  
 Installieren, RSI Visual 24  
 IP 9  
 IPO\_FAST, Sensormodus 9, 12, 37  
 IPO, Sensormodus 9, 12, 18, 23, 28, 37

### K

Kanalnummer, einstellen 33  
 Kenntnisse, benötigte 7  
 KLI 8, 27  
 Kommunikation 11  
 Konfiguration 27  
 KR C 8  
 KUKA Customer Support 77

### L

LESS (RSI Objekt) 73  
 LIMIT (RSI Objekt) 74  
 LOG (RSI Objekt) 73  
 LOG-Level, konfigurieren 67  
 Logbuch 67  
 Logging\_RSI.xml 67, 68

### M

MAP2ANOUT (RSI Objekt) 75  
 MAP2DIGOUT (RSI Objekt) 75  
 MAP2OV\_PRO (RSI Objekt) 75  
 MAP2SEN\_PINT (RSI Objekt) 75  
 MAP2SEN\_PREA (RSI Objekt) 75  
 Marken 9  
 Meldungen 69  
 MINMAX (RSI Objekt) 74  
 MONITOR (RSI Objekt) 75

MOTORCURRENT (RSI Objekt) 72  
 MOTORCURRENTTEXT (RSI Objekt) 72  
 MULTI (RSI Objekt) 73

## N

Netzwerkverbindung 27  
 Netzwerkverbindung, konfigurieren 27  
 NOT (RSI Objekt) 72

## O

OR (RSI Objekt) 72  
 OV\_PRO (RSI Objekt) 72

## P

P (RSI Objekt) 73  
 PD (RSI Objekt) 73  
 PID (RSI Objekt) 74  
 POSACT (RSI Objekt) 71  
 POSCORR (RSI Objekt) 75  
 POSCORR (RSI Objekt) 16  
 POSCORRMON (RSI Objekt) 71  
 POW (RSI Objekt) 73  
 Produktbeschreibung 11  
 Programmierung 35  
 PT1 (RSI Objekt) 74  
 PT2 (RSI Objekt) 74

## R

RESETDIGOUT (RSI Objekt) 75  
 RoboTeam 16, 23  
 RobotSensorInterface, Übersicht 11  
 ROUND (RSI Objekt) 73  
 RSI 8  
 RSI Befehle, Übersicht 35  
 RSI Befehle, Verhalten 40  
 RSI Container 8  
 RSI Container-ID 8  
 RSI Kontext 8  
 RSI Monitor 8  
 RSI Monitor (Menüpunkt) 32  
 RSI Monitor, Bedienoberfläche 32  
 RSI Objekt 8  
 RSI Objekt-Bibliothek 8, 71  
 RSI Objekt-Parameter 8  
 RSI Objekt-Parameter, einstellen 31  
 RSI Objekt-Parameter, freischalten 31  
 RSI Visual 8  
 RSI Visual, Bedienoberfläche 29  
 RSI Visual, deinstallieren 25  
 RSI Visual, installieren 24  
 RSI\_CHECKID() 39  
 RSI\_CREATE() 35  
 RSI\_DELETE() 36  
 RSI\_DISABLE() 40  
 RSI\_ENABLE() 40  
 RSI\_GETPUBLICPAR() 38  
 RSI\_MOVECORR() 38  
 RSI\_OFF() 37  
 RSI\_ON() 37  
 RSI\_RESET() 39  
 RSI\_SETPUBLICPAR() 39

RSI.DAT 27  
 RSIERRMSG 28  
 RSITECHIDX 28

## S

Schlüsselwörter, Daten lesen 48  
 Schlüsselwörter, Daten schreiben 49  
 Schriftarten 35  
 Schulungen 7  
 SEN\_PINT (RSI Objekt) 71  
 SEN\_PREA (RSI Objekt) 71  
 Sensorbetrieb, Sicherheit 21  
 Sensorkorrektur, Funktionsweise 15  
 Sensorkorrektur, Sicherheit 21  
 Sensormodus 9, 12, 37  
 Sensortakt 9  
 Server-Programm 51  
 Server-Programm, Bedienoberfläche 52  
 Server-Programm, Kommunikationsparameter einstellen 54  
 Service, KUKA Roboter 77  
 SETDIGOUT (RSI Objekt) 75  
 Sicherheit 21  
 Sicherheitshinweise 7  
 Signalaufzeichnung, in RSI Monitor laden 34  
 Signalaufzeichnung, speichern 34  
 Signaleigenschaften, RSI Monitor 33  
 Signalfluss-Editor, öffnen 30  
 Signalfluss-Parameter, in KRL ändern 43  
 Signalfluss, in KRL-Programm einbinden 42  
 Signalfluss, Konfiguration laden 32  
 Signalfluss, Konfiguration speichern 31  
 SIGNALSWITCH (RSI Objekt) 75  
 Signalverarbeitung, Funktionsweise 12  
 Signalverlauf, anzeigen 34  
 SIN (RSI Objekt) 73  
 smarHMI 9  
 Software 23  
 Software-Endschalter 21  
 SOURCE (RSI Objekt) 71  
 Speicher, erhöhen 71  
 STATUS (RSI Objekt) 72  
 STOP (RSI Objekt) 75  
 SUM (RSI Objekt) 73  
 Support-Anfrage 77  
 Systemvoraussetzungen 23

## T

TAN (RSI Objekt) 73  
 TestServer.exe 51  
 TIMER (RSI Objekt) 74  
 TRAFO\_ROBFRAME (RSI Objekt) 72  
 TRAFO\_USERFRAME (RSI Objekt) 72  
 TTS 9

## U

UDP 9  
 Updaten, RobotSensorInterface 23

## Ü

Übersicht, RobotSensorInterface 11

Übersicht, RSI Befehle 35

## **V**

Verwendete Begriffe 8

## **X**

XML 9

XML-Datei, Ethernet-Verbindung 43

XML-Schema 47

XOR (RSI Objekt) 72

## **Z**

Zeichen 35

Zielgruppe 7





