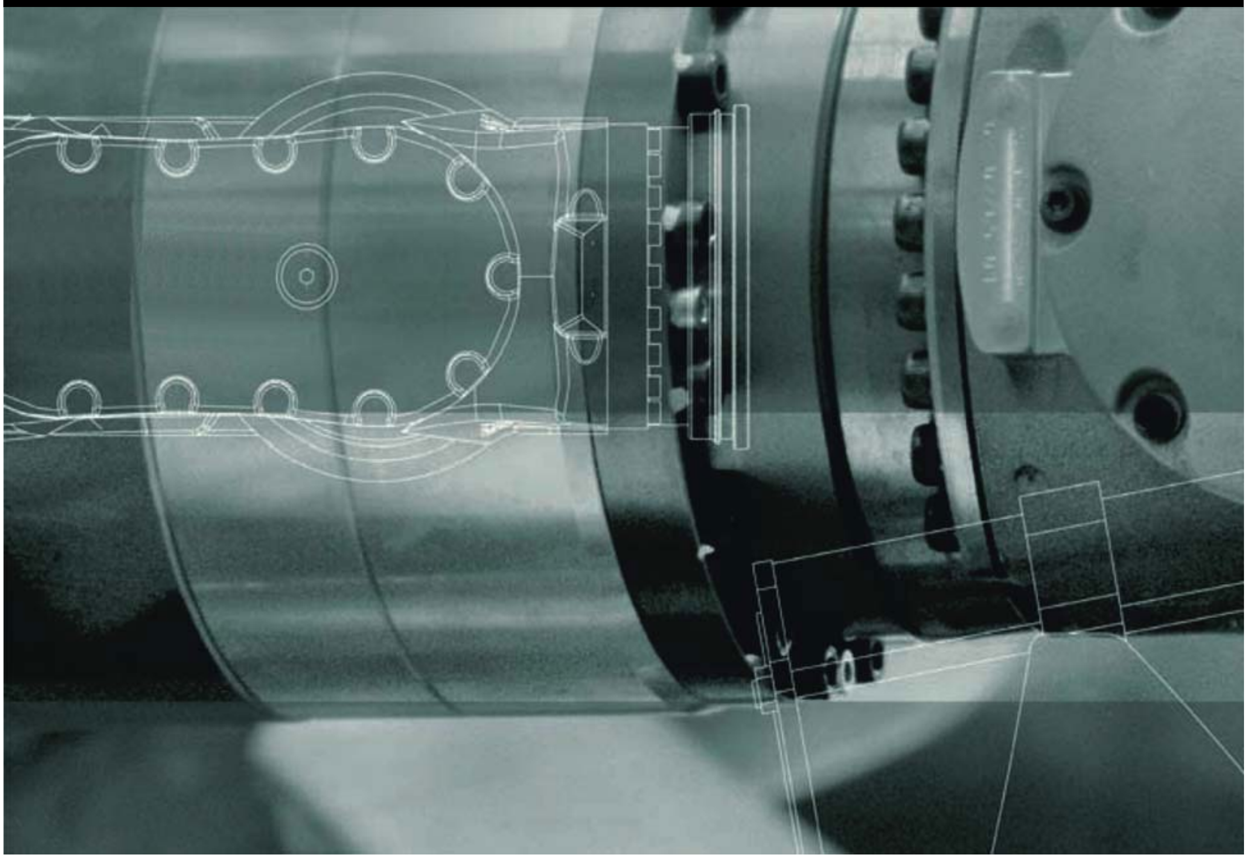
KUKA System Technology

KUKA Roboter GmbH

KUKA.ConveyorTech 5.0

For KUKA System Software 8.2



Issued: 25.04.2012

Version: KST ConveyorTech 5.0 V2 en

KUKA Roboter GmbH
Zugspitzstraße 140
D-86165 Augsburg
Germany

Other functions not described in this documentation may be operable in the controller. The user has no claims to these functions, however, in the case of a replacement or service work.

We have checked the content of this documentation for conformity with the hardware and software described. Nevertheless, discrepancies cannot be precluded, for which reason we are not able to guarantee total conformity. The information in this documentation is checked on a regular basis, however, and necessary corrections will be incorporated in the subsequent edition.

Subject to technical alterations without an effect on the function.

Translation of the original documentation

KIM-PS5-DOC

Publication:	Pub KST ConveyorTech 5.0 en
Bookstructure:	KST ConveyorTech 5.0 V2.1
Version:	KST ConveyorTech 5.0 V2 en

Contents

1 Introduction

1.1 Target group

This documentation is aimed at users with the following knowledge and skills:

- Advanced knowledge of the robot controller system
- Advanced KRL programming skills

For optimal use of our products, we recommend that our customers take part in a course of training at KUKA College. Information about the training program can be found at www.kuka.com or can be obtained directly from our subsidiaries.

1.2 Industrial robot documentation

The industrial robot documentation consists of the following parts:

- Documentation for the manipulator
- Documentation for the robot controller
- Operating and programming instructions for the KUKA System Software
- Documentation relating to options and accessories
- Parts catalog on storage medium

Each of these sets of instructions is a separate document.

1.3 Representation of warnings and notes

Safety

These warnings are relevant to safety and **must** be observed.

DANGER These warnings mean that it is certain or highly probable that death or severe physical injury **will** occur, if no precautions are taken.

WARNING These warnings mean that death or severe physical injury **may** occur, if no precautions are taken.

CAUTION These warnings mean that minor physical injuries **may** occur, if no precautions are taken.

NOTICE These warnings mean that damage to property **may** occur, if no precautions are taken.

These warnings contain references to safety-relevant information or general safety measures. These warnings do not refer to individual hazards or individual precautionary measures.

Notes

These hints serve to make your work easier or contain references to further information.

Tip to make your work easier or reference to further information.

1.4 Terms used

Term	Description
Conveyor	Moving conveyor equipment, e.g. belt, chain, etc.
ESD	Electrostatic sensitive devices
RDC	Resolver digital converter
AMI	Advanced Motion Interface
Tracking distance	Distance over which the robot tracks the workpiece on the conveyor.

2 Product description

2.1 Overview of ConveyorTech

Functions

ConveyorTech is an add-on technology package with the following functions:

- Programming of translational and rotational conveyor applications within the robot controller.
- Coordination / synchronization of robot motions with linear or circular motions of conveyor applications.
- Segmentation of conveyor operation
- Multiple conveyor tracking

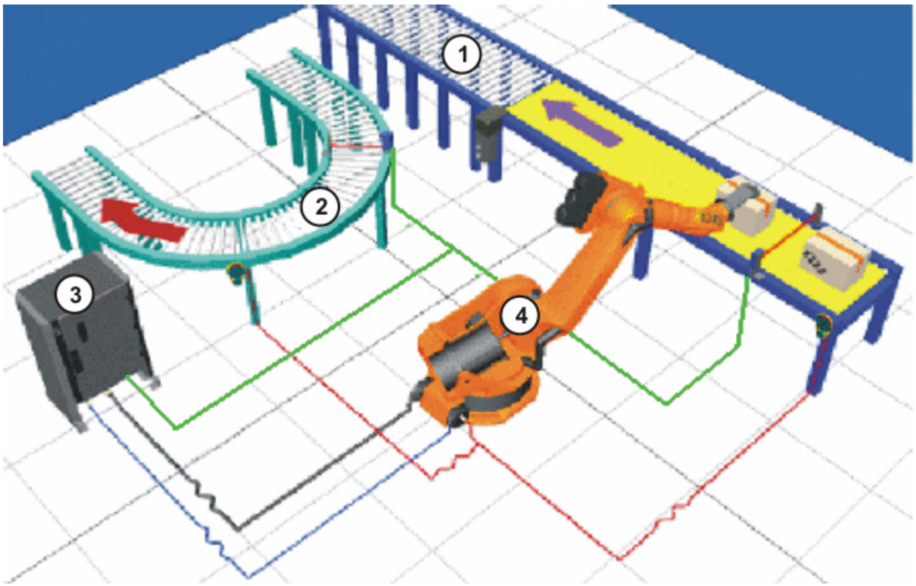


Fig. 2-1: Overview of a conveyor system

1 Linear conveyor
2 Circular conveyor
3 Robot controller
4 Robot

Areas of application

Conveyor applications enable:

- Transfer
- Machining
- Unloading
- Positioning

of workpieces within a feed system.

Communication

The resolver transfers data from the conveyor to the robot controller. The controller uses these data to calculate the transformations necessary to synchronize the robot with the conveyor.

3 Safety

This documentation contains safety instructions which refer specifically to the software described here.

The fundamental safety information for the industrial robot can be found in the “Safety” chapter of the Operating and Programming Instructions for System Integrators or the Operating and Programming Instructions for End Users.

The “Safety” chapter in the operating and programming instructions must be observed. Death to persons, severe physical injuries or considerable damage to property may otherwise result.

NOTICE The limit values for input and supply voltages must not be exceeded; individual components or complete modules may otherwise be damaged. The corresponding ESD regulations must be observed when handling modules for use in the robot controller. These modules are fitted with high-quality components and are very sensitive to electrostatic discharges (e.s.d.).

4 Planning

4.1 Layout of a conveyor system

Description

The resolver is connected to the RDC in the base frame of the robot. A synchronization switch (e.g. photo-electric barrier) is required for synchronizing the robot with the workpiece. The synchronization switch is mounted on the conveyor or on the robot and connected via the Fast Measurement input. It can also be fitted on the robot tool (e.g. gripper), for example, if different start positions are required. In this case, the application must be specially programmed accordingly.

A total of 2 conveyor systems may be operated using one robot controller.

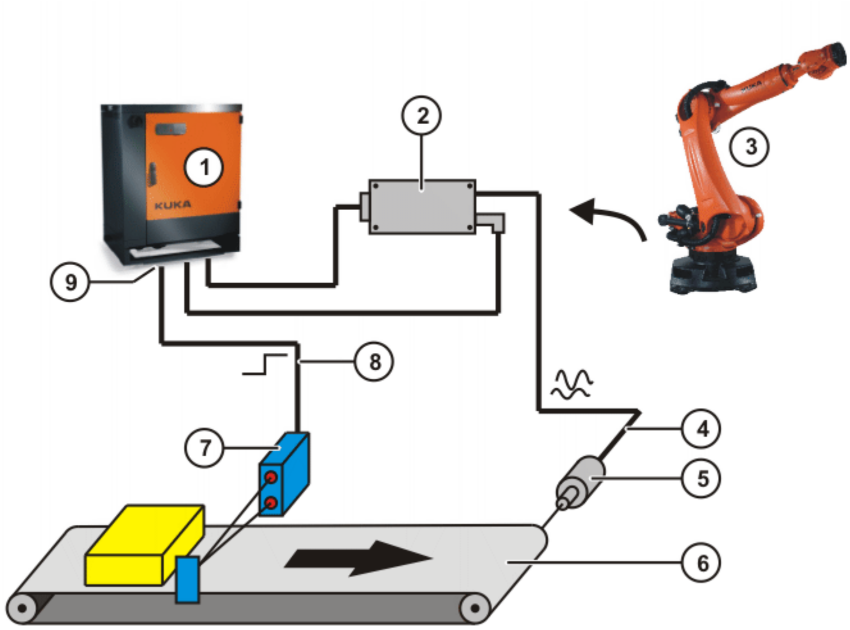


Fig. 4-1: Layout of a conveyor system

1 KR C4 controller
2 RDC
3 Robot
4 Resolver cable
5 Resolver
6 Conveyor
7 Synchronization switch
8 Cable for synchronization switch
9 “Fast measurement” option

5 Start-up and recommissioning

5.1 Start-up overview

Step	Description
1	Connect the required hardware. (>>> 5.2 "Connecting the hardware" Page 13)
2	Install the ConveyorTech software. (>>> 6.2 "Installing or updating ConveyorTech" Page 15)
3	Edit the configuration data (optional). (>>> 7.1 "Editing configuration data" Page 17)
4	Calibrate the conveyor. (>>> 8.1 "Calibrating a linear conveyor" Page 25) (>>> 8.2 "Calibrating a circular conveyor" Page 26)
5	Check the conveyor distance (optional). (>>> 8.3 "Checking the conveyor distance" Page 27)
6	Calibrate the workpiece on the conveyor (optional). (>>> 8.4 "Calibrating the workpiece on the conveyor" Page 27)

5.2 Connecting the hardware

Description

The following hardware is required:

- Resolver with cable (1 cable per conveyor)
- Synchronization switch (1 switch per conveyor)
- “Fast measurement” option

Only resolvers from KUKA Roboter GmbH may be connected to the robot’s RDC.

When designing the mechanical coupling of the resolver (e.g. defining the transmission ratio for a belt coupling), one should aim for the conveyor to cover a distance of approximately 0.03 mm per increment. The minimum acceptable degree of accuracy is provided by a distance of 0.1 mm per increment. Values over 0.1 mm per increment should be avoided.

Procedure

1. Unscrew 4 fastening screws from the housing cover of the RDC and remove the cover.

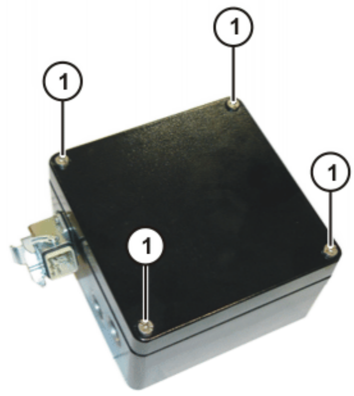


Fig. 5-1: RDC housing

1 Fastening screw on the housing cover

2. Plug the flat connector of the connecting cable into one of the free connections. The connector locks itself in place (>>> Fig. 5-2).

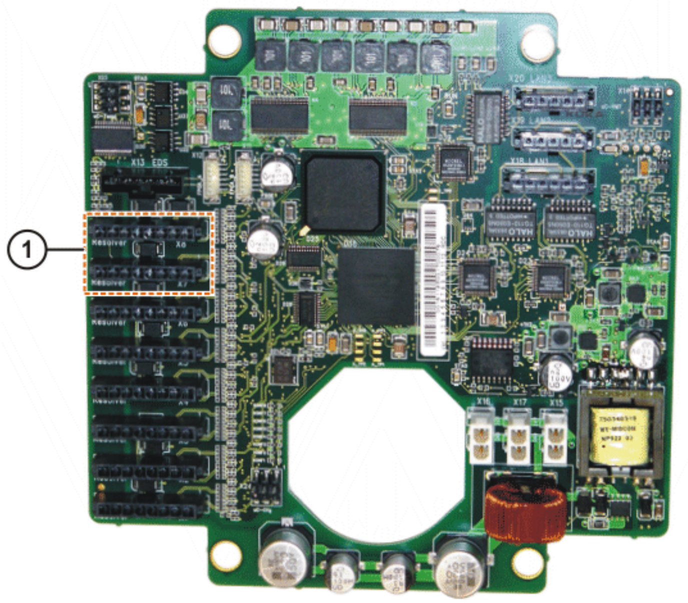


Fig. 5-2: Connecting the resolver cable to the RDC

1 Free connections on the RDC

3. Connect the other end of the connecting cable to the resolver (>>> Fig. 5-3).

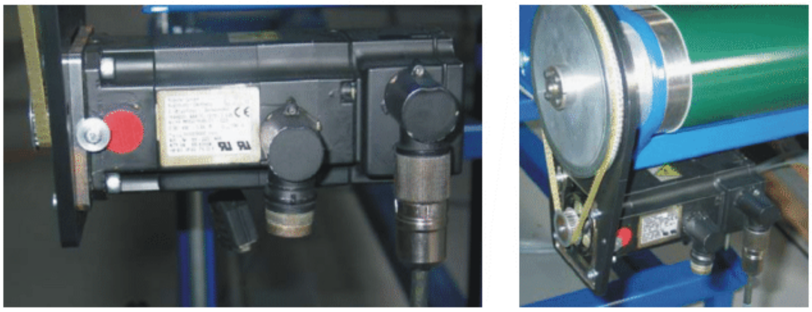


Fig. 5-3: Connecting the resolver cable to the resolver

4. Remount the housing cover and secure it with 4 fastening screws.

6 Installation

6.1 System requirements

Hardware

Robot controller:

- KR C4

Robot:

- Any robot that can be operated with a KR C4.

> If the robot is to be operated with external axes, please contact KUKA Roboter GmbH.

Conveyor system:

- Resolver connected to RDC combined with synchronization switch connected via the “Fast Measurement” option.
- A maximum of 2 conveyor systems may be operated using one robot controller.

Software

- KUKA System Software 8.2
- KUKA.UserTech

6.2 Installing or updating ConveyorTech

> It is advisable to archive all relevant data before updating a software package.

> The manually set values are saved in the file $CONFIG.DAT. If only these values are to be retained, it is sufficient to make a backup of the file $CONFIG.DAT.

> If an external axis or a sensor has been configured at the Fast Measurement input before installation, the corresponding configuration files must be adapted in order to avoid double declarations. Double declarations lead to errors when booting the controller.

Precondition

- Software on KUKA.USBData stick
- No program is selected.
- T1 or T2 operating mode
- “Expert” user group

> **NOTICE** Only the KUKA.USB data stick may be used. Data may be lost or modified if any other USB stick is used.

Procedure

1. Plug in USB stick.
2. Select **Start-up** > **Install additional software** in the main menu.
3. Press **New software**. If a software package that is on the USB stick is not displayed, press **Refresh**.
4. Mark the **Conveyor** entry and press **Install**. Reply to the request for confirmation with **Yes**. The files are copied onto the hard drive.
5. Repeat step 4 if another software package is to be installed from this stick.
6. Remove USB stick.

7. It may be necessary to reboot the controller, depending on the additional software. In this case, a corresponding prompt is displayed. Confirm with **OK** and reboot the robot controller. Installation is resumed and completed.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

6.3 Uninstalling ConveyorTech

It is advisable to archive all relevant data before uninstalling a software package.

Precondition

- "Expert" user group

Procedure

1. Select **Start-up** > **Install additional software** in the main menu. All additional programs installed are displayed.
2. Select the entry **Conveyor** and press **Uninstall**. Reply to the request for confirmation with **Yes**. Uninstallation is prepared.
3. Reboot the robot controller. Uninstallation is resumed and completed.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

7 Configuration

7.1 Editing configuration data

If two conveyor systems are used or if the default settings are not compatible with a certain configuration, the I/O configuration must be modified.

The following files must be overwritten:

- MOTIONDRV.INI
  (>>> 7.1.1 "Adapting MOTIONDRV.INI" Page 17)
- MDRCONVEYOR.INI
  (>>> 7.1.2 "Adapting MDRCONVEYOR.INI" Page 17)
- $CONFIG.DAT
  (>>> 7.1.3 "Adapting $CONFIG.DAT" Page 19)
- KRCAxes.XML
  (>>> 7.1.4 "Adapting KRCAxes.XML" Page 23)
- NextGenDriveTech.XML
  (>>> 7.1.5 "Adapting NextGenDriveTech.XML" Page 23)
- CFCore.XML
  (>>> 7.1.6 "Adapting CFCore.XML" Page 23)
- Sensor2.XML
  (>>> 7.1.7 "Creating Sensor2.XML" Page 24)

> ⚠ If the machine data are changed, the configuration files are overwritten. Changes to the files are thus lost. It is advisable not to modify the configuration files until after the machine data have been changed.

7.1.1 Adapting MOTIONDRV.INI

Precondition

- “Expert” user group

Procedure

1. Open the file MOTIONDRV.INI in the directory C:/KRC/ROBOTER/Config/User/Common/MotionDrivers.
2. Declare the conveyor name in the [BASE_DRIVER] section. If 2 conveyors are used, declare 2 conveyor names.

> ℹ The conveyor names in MOTIONDRV.INI must match the conveyor names in the files MDRCONVEYOR.INI and $CONFIG.DAT.

3. Save the file.

Example

2 conveyors are declared: CONV1 and CONV2.

```
[BASE_DRIVER]
CONV1, mdrConveyor.o
CONV2, mdrConveyor.o
```

7.1.2 Adapting MDRCONVEYOR.INI

Precondition

- “Expert” user group

Procedure

1. Open the file MDRCONVEYOR.INI in the directory C:/KRC/ROBOTER/CONFIG/User/Common/MotionDrivers.

2. If 2 conveyor systems are used, insert a section for the 2nd system beneath the section for the 1st system with the same parameters. Adapt the parameters if necessary.
3. Save the file.

Example

[CONV1] is the section for the 1st system; it is present by default after installation. [CONV2] is the section for the 2nd system.

```
[CONV1]
POSITION_TRACKER_ID=1
FILTER_LENGTH=10
DELAY_TIME=24.0
PATH_FACTOR=1.0
TRANSLATORY_VECTOR=1,0,0
ROTATORY_VECTOR=0,0,0
SEN_PREA_INDEX=1
FILTER_BASE=0
REACTION_TIME=12.0
TRACE_INDEX=0

[CONV2]
POSITION_TRACKER_ID=2
FILTER_LENGTH=10
DELAY_TIME=24.0
PATH_FACTOR=1.0
TRANSLATORY_VECTOR=1,0,0
ROTATORY_VECTOR=0,0,0
SEN_PREA_INDEX=2
FILTER_BASE=0
TRACE_INDEX=10
REACTION_TIME=12.0
```

Description

The file MDRCONVEYOR.INI contains all the relevant settings for operation. The name of the conveyor must match the name defined in MOTIONDRV.INI. After the first initialization, some of these parameters (e.g. CAL_DIG, DELAY_TIME) are reinitialized by the conveyor application. These parameters are contained as variables in the file $CONFIG.DAT.

Parameter	Description
[CONV1], [CONV2]	Name of the conveyor defined in the file MOTIONDRV.INI.
POSITION_TRACKER_ID	Index of the sensor for position sensing of the conveyor. The index must match the sensor ID configured in the file KRCAxes.XML.
FILTER_LENGTH	Number of interpolation cycles for calculating the average conveyor velocity. If higher values are used, the robot motions are smoother. This also has the effect, however, that the robot responds more slowly to changes in the velocity of the conveyor.
DELAY_TIME	Time (in ms) for compensation for the delay and reaction times inherent in the motion control. Default value: 24 ms
PATH_FACTOR	Relationship between resolver angle and conveyor motion (in mm or °).
TRANSLATORY_VECTOR	Translation vector of the conveyor BASE for calculating the tracking. The value for a linear conveyor is 1, 0, 0.
ROTATORY_VECTOR	Rotation vector of the conveyor BASE for calculating the tracking. The value for a circular conveyor is 0, 0, 1.

Parameter	Description
SEN_PREA_INDEX	Reference to the variable $SEN_PREA_C[SEN_PREA_INDEX], which contains the current distance in millimeters [mm] for a linear conveyor or the current angle [°] for a circular conveyor.
FILTER_BASE	■ **1**: The conveyor BASE is calculated with filtered position information. ■ **0**: The conveyor BASE is calculated without filtered position information.
REACTION_TIME	Minimum time between two measurements (in ms).
TRACE_INDEX	Offset to KRCIpoTraceFeature-Channel ConveyorData.

7.1.3 Adapting $CONFIG.DAT

Description

A range of preset data is inserted into the file $CONFIG.DAT during installation. These data must be adapted to the given conveyor configuration. Application data must be entered manually; geometrical measurement data of the conveyor are determined and entered by the measurement program CONV_MSR. Most variables are indexed, with the index referring to the respective conveyor number.

Precondition

- “Expert” user group

Procedure

1. Open the file $CONFIG.DAT in the directory C:/KRC/ROBOTER/KRC/R1/System.
2. If 2 conveyor systems are used, open the `CONVEYORTECH GLOBALS` fold and set the value of the variable Z_CONV_AMOUNT in the `CONV Integers` section to 2:

```
INT Z_CONV_AMOUNT=2
```

3. Enter any other modifications that are required.
4. Save the file.

7.1.3.1 Preset configuration data

Description

The configuration data are preset. For normal applications, they can be used as they are.

The name of the conveyor must match in the following files:

- MOTIONDRV.INI
- MDRCONVEYOR.INI
- $CONFIG.DAT

```
DECL CHAR BASE_NAME[32,24]
BASE_NAME[11,]="CONV1"
BASE_NAME[12,]="CONV2"
```

A separate name must be entered for each conveyor:

```
DECL MACHINE_DEF_T MACHINE_DEF[16]

MACHINE_DEF[11]={NAME[] "CONV1",COOP_KRC_INDEX 0,PARENT[]
" ",ROOT {x 0.0,y 0.0,z 0.0,a 0.0,b 0.0,c 0.0},
MECH_TYPE #CONVEYOR,GEOMETRY[] " "}

MACHINE_DEF[12]={NAME[] "CONV2",COOP_KRC_INDEX 0,PARENT[]
" ",ROOT {x 0.0,y 0.0,z 0.0,a 0.0,b 0.0,c 0.0},
MECH_TYPE #CONVEYOR,GEOMETRY[] " "}
```

Each conveyor unit corresponds to the type “#CONVEYOR”.

The entries in MACHINE_DEF[x].ROOT refer to the conveyor BASE:

```
DECL MACHINE_FRAME_T MACHINE_FRAME_DAT[32]
MACHINE_FRAME_DAT[11]={MACH_DEF_INDEX      11,PARENT[]
",GEOMETRY[] " "}
MACHINE_FRAME_DAT[12]={MACH_DEF_INDEX      12,PARENT[]
",GEOMETRY[] " "}
```

The parameter MACHINE_FRAME_DAT[x].MACH_DEF_INDEX refers to the MACHINE_DEF[x] data.

Example

BASE_DATA[11] is selected for teaching a motion of conveyor 1. Index 11 refers to MACHINE_FRAME_DAT[11].MACH_DEF_INDEX.

This index, which is also “11”, refers to the variable MACHINE_DEF [11]. In this way the interpreter recognizes that the coordinates of the taught point are those of a conveyor point, because MECH_TYPE (#CONVEYOR) refers to a conveyor.

Default settings

- Timer used by the conveyor system (default):

```
INT Z_CONV_TIMER_NBR=1
```

  This timer cannot be used by other programs.
- Interrupt number for the tracking distance alarm (default):

```
INT Z_ALARM_DIST_INT_NBR[2]
Z_ALARM_DIST_INT_NBR[1]=31
Z_ALARM_DIST_INT_NBR[2]=32
```

  This is used to generate a warning message and reset an output indicating that the robot has reached the preset alarm distance and is continuing to track a part on the conveyor.
- Interrupt number for the maximum tracking distance (default):

```
INT Z_MAX_DIST_INT_NBR[2]
Z_MAX_DIST_INT_NBR[1]=34
Z_MAX_DIST_INT_NBR[2]=35
```

  This is triggered when the robot has reached the maximum tracking distance and the tracking motion has been completed.
- Interrupt number in the case of a $STOPMESS error during the conveyor tracking motion:

```
INT Z_EMS_INT_NBR[2]
Z_EMS_INT_NBR[1]=37
Z_EMS_INT_NBR[2]=38
```

  Depending on the setting of the Boolean variable B_CONTINUE_AFTER_EMS, the conveyor motion is either aborted or continued.
- The following entry defines which variable ($SEN_PREA_C[x]) is used to monitor the conveyor area:

```
INT Z_SEN_PREA_NBR[2]
Z_SEN_PREA_NBR[1]=1
Z_SEN_PREA_NBR[2]=2
```

- The output number for an EMERGENCY STOP message (application) is defined as follows:

```
INT Z_EMS_OUT_NBR[2]
Z_EMS_OUT_NBR[1]=1
Z_EMS_OUT_NBR[2]=2
```

The fault message indicates a malfunction during execution of the conveyor application. The output is set to the value TRUE when a conveyor application is initialized.

If the variable $STOPMESS takes the value TRUE while the conveyor is active (e.g. an EMERGENCY STOP button is pressed), this output is reset.

- Definition of the output number for a conveyor alarm message:

```
INT Z_DIST_ALARM_OUT_NBR[2]
Z_DIST_ALARM_OUT_NBR[1]=4
Z_DIST_ALARM_OUT_NBR[2]=5
```

The conveyor alarm message is generated if the tracking distance of the robot is greater than that defined in the variable ALARM_CONV[x]. This output is set to TRUE during the initialization at the start of an application program and reset when the ALARM_CONV[x] distance is reached.

This variable is read when the output is set or reset.

- Boolean variable:

```
BOOL B_CONTINUE_AFTER_EMS
```

This Boolean variable is used to define whether or not the conveyor motion is to be continued following a $STOPPMESS error.

If the value is set to FALSE, a motion is executed in the subprogram CONV_QUIT and the robot is moved out of the conveyor area.

7.1.3.2 Configuration data of conveyor applications

Description

The conveyor system configuration data are preset by the setup program and must be adapted to the application in question.

> Distances are specified in millimeters for the linear conveyor and degrees for the circular conveyor.

Settings

The following settings must be made:

- Specification in millimeters [mm] or degrees [°] of the conveyor tracking distance; when this distance is reached, an alarm signal is generated:

```
REAL R_ALARM_DIST_CONV[2]
R_ALARM_DIST_CONV[1]=2000.0
R_ALARM_DIST_CONV[2]=2000.0
```

If the robot follows a part on the conveyor for the distance defined in the variable R_ALARM_DIST_CONV[x], a corresponding message is generated and the output Z_DIST_ALARM_OUT_NBR[x] is reset.

> This signal is used to indicate potentially dangerous situations.

- Specification in millimeters [mm] or degrees [°] of the maximum conveyor tracking distance:

```
REAL R_MAX_DIST_CONV[2]
R_MAX_DIST_CONV[1]=3000.0
R_MAX_DIST_CONV[2]=3000.0
```

R_MAX_DIST_CONV[x] is the maximum distance the robot can travel as a synchronized motion. As soon as the robot has reached the defined value, the synchronized motion is canceled and the robot executes the motion programmed in CONV_QUIT.

- R_SYNCH_OFF[x] is not used in this version of the software:

```
REAL R_SYNCH_OFF[2]
R_SYNCH_OFF[1]=0.0
R_SYNCH_OFF[2]=0.0
```

- R_DELAY_TIME_CONV[x] is used for compensation for the delay and reaction times inherent in the motion control.

```
R_DELAY_TIME_CONV[2]
R_DELAY_TIME_CONV[1]=24.0
R_DELAY_TIME_CONV[2]=24.0
```

Tests show that the optimal value is 24 ms.

7.1.3.3 Geometric measurement

Description

The parameters described here are preset by the setup program and must be adapted for certain applications by modifying them accordingly. These data are determined by the program CONV_MSR and require no manual modification.

CAL_DIG[x] is the calibration factor for the digital input (conveyor measurement):

```
REAL R_CAL_DIG[2]
R_CAL_DIG[1]=0.1
R_CAL_DIG[2]=0.1
```

This value determines the relationship between the number of increments supplied by the resolver/incremental encoder and the translational/rotational distance of the conveyor.

The calculation is carried out automatically when the BASE F_BASE_CONV[x] is determined.

The settings that have been made can be checked using the program CONV_MSR.

COR_DIR[x] is a variable used in the program to specify the type of conveyor used:

```
DECL TECHGEOREF COR_DIR[2]
COR_DIR[1]=#NONE
COR_DIR[2]=#NONE
```

#X = linear; #A = circular.

A BASE is defined for each conveyor using the program CONV_MSR. The results are saved in MACHINE_DEF[x].ROOT.

```
DECL MACHINE_DEF_T MACHINE_DEF[16]

MACHINE_DEF[11]={NAME[] "CONV1",COOP_KRC_INDEX 0,PARENT[]
",ROOT {x 0.0,y 0.0,z 0.0,a 0.0,b 0.0,c 0.0},
MECH_TYPE #CONVEYOR,GEOMETRY[] " "}

MACHINE_DEF[12]={NAME[] "CONV2",COOP_KRC_INDEX 0,PARENT[]
" ",ROOT {x 0.0,y 0.0,z 0.0,a 0.0,b 0.0,c 0.0},
MECH_TYPE #CONVEYOR,GEOMETRY[] " "}
```

7.1.4 Adapting KRCAxes.XML

Description If 2 conveyor systems are used, 2 lines must be added to the file KRCAxes.XML for the 2nd conveyor.

Precondition

- "Expert" user group

Procedure

1. Open the file KRCAxes.XML in the directory C:\KRC\Roboter\Config\User\Common\Mada.
2. Under the line

```
<Sensor ID="1" MeasureChannelId="1"/>
```

insert the following line:

```
<Sensor ID="2" MeasureChannelId="2"/>
```

3. Under the line

```
<MeasureChannel ID="1"/>
```

insert the following line:

```
<MeasureChannel ID="2"/>
```

4. Save the file.

7.1.5 Adapting NextGenDriveTech.XML

Description If 2 conveyor systems are used, a line must be added to the file NextGenDriveTech.XML for the 2nd conveyor.

Precondition

- "Expert" user group

Procedure

1. Open the file NextGenDriveTech.XML in the directory C:\KRC\Roboter\Config\User\Common\Mada.
2. Under the line

```
<Sensor ID="1" ConfigFile="Mada/NGAxis/Sensor1.xml>
```

insert the following line:

```
<Sensor ID="2" ConfigFile="Mada/NGAxis/Sensor2.xml>
```

3. Save the file.

7.1.6 Adapting CFCore.XML

Description If 2 conveyor systems are used, a section must be added to the file CFCore.XML for the 2nd conveyor.

Precondition

- "Expert" user group

Procedure

1. Open the file CFCore.XML in the directory C:\KRC\Roboter\Config\User\Common\Mada.
2. Under the lines

```
<!-- Achse 7 als Position Tracker -->
    <Axis Passive="true">
      <Control>
        <Controller ID="7" />
      </Control>
      <WaggonDriverInterfaces>
        <IPos Index="2" Channel="6" />
      </WaggonDriverInterfaces>
    </Axis>
```

insert the following lines:

```
<!-- Achse 8 als Position Tracker -->
    <Axis Passive="true">
      <Control>
        <Controller ID="7" />
      </Control>
      <WaggonDriverInterfaces>
        <IPos Index="2" Channel="7" />
      </WaggonDriverInterfaces>
    </Axis>
```

3. Save the file.

7.1.7 Creating Sensor2.XML

Description

If 2 conveyor systems are used, a sensor file must be created for the 2nd conveyor.

Precondition

- “Expert” user group

Procedure

1. Copy the file Sensor1.XML in the directory C:\KRC\Roboter\Config\User\Common\Mada\NGAxis.
2. Name this copy Sensor2.XML.

8 Calibration

8.1 Calibrating a linear conveyor

Description

Calibration is carried out using 3 points. These points are labeled P1, P2 and P3 in the following diagram.

The accuracy of the conveyor applications created subsequently depends on the accuracy with which these points are calibrated. Ensure that the TCP points exactly to the point in question.

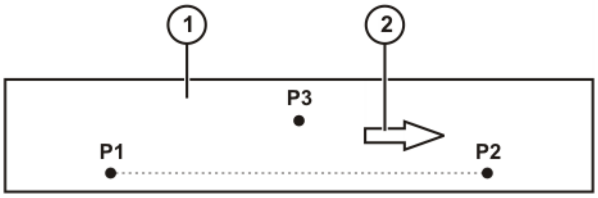


Fig. 8-1: Calibration, linear

1 Conveyor
2 Direction of travel

The point P1 can be any point on the conveyor, but should be clearly identifiable for further use. It is advisable to select a point in the start area where the robot starts to track the part. This reference point defines the origin of the conveyor BASE. All conveyor programs will use this BASE as a reference. In order to be able to repeat the calibration, it is important to define and mark this reference point in such a way that it can be used again.

Point P2 is point P1, but offset by the distance traveled by the conveyor. P2 defines the positive X axis of the conveyor BASE. It is advisable to select a position for P2 at the end of the area in which the robot can track the part on the conveyor. As large a distance as possible should be selected between P1 and P2.

Point P3 must be in the positive XY plane of the conveyor BASE. This point determines the orientation of the conveyor plane.

Precondition

- Operating mode T1

Procedure

1. Select the program CONV_MSR and start it with the Start key.
2. Select the number of the tool used (gauge pin) and confirm the entry with the Start key.
3. Select **Conveyor** and confirm the entry with the Start key.
4. Select the number of the conveyor and confirm the entry with the Start key.
5. Select the conveyor type **LINear** and confirm the selection with the Start key.
6. Place the workpiece on the conveyor and start the conveyor.
7. Once the workpiece has passed the synchronization switch and the synchronization signal has been detected, stop the conveyor.
8. Move the TCP to the origin of the conveyor BASE (P1) and confirm with the Start key.
9. Jog the conveyor.
10. Position the TCP at the offset origin (P2) and confirm with the Start key.
11. Move the TCP to any point in the positive XY plane of the conveyor BASE (P3) and confirm with the Start key.

8.2 Calibrating a circular conveyor

Description

Calibration is carried out using 3 points. These points are labeled P1, P2 and P3 in the following diagram.

> The accuracy of the conveyor applications created subsequently depends on the accuracy with which these points are calibrated. Ensure that the TCP points exactly to the point in question.

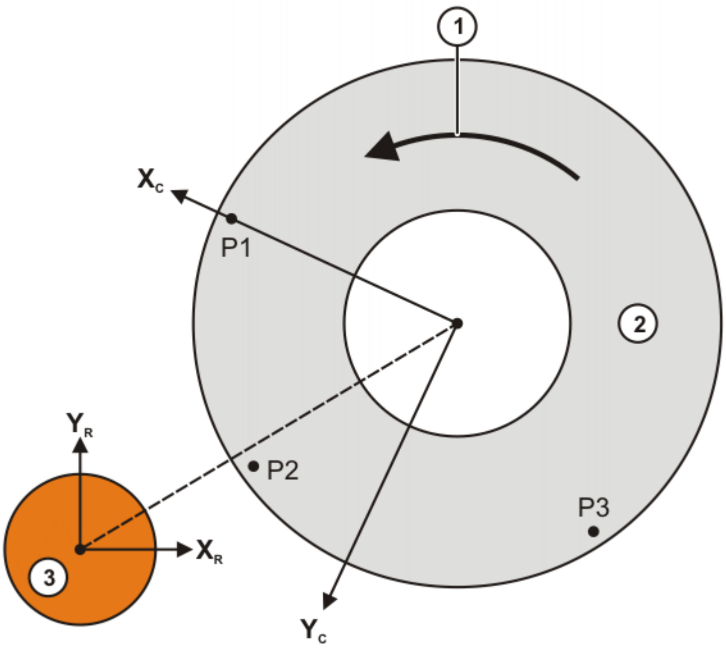


Fig. 8-2: Calibration, circular

1. Direction of travel of the conveyor
2. Conveyor
3. Robot

X_R is the X axis and Y_R is the Y axis of the ROBROOT coordinate system. X_C is the X axis and Y_C is the Y axis of the conveyor BASE (MACHINE_DEF[].ROOT).

The point P1 can be any point on the conveyor, but should be clearly identifiable for further use. It is advisable to select a point in the start area where the robot starts to track the part. This reference point is situated on the X axis of the conveyor BASE and defines the orientation of the conveyor BASE. The origin of the conveyor BASE is situated in the (virtual) pivot point of the circular conveyor. All conveyor programs will use this BASE as a reference. In order to be able to repeat the calibration, it is important to define and mark this reference point in such a way that it can be used again.

Point P2 is point P1, but offset by the distance traveled by the conveyor. P2 should be in the middle of the area in which the robot can track the part on the conveyor.

Point P3 is also point P1, but offset by the distance traveled by the conveyor after P2. P3 should be as near as possible to the end of the area in which the robot can track the part on the conveyor.

Precondition

- Operating mode T1

Procedure

1. Select the program CONV_MSR and start it with the Start key.

2. Select the number of the tool used (gauge pin) and confirm the entry with the Start key.
3. Select **Conveyor** and confirm the entry with the Start key.
4. Select the number of the conveyor and confirm the entry with the Start key.
5. Select the conveyor type **CIRCular** and confirm the selection with the Start key.
6. Place the workpiece on the conveyor and start the conveyor.
7. Once the workpiece has passed the synchronization switch and the synchronization signal has been detected, stop the conveyor.
8. Move the TCP to a point on the positive X axis of the conveyor BASE (point P1).
9. Jog the conveyor.
10. Position the TCP at the offset point P1 (P2) and confirm with the Start key. The point should be near the middle of the tracking zone.
11. Jog the conveyor.
12. Position the TCP at the further offset point P1 (P3) and confirm with the Start key. The point should be as near as possible to the end of the tracking zone.

8.3 Checking the conveyor distance

Description

Once the conveyor has been calibrated it is possible to check the measured conveyor distance. This is saved in the variable CAL_DIG[x].

Precondition

- Operating mode T1

Procedure

1. Select the program CONV_MSR and start it with the Start key.
2. Select the number of the tool used (gauge pin) and confirm the entry with the Start key.
3. Select **Calib.chk** and confirm the entry with the Start key.
4. Select the number of the conveyor and confirm the entry with the Start key.
5. Place the workpiece on the conveyor and start the conveyor.
6. Once the workpiece has passed the synchronization switch and the synchronization signal has been detected, allow the conveyor to run a little further and then stop it. The distance should be long enough to be measured easily.
7. Measure the distance between the workpiece and the synchronization point.
8. Compare the displayed value with the measured value.
9. If the value matches, answer the query with **Yes**. The program is ended.

   If the value does not match, answer the query with **No** and carry out the following steps:

   a. Select **Display** > **Variable** > **Single** in the main menu.
   b. Enter the manually measured value as the new value in the variable R_CONV_POSI.
   c. Close the input window and press the Start key. The program waits for a new synchronization signal.
   d. Repeat steps 5 to 8 until the calculated distance is correct.

8.4 Calibrating the workpiece on the conveyor

Description

Calibration of a workpiece on the conveyor is carried out using 3 points. These points are labeled P4, P5 and P6 in the following diagram.

The accuracy of the conveyor applications created subsequently depends on the accuracy with which these points are calibrated. Ensure that the TCP points exactly to the point in question.

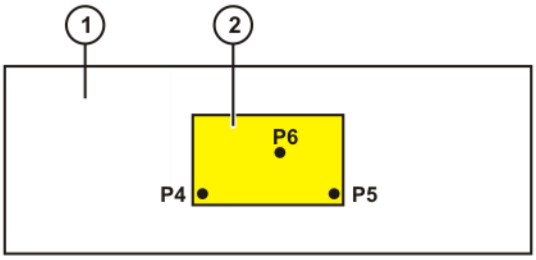


Fig. 8-3: Calibrating the workpiece on the conveyor

1 Conveyor
2 Workpiece

Point P4 is the coordinate origin of the workpiece BASE. The definition of the workpiece BASE corresponds to the default definition of BASE in the robot. It is important for the points on the workpiece to be defined unambiguously. The exact position at which the conveyor was stopped is insignificant.

Point P5 is a point on the positive X axis of the workpiece BASE.

Point P6 is a point in the positive XY plane of the workpiece BASE.

Each conveyor has just one part BASE:

- Variable for conveyor 1: BASE_DATA[11]
- Variable for conveyor 2: BASE_DATA[12]

Following calibration of the workpiece, the calculated BASE data are automatically written to the corresponding variable.

Precondition

- The conveyor is calibrated.
- Operating mode T1

Procedure

1. Select the program CONV_MSR and start it with the Start key.
2. Select the number of the tool used (gauge pin) and confirm the entry with the Start key.
3. Select **Part-BASE** and confirm the entry with the Start key.
4. Select the number of the conveyor and confirm the entry with the Start key. The selected conveyor is initialized.
5. Place the workpiece on the conveyor and start the conveyor.
6. Once the workpiece has passed the synchronization switch and the synchronization signal has been detected, allow the conveyor to run a little further and stop it at a position in which the workpiece BASE can be measured easily.
7. If the calculated distance is correct, answer the query with **Yes**. Continue with step 8.

   If the calculated distance is not correct, answer the query with **No** and modify the value (>>> 8.3 "Checking the conveyor distance" Page 27).
8. Move the TCP to the coordinate origin of the workpiece BASE (P4) and confirm with the Start key.
9. Move the TCP to a point on the positive X axis (P5) and confirm with the Start key.
10. Move the TCP to a point in the positive XY plane of the workpiece BASE (P6) and confirm with the Start key. The measurement program carries out the required calculations and then closes itself automatically.

9 Operation

9.1 Commands

The following commands are specific to this technology package:

Commands

- **ConveyorTech**
  - FOLLOW
  - QUIT
  - INI_OFF
  - ON
  - SKIP

10 Programming

10.1 Inline form “INI_OFF”

This instruction can be used to initialize a conveyor. The AMI is set to the state #INITIALIZED and the conveyor distance to 0.

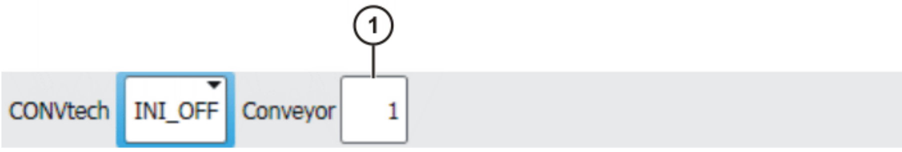


Fig. 10-1: Inline form “INI_OFF”

Item	Description
1	Conveyor number ■ **1** ■ **2**

10.2 Inline form “ON”

This instruction can be used to activate the AMI, i.e. set it to the state #ACTIVE. Following the command, the robot waits for a synchronization signal in order to carry out a motion in synchronization with the conveyor.

The conveyor offset can be detected out in the background leaving the robot controller free to perform other tasks. This allows the robot to carry out on-the-fly tracking of a part on the conveyor. The conveyor distance can be monitored using the system variable $SEN_PREA_C[].

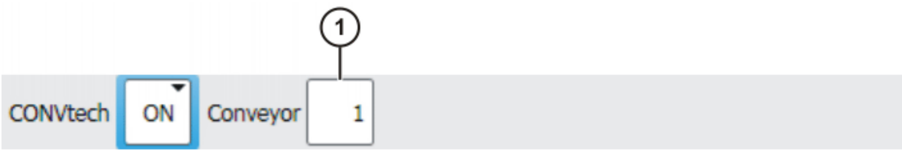


Fig. 10-2: Inline form “ON”

Item	Description
1	Conveyor number ■ **1** ■ **2**

10.3 Inline form “FOLLOW”

The instruction enables the robot to track a workpiece on a conveyor. It is possible to define an area on the conveyor in which the robot tracks the workpiece.

> ⚠ This command can only be executed if the AMI has first been activated using the command ON.

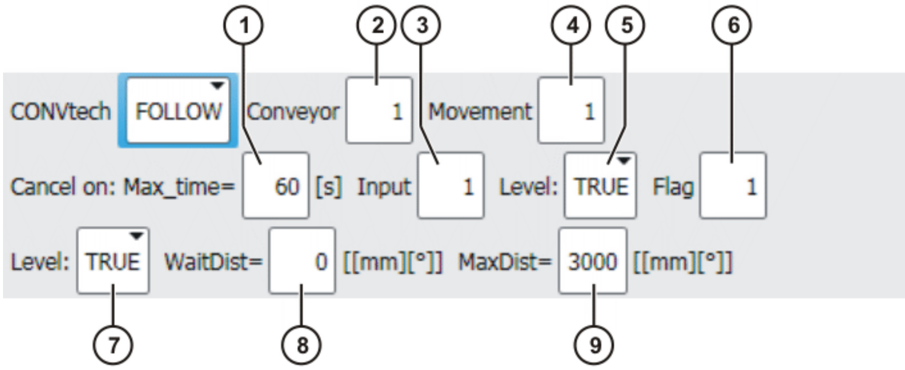


Fig. 10-3: Inline form “FOLLOW”

Item	Description
1	Maximum wait time (in seconds) It is possible to define the maximum time to wait for synchronization. Once the specified time has elapsed, execution of the command is aborted.
2	Conveyor number ■ **1** ■ **2**
3	Number of the digital input Digital input that responds to the configured logic level (item 5).
4	Number of the motion group Two different motion groups can be programmed for each conveyor.
5	Logic level ■ **TRUE** ■ **FALSE** When the specified value arises at the configured digital input (item 3), execution of the command is aborted.
6	Number of the flag Flag that responds to the configured logic level (item 7).
7	Logic level ■ **TRUE** ■ **FALSE** When the specified value arises at the configured flag (item 6), execution of the command is aborted.
8	Distance traveled by the workpiece while the robot waits before starting to track the workpiece on the conveyor. ■ For a linear conveyor: specified in millimeters ■ For a circular conveyor: specified in degrees
9	Maximum distance traveled by the workpiece before the robot cancels tracking of the workpiece on the conveyor. ■ For a linear conveyor: specified in millimeters ■ For a circular conveyor: specified in degrees

10.4 Inline form “SKIP”

This instruction can be used to define which workpieces are to be picked up, e.g. every 2nd workpiece, every 3rd workpiece, etc. A total of up to 1024 workpieces can be monitored in the background.

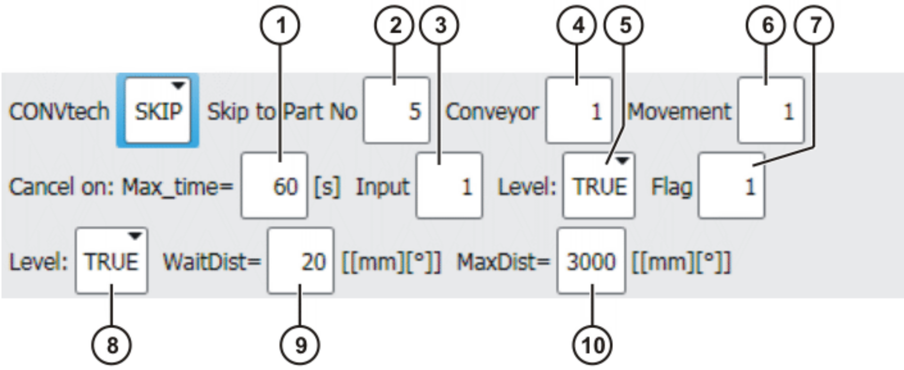


Fig. 10-4: Inline form “SKIP”

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>1</td><td>Maximum wait time (in seconds)

It is possible to define the maximum time to wait for synchronization. Once the specified time has elapsed, execution of the command is aborted.</td></tr>
<tr><td>2</td><td>The number entered specifies which workpieces are to be picked up.

Examples:
■ 1: Every workpiece is picked up.
■ 3: Every 3rd workpiece is picked up.
■ 5: Every 5th workpiece is picked up.</td></tr>
<tr><td>3</td><td>Number of the digital input

Digital input that responds to the configured logic level (item 5).</td></tr>
<tr><td>4</td><td>Conveyor number
■ 1
■ 2</td></tr>
<tr><td>5</td><td>Logic level
■ TRUE
■ FALSE

When the specified value arises at the configured digital input (item 3), execution of the command is aborted.</td></tr>
<tr><td>6</td><td>Number of the motion group

Two different motion groups can be programmed for each conveyor.</td></tr>
<tr><td>7</td><td>Number of the flag

Flag that responds to the configured logic level (item 8).</td></tr>
<tr><td>8</td><td>Logic level
■ TRUE
■ FALSE

When the specified value arises at the configured flag (item 7), execution of the command is aborted.</td></tr>
</table>

Item	Description
9	Distance traveled by the workpiece while the robot waits before starting to track the workpiece on the conveyor. ■ For a linear conveyor: specified in millimeters ■ For a circular conveyor: specified in degrees
10	Maximum distance traveled by the workpiece before the robot cancels tracking of the workpiece on the conveyor. ■ For a linear conveyor: specified in millimeters ■ For a circular conveyor: specified in degrees

10.5 Inline form “QUIT”

This instruction can be used to define whether the robot should move out of the conveyor area in the following cases:

- EMERGENCY STOP in the case of a synchronous motion
- Maximum permissible distance between robot and conveyor is reached

> This motion is not synchronized and is executed relative to the WORLD coordinate system.

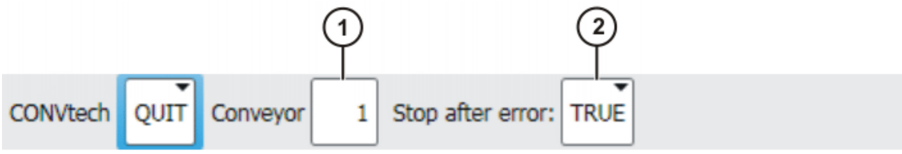


Fig. 10-5: Inline form “QUIT”

Item	Description
1	Conveyor number ■ **1** ■ **2**
2	■ **TRUE**: Program execution is stopped after the programmed QUIT motion blocks. ■ **FALSE**: Program execution is not stopped.

10.6 Creating conveyor application programs

Description

A special program structure must be used for all conveyor applications. The conveyor commands will only function within this structure which must not be altered.

Groups of synchronized motions are programmed in the subprogram CONV_MOV. 2 independent groups can be formed for each conveyor. Each conveyor is assigned its own BASE in which the motions are to be taught:

- Conveyor 1: BASE_DATA[11]
- Conveyor 2: BASE_DATA[12]

In order to update the synchronization with the conveyor in the case of BASE offsets or other calculations in KRL, a motion block that does not contain a motion relative to the BASE, e.g. LIN_REL {x 0} C_DIS, must be programmed before the first motion is executed. Without this motion block, an error message is displayed.

Motions to move the robot out of the conveyor area are programmed in the subprogram CONV_QUIT. This is necessary if the maximum tracking distance has been reached or in the case of an EMERGENCY STOP.

> The motions in the subprogram CONV_QUIT cannot be synchronized with the conveyor.

```
DEF Conveyor( )

 INI
 CONV INI

   PTP HOME  Vel= 100 % DEFAULT

 LOOP

 ENDLOOP

 END

 DEF CONV_MOV(Z_CONV_NBR:IN,Z_MOV_NBR:IN)
 INI CONV_MOV

 CONVEYOR 1 MOVEMENT GROUP 1
 CONVEYOR 1 MOVEMENT GROUP 2
 CONVEYOR 2 MOVEMENT GROUP 1
 CONVEYOR 2 MOVEMENT GROUP 2
 CONVEYOR 3 MOVEMENT GROUP 1
 CONVEYOR 3 MOVEMENT GROUP 2

 END CONV_MOV
  CONTINUE
  END

  DEF CONV_QUIT(Z_CONV_NBR:IN,B_STOP_AFTER_LEAVING:IN)
  INT Z_CONV_NBR
  BOOL B_STOP_AFTER_LEAVING
  INT Z_ADVANCE_OLD

  Z_ADVANCE_OLD=$ADVANCE
  IF (B_QUIT_BECAUSE_EMS OR B_QUIT_BECAUSE_MAX_DIST) THEN
    ;********************************************
    ;Strategy for the robot to leave the conveyor
    ;if there was an EMS or MAX_DIST reached
    ;while synchronising.
    ;********************************************
    SWITCH Z_CONV_NBR
    CASE 1
      $BASE=$NULLFRAME
      LIN_REL {Z 0.0}
    CASE 2
      $BASE=$NULLFRAME
      LIN_REL {Z 0.0}
    CASE 3
      $BASE=$NULLFRAME
      LIN_REL {Z 0.0}
    ENDSWITCH
    IF B_STOP_AFTER_LEAVING THEN
      ;****************************
      ;* Robot Stopped because of  *
      ;* error while synchronising *
      ;* EMS or MAX_DIST reached   *
      ;****************************
      LOOP
        HALT
      ENDLOOP
    ENDIF
  ENDIF
  $ADVANCE=Z_ADVANCE_OLD
 END
```

Line	Description
8-10	The conveyor commands can be inserted at this point.
14-24	Subprogram CONV_MOV
28-63	Subprogram CONV_QUIT

Precondition

- Not all folders allow the creation of new folders within them.
- “Expert” user group

Procedure

1. In the directory structure, select the folder in which the program is to be created, e.g. the folder **Program**.
2. Press **New**.
3. The **Template selection** window is opened. Select the **Conveyor** template and confirm with **OK**.
4. Enter a name for the program and confirm it with **OK**.

It is advisable to start by defining the general program structure for the application, as illustrated in the sample program (>>> 11.1 "Example basic conveyor program" Page 39).

10.7 Teaching synchronized motions

Precondition

- Operating mode T1

Procedure

1. Create application program.
2. Execute program until the robot has reached the position where it waits for synchronization.
3. Execute FOLLOW command.
4. Answer message with **Yes**. Teach mode is activated.
5. Start conveyor.
6. As soon as the workpiece has passed the synchronization switch and the signal has been received from the synchronization switch, stop the conveyor.
7. The calculated conveyor distance is displayed. Check whether the conveyor distance is correct.
8. If the conveyor distance is correct, press the **Yes** softkey. If the conveyor distance is not correct, correct the conveyor distance (>>> 8.3 "Checking the conveyor distance" Page 27). Then repeat steps 2 to 6.
9. Teach the points of the synchronized motions. In the following diagram, these points are P2, P3 and P4.

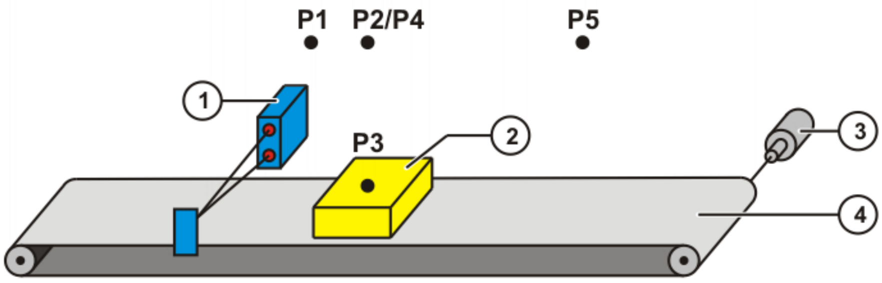


Fig. 10-6: Teaching the points of a synchronized motion

1	Synchronization switch	3	Resolver
2	Workpiece	4	Conveyor

The following diagram illustrates the synchronized motions between points P1 and P5.

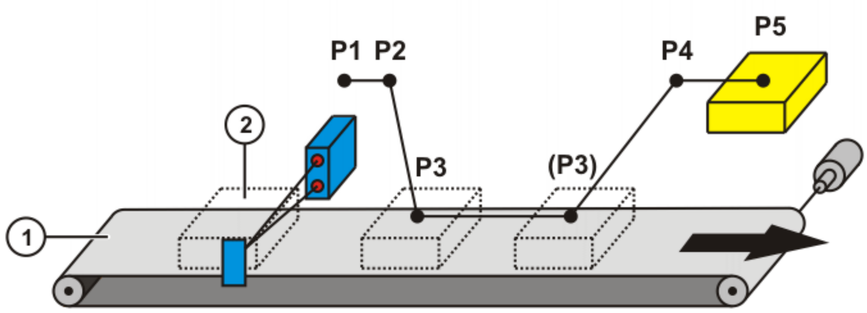


Fig. 10-7: Synchronized motion between points P1 and P5

1	Conveyor	2	Workpiece

10. Once the points have been taught, the program can be tested in T2 mode.

> **NOTICE** Program execution must not be stopped while the robot is in the conveyor area! Damage to the robot, the workpiece and other equipment may otherwise result.

> **NOTICE** Ensure that the last synchronized motion moves the robot out of the conveyor area. There may otherwise be collisions between the robot and the conveyor.

> ⚠ Synchronous motions should always be executed in AUT or AUT EXT mode with 100% program override. The robot may not otherwise be able to execute the synchronous motions completely with the specified conveyor distance.

> ℹ For test purposes, the program override ($OV_PRO) in the conveyor area can be increased to 100%. Once the robot has left the conveyor, the program override is reset to its original value. This makes it possible to conduct a test inside the conveyor area at full velocity while the remainder of the application is executed with reduced program override.

11 Example programs

11.1 Example basic conveyor program

```
 2  INI
 3  CONV INI
 4
 5  PTP HOME Vel= 100 % DEFAULT
 6  LIN P1 Vel= 2 m/s CPDAT11 Tool[1]:pen Base[0]
 7
 8  CONVtech.INI_OFF Conveyor 1
 9  CONVtech.ON Conveyor 1
10  CONVtech.FOLLOW Conveyor 1, Movement 1, Cancel on: Max_time=60s,
    Input 1, Level: TRUE, Flag 1, Level: TRUE, WaitDist=10[mm][°],
    MaxDist=100mm
11  CONVtech.QUIT Conveyor 1, Stop after error: FALSE
12
13  LIN P2 Vel= 0.5 m/s CPDAT14 Tool[1]:pen Base[0]
14  PTP HOME Vel=100 % DEFAULT
15  ________________________________________________________________
16  DEF CONV_MOV(Z_CONV_NBR:IN,Z_MOV_NBR:IN)
17  INI CONV_MOV
18
19  CONVEYOR 1 MOVEMENT GROUP 1
20  LIN pc10 Vel= 0.5 m/s CPDAT2 Tool[1]:pen Base[11]:CONV1
21  LIN pc11 Vel= 0.5 m/s CPDAT3 Tool[1]:pen Base[11]:CONV1
22  LIN pc12 Vel= 0.2 m/s CPDAT4 Tool[1]:pen Base[11]:CONV1
23  LIN pc13 Vel= 0.5 m/s CPDAT5 Tool[1]:pen Base[11]:CONV1
24
25  CONVEYOR 1 MOVEMENT GROUP 2
26
```

Description

Line	Description
6	The robot is moved from the HOME position to the conveyor wait position P1.
8	The INI_OFF command initializes the AMI.
9	The ON command sets the AMI to the state #ACTIVE.
10	The robot waits a maximum of 60 seconds for a synchronization signal.
11	The program jumps to this point if the wait time for a synchronization signal has expired or if input 1 or flag 1 is set to the level TRUE. The process is then terminated.
19	Once a synchronization signal has been received, program execution is resumed from this point.

11.2 Example program with the FOLLOW command

```
2   INI
3   CONV INI
4
5   PTP HOME Vel= 100 % DEFAULT
6   PTP P5 Vel=10 % PDAT2 Tool[1]:stift Base[0]
7
8   LOOP
9
10  PTP P7 Vel=10 % PDAT4 Tool[1]:stift Base[0]
11
12  CONVtech.INI_OFF Conveyor 1
13  CONVtech.ON Conveyor 1
14  CONVtech.FOLLOW Conveyor 1, Movement 1, Cancel on:
    Max_time=240s, Input 1, Level: TRUE, Flag 1, Level: TRUE,
    WaitDist=100[mm][°], MaxDist=1500[mm][°]
15  CONVtech.QUIT Conveyor 1, Stop after error: FALSE
16
17  ENDLOOP
_______________________________________________________________
18
19  DEF CONV_MOV(Z_CONV_NBR:IN,Z_MOV_NBR:IN)
20  INI CONV_MOV
21
22  CONVEYOR 1 MOVEMENT GROUP 1
23
24  LIN P2 Vel=0.3 m/s CPDAT1 Tool[1]:stift Base[11]:CONV1
25  LIN P3 Vel=0.3 m/s CPDAT2 Tool[1]:stift Base[11]:CONV1
26
27  CONVEYOR 1 MOVEMENT GROUP 2
28
```

Description

Line	Description
10	The robot is moved to the conveyor wait position P7.
12	The INI_OFF command initializes the AMI. **Note**: If the FOLLOW command is used within the LOOP, the INI_OFF command must also be located within the LOOP.
13	The ON command sets the AMI to the state #ACTIVE. **Note**: If the FOLLOW command is used within the LOOP, the ON command must also be located within the LOOP.
14	The robot waits a maximum of 240 seconds for a synchronization signal.
15	The program jumps to this point if the wait time for a synchronization signal has expired or if input 1 or flag 1 is set to the level TRUE. The process is then terminated.
22	Once a synchronization signal has been received, program execution is resumed from this point.

11.3 Example program with the SKIP command

```
2   INI
3   CONV INI
4
5   PTP HOME Vel= 100 % DEFAULT
6   PTP P5 Vel=10 % PDAT2 Tool[1]:stift Base[0]
7
8   CONVtech.INI_OFF Conveyor 1
9   CONVtech.ON Conveyor 1
10  CONVtech.FOLLOW Conveyor 1, Movement 1, Cancel on:
    Max_time=240s, Input 1, Level: TRUE, Flag 1, Level: TRUE,
    WaitDist=100[mm][°], MaxDist=1900[mm][°]
11
12  LOOP
13
14  PTP p5 Vel=10 % PDAT3 Tool[1]:stift Base[0]
15
16  CONVtech.SKIP Skip to Part No 2, Conveyor 1, Movement 1,
    Cancel on: Max_time=240s, Input 1, Level:TRUE, Flag 1,
    Level:TRUE, WaitDist=100[mm][°], MaxDist=1900[mm][°]
17  CONVtech.QUIT Conveyor 1, Stop after error: FALSE
18
19  ENDLOOP
_________________________________________________________________
20
21  DEF CONV_MOV(Z_CONV_NBR:IN,Z_MOV_NBR:IN)
22  INI CONV_MOV
23
24  CONVEYOR 1 MOVEMENT GROUP 1
25
26  LIN P2 Vel=0.3 m/s CPDAT1 Tool[1]:stift Base[11]:CONV1
27
```

Description

Line	Description
6	The robot is moved from the HOME position to the conveyor wait position P5.
8	The INI_OFF command initializes the AMI. **Note**: If the SKIP command is used, the INI_OFF command must always be located outside the LOOP.
9	The ON command sets the AMI to the state #ACTIVE. **Note**: If the SKIP command is used, the ON command must always be located outside the LOOP.
10	The FOLLOW command is called. **Note**: If the SKIP command is used, the FOLLOW-1 command must be called once outside the LOOP.
16	The robot waits a maximum of 240 seconds for a synchronization signal.
17	The program jumps to this point if the wait time for a synchronization signal has expired or if input 1 or flag 1 is set to the level TRUE. The process is then terminated.
24	Once a synchronization signal has been received, program execution is resumed from this point.

12 Troubleshooting

> ⚠ **WARNING** A robot that is synchronized with a conveyor is controlled not only by the robot controller, but also partially by the conveyor.
> When tracking a part on a conveyor, the robot executes Cartesian motions in a BASE coordinate system that moves with the conveyor.
> In some cases, this can result in hazardous situations if the interface between the robot controller and the conveyor controller has not been designed correctly.

12.1 Robot leaves the path in external mode EXT

> ⚠ **WARNING** If a robot that is tracking a part on a conveyor is stopped, the robot leaves the path relative to the conveyor.
> In most cases, this is due to the fact that a moving conveyor cannot stop as quickly as a robot. In the external mode EXT, in particular, this can lead to hazardous situations.

Example

1. A robot tracks a part on a conveyor.
2. An EMERGENCY STOP is triggered in the robot cell.
3. The robot stops.
4. Under normal circumstances, the conveyor also stops.

   The robot has probably left the path because the conveyor has stopped with a slight delay.
5. The EMERGENCY STOP is reset.
6. The conveyor starts up again and the part moves away from the robot.
7. An external start of the robot is triggered.
8. The robot attempts to track the part on the conveyor.
9. Range limits and workspaces are ineffective.
10. The robot collides with the conveyor.

Possible remedy

- Prevent restart of the robot following a synchronization error. It is possible to cancel the robot program in the submit interpreter SPS.SUB.

> ⚠ Never restart the conveyor if the robot has left the path ($ON_PATH) and has not been restarted ($PRO_ACT).

12.2 Modifying the program override

Description

Reducing the program override results in the robot spending more time tracking a part on a conveyor.

If the robot tracks the part for a longer time, it also tracks it over a greater distance. This may result in collisions when range limits are reached.

Possible remedy

- Ensure that the program override is set to 100% before synchronization is started.

  Example:

```
$OV_PRO=100
ConvTech.FOLLOW...
```

- Ensure that robot and conveyor are stopped when range limits are reached.

  Range limits can be defined with the following parameters:

Parameter	Description
R_AlarmDist_Conv[x]	As soon as the robot reaches this distance, an output is set. This output can force the conveyor controller (PLC) to stop the conveyor in order to prevent the robot from moving too far.
R_MaxDist_Conv[x]	As soon as the robot reaches this preset distance, tracking of the part is aborted. It is possible to move the robot out of the conveyor area automatically (ConvQuit). **Note**: These motions are not synchronized with the conveyor. Ensure that there is no collision with the conveyor.

- Set up workspaces. This enables the robot controller to respond as soon as the robot enters a workspace (e.g. safety fence, machine).

12.3 The decision to track a part is made in the advance run

Description

In the inline form “FOLLOW”, an area is defined which determines whether the robot is to track a part on the conveyor. If the part is within both defined range limits, tracking of the part is enabled.

Since the robot can be synchronized with and track the part on the conveyor on the fly, decisions are also made on the fly.

Example

1. A robot is working on side A and wants to track a part on a conveyor on side B of the robot cell.

   For this, a 1 s motion towards the conveyor on side B is required.
2. The conveyor moves at a velocity of 1 m/s.

   The range limits defined for part tracking are 0 mm and 500 mm.
3. The robot leaves side A and already evaluates whether it can track a part on conveyor B.
4. The robot decides to track the part and moves to side B.
5. After 1 s, the robot is synchronized on the fly with the part on conveyor B, which has meanwhile moved forwards 1 m, and tracks it.
6. This means that the robot is tracking a part that has already left the part tracking range.

Possible remedy

- No on-the-fly synchronization with parts on the conveyor. Ensure that no advance run is defined ($ADVANCE=0) and that the wait position before synchronization is programmed without approximate positioning.
- If on-the-fly synchronization is required, ensure that the advance run is reduced to a minimum ($ADVANCE=1). Teach multiple positions at the start of the conveyor. Since there is more than one position, the decision about part tracking is delayed.

12.4 Connection to position tracker broken

Description

If the electrical connection to a position tracker (e.g. resolver on the conveyor) is broken, the robot receives no new position values of the conveyor. The robot thus fails to detect that the conveyor is moving. This can result in the robot colliding with the workpiece.

The robot controller generates an error message, but this does not trigger a stop:

Error 395: Encoder cable defective, Position Tracker at DSEx

If the connection is broken while the robot is tracking the workpiece on the conveyor, the robot does not track the workpiece for the remaining distance.

Possible remedy

- Respond to the signal SIGNAL $POS_TRACKER_ERROR FALSE in the file $MACHINE.DAT.

  Example: Set the signal at an output:

```
SIGNAL $POS_TRACKER_ERROR $OUT[120]
```

13 KUKA Service

13.1 Requesting support

Introduction

The KUKA Roboter GmbH documentation offers information on operation and provides assistance with troubleshooting. For further assistance, please contact your local KUKA subsidiary.

Information

The following information is required for processing a support request:

- Model and serial number of the robot
- Model and serial number of the controller
- Model and serial number of the linear unit (if applicable)
- Model and serial number of the linear unit (if applicable)
- Version of the KUKA System Software
- Optional software or modifications
- Archive of the software

  For KUKA System Software V8: instead of a conventional archive, generate the special data package for fault analysis (via **KrcDiag**).
- Application used
- Any external axes used
- Description of the problem, duration and frequency of the fault

13.2 KUKA Customer Support

Availability

KUKA Customer Support is available in many countries. Please do not hesitate to contact us if you have any questions.

Argentina

Ruben Costantini S.A. (Agency)
Luis Angel Huergo 13 20
Parque Industrial
2400 San Francisco (CBA)
Argentina
Tel. +54 3564 421033
Fax +54 3564 428877
ventas@costantini-sa.com

Australia

Headland Machinery Pty. Ltd.
Victoria (Head Office & Showroom)
95 Highbury Road
Burwood
Victoria 31 25
Australia
Tel. +61 3 9244-3500
Fax +61 3 9244-3501
vic@headland.com.au
www.headland.com.au

Belgium	KUKA Automatisering + Robots N.V. Centrum Zuid 1031 3530 Houthalen Belgium Tel. +32 11 516160 Fax +32 11 526794 info@kuka.be www.kuka.be
Brazil	KUKA Roboter do Brasil Ltda. Avenida Franz Liszt, 80 Parque Novo Mundo Jd. Guançã CEP 02151 900 São Paulo SP Brazil Tel. +55 11 69844900 Fax +55 11 62017883 info@kuka-roboter.com.br
Chile	Robotec S.A. (Agency) Santiago de Chile Chile Tel. +56 2 331-5951 Fax +56 2 331-5952 robotec@robotec.cl www.robotec.cl
China	KUKA Automation Equipment (Shanghai) Co., Ltd. Songjiang Industrial Zone No. 388 Minshen Road 201612 Shanghai China Tel. +86 21 6787-1808 Fax +86 21 6787-1805 info@kuka-sha.com.cn www.kuka.cn
Germany	KUKA Roboter GmbH Zugspitzstr. 140 86165 Augsburg Germany Tel. +49 821 797-4000 Fax +49 821 797-1616 info@kuka-roboter.de www.kuka-roboter.de

France

KUKA Automatisme + Robotique SAS
Techvallée
6, Avenue du Parc
91140 Villebon S/Yvette
France
Tel. +33 1 6931660-0
Fax +33 1 6931660-1
commercial@kuka.fr
www.kuka.fr

India

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

Italy

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

Japan

KUKA Robotics Japan K.K.
Daiba Garden City Building 1F
2-3-5 Daiba, Minato-ku
Tokyo
135-0091
Japan
Tel. +81 3 6380-7311
Fax +81 3 6380-7312
info@kuka.co.jp

Korea

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
info@kukakorea.com

Malaysia	KUKA Robot Automation Sdn Bhd South East Asia Regional Office No. 24, Jalan TPP 1/10 Taman Industri Puchong 47100 Puchong Selangor Malaysia Tel. +60 3 8061-0613 or -0614 Fax +60 3 8061-7386 info@kuka.com.my
Mexico	KUKA de Mexico S. de R.L. de C.V. Rio San Joaquin #339, Local 5 Colonia Pensil Sur C.P. 11490 Mexico D.F. Mexico Tel. +52 55 5203-8407 Fax +52 55 5203-8148 info@kuka.com.mx
Norway	KUKA Sveiseanlegg + Roboter Sentrumsvegen 5 2867 Hov Norway Tel. +47 61 18 91 30 Fax +47 61 18 62 00 info@kuka.no
Austria	KUKA Roboter Austria GmbH Vertriebsbüro Österreich Regensburger Strasse 9/1 4020 Linz Austria Tel. +43 732 784752 Fax +43 732 793880 office@kuka-roboter.at www.kuka-roboter.at
Poland	KUKA Roboter Austria GmbH Spółka z ograniczoną odpowiedzialnością Oddział w Polsce Ul. Porcelanowa 10 40-246 Katowice Poland 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

Russia

OOO KUKA Robotics Rus
Webnaja ul. 8A
107143 Moskau
Russia
Tel. +7 495 781-31-20
Fax +7 495 781-31-19
kuka-robotics.ru

Sweden

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

Switzerland

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

Spain

KUKA Robots IBÉRICA, S.A.
Pol. Industrial
Torrent de la Pastera
Carrer del Bages s/n
08800 Vilanova i la Geltrú (Barcelona)
Spain
Tel. +34 93 8142-353
Fax +34 93 8142-950
Comercial@kuka-e.com
www.kuka-e.com

South Africa	Jendamark Automation LTD (Agency) 76a York Road North End 6000 Port Elizabeth South Africa Tel. +27 41 391 4700 Fax +27 41 373 3869 www.jendamark.co.za
Taiwan	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
Thailand	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
Czech Republic	KUKA Roboter Austria GmbH Organisation Tschechien und Slowakei Sezemická 2757/2 193 00 Praha Horní Počernice Czech Republic Tel. +420 22 62 12 27 2 Fax +420 22 62 12 27 0 support@kuka.cz
Hungary	KUKA Robotics Hungaria Kft. Fö út 140 2335 Taksony Hungary Tel. +36 24 501609 Fax +36 24 477031 info@kuka-robotics.hu

USA

KUKA Robotics Corp.
22500 Key Drive
Clinton Township
48036
Michigan
USA
Tel. +1 866 8735852
Fax +1 586 5692087
info@kukarobotics.com
www.kukarobotics.com

UK

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

Index