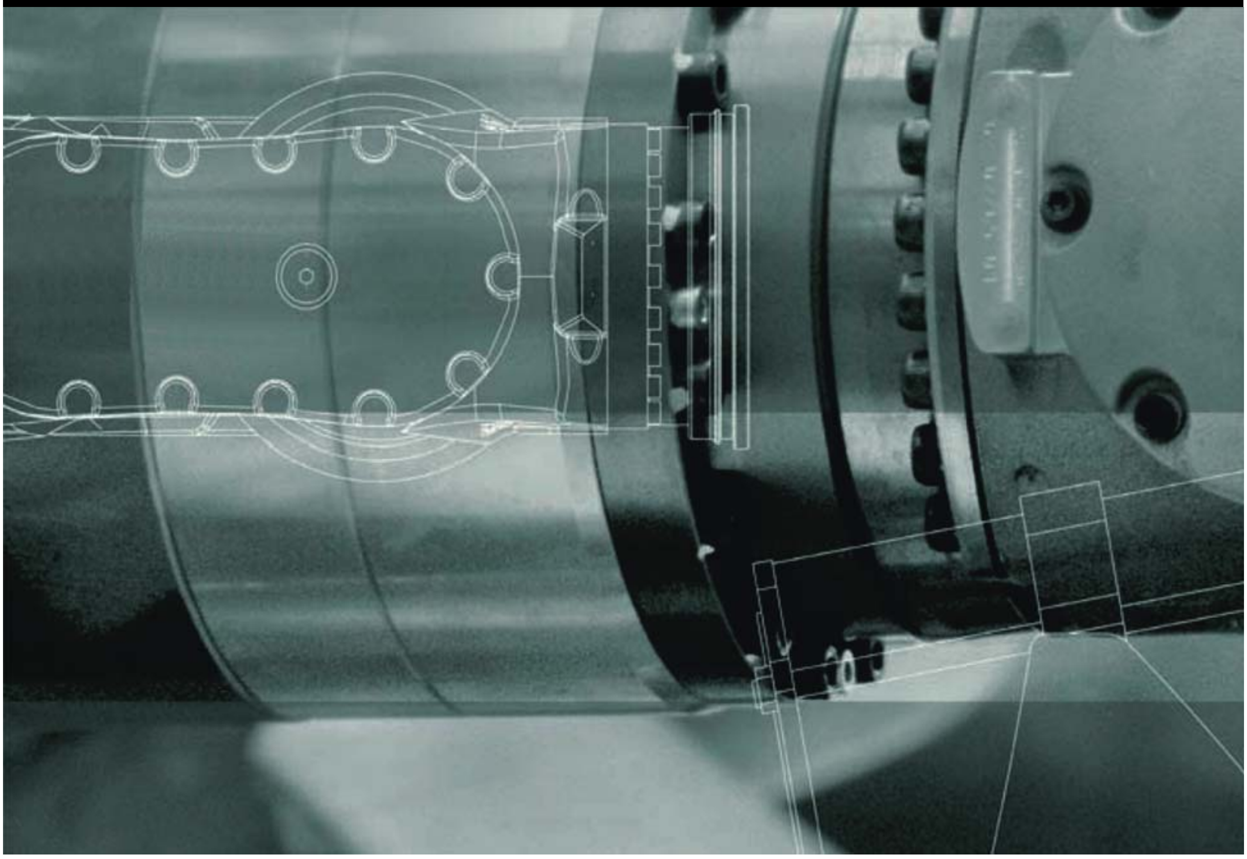
KUKA System Technology

KUKA Roboter GmbH

KUKA.Ethernet KRL XML 1.2

For KUKA System Software 5.2, 5.3, 5.4, 5.5, 5.6 and 7.0



Issued: 16.06.2009

Version: KST Ethernet KRL XML 1.2 V1 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 operating instructions

KIM-PS5-DOC

Publication: Pub KST Ethernet KRL XML 1.2 en
Book structure: KST Ethernet KRL XML 1.2 V1.1
Label: KST Ethernet KRL XML 1.2 V1

Contents

1 Introduction

1.1 Target group

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

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

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 Robot system documentation

The robot system documentation consists of the following parts:

- Operating instructions for the robot
- Operating instructions for the robot controller
- Operating and programming instructions for the KUKA System Software
- Documentation relating to options and accessories

Each of these sets of instructions is a separate document.

1.3 Trademarks

Windows is a trademark of Microsoft Corporation.

Suse Linux is a trademark of Linus Torvalds.

1.4 Terms used

The following terms are used in this documentation:

Term	Description
API	Application Programming Interface (API) is an interface for programming applications. API enables the integration of other programs into the software system.
Ethernet	Ethernet is a data network technology for local area networks (LANs). It allows the exchange of data between the connected devices in the form of data frames.
Router	A router is a hardware or software component that routes incoming data in a network to a target network or subnetwork.
Topology	In a computer network, topology refers to the structure of the connections between the individual computers.
Parser	A parser is a program that syntactically interprets textual components of a document and replaces them with commands or codes.

Term	Description
TCP/IP	The Transmission Control Protocol (TCP) is a protocol for data exchange between the devices in a network. TCP constitutes a virtual channel between 2 sockets in a network connection. Data can be transmitted along this channel in both directions. The Internet Protocol (IP) has the task of transporting data packets via a number of networks from a transmitter to a receiver.
Well formed	An XML file is well formed if it is structured according to the W3C standards.
XML	The Extensible Markup Language (XML) is a standard for creating machine- and human-readable documents in the form of a specified tree structure.

2 Product description

2.1 Overview of KUKA.Ethernet KRL XML

KUKA.Ethernet KRL XML is an add-on technology package with the following functions:

Functions

- Data transmission from the robot controller to an external system
- Data transmission from an external system to the robot controller
- Sending and receiving of data within a KRL program

Characteristics

- Up to 9 external systems can be connected to the robot controller.
- The robot controller is always a client which connects to a server (external system).
- The received data are saved in a buffer so that no data can be lost.
- An incoming XML tag can directly set a controller output.

Communication

The data are transmitted via the Ethernet TCP/IP protocol as XML strings. Communication within the interpolation cycle cannot be implemented.

Components

- KUKA.Ethernet KRL XML
- KUKA Router

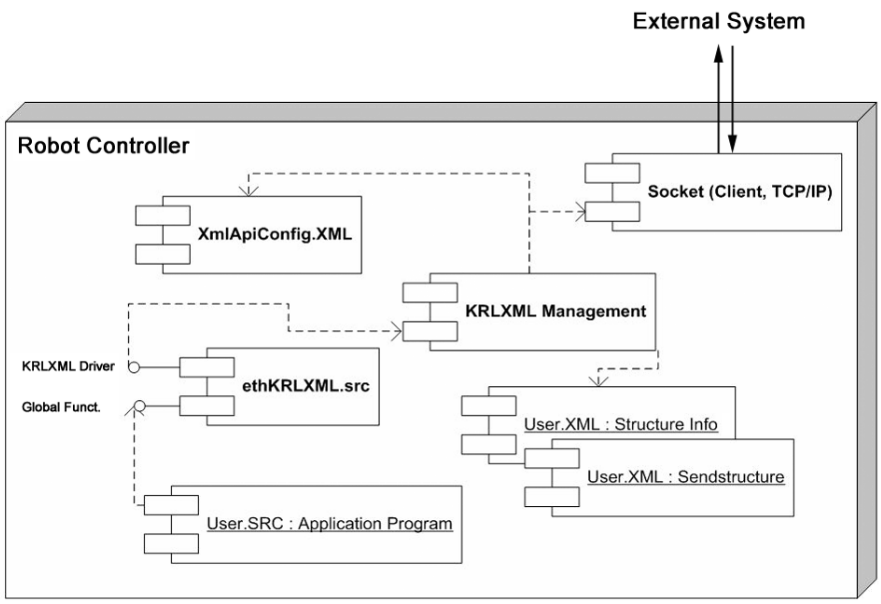


Fig. 2-1: KRL XML components

Components	Description
Socket (Client, TCP/IP)	Interface for data exchange with the external system
KRLXML Management	Kernel system
ethKRLXML.src	Source file containing all the user functions.

Components	Description
XmlApiConfig.xml	File in which the communication parameters are defined: ■ Device name ■ IP address ■ Port number (>>> 5.5 "Configuring communication parameters" page 46)
User.XML	File in which the XML structure of the data to be transmitted is defined: ■ Structure Info: XML structure for receiving data ■ Sendstructure: XML structure for sending data The file name depends on the device name set in XmlApiConfig.xml.
User.SRC	Application program that uses the XML API to exchange data.

2.2 Functional principle

2.2.1 Data transmission

Description

Data transmission functions as follows:

1. The external system evaluates the sensor data and determines the data to be sent.
2. The data to be sent are packed into an XML string.
3. The XML string is transmitted via the Ethernet TCP/IP protocol to the robot controller.
4. The XML parser contained in KUKA.Ethernet KRL XML extracts the data from the XML string.
5. These data are stored in an intermediate buffer.
6. KUKA.Ethernet KRL XML functions then enable access to these data from the KRL program (>>> 6.2.3 "Access functions" page 54).

The driver layer carries out the communication between the external system and the robot controller using Ethernet TCP/IP. The format of the sent and received data is checked in the XML layer and sorted into the buffer. KUKA.Ethernet KRL XML provides the KRL functions necessary for the application programming interface (API).

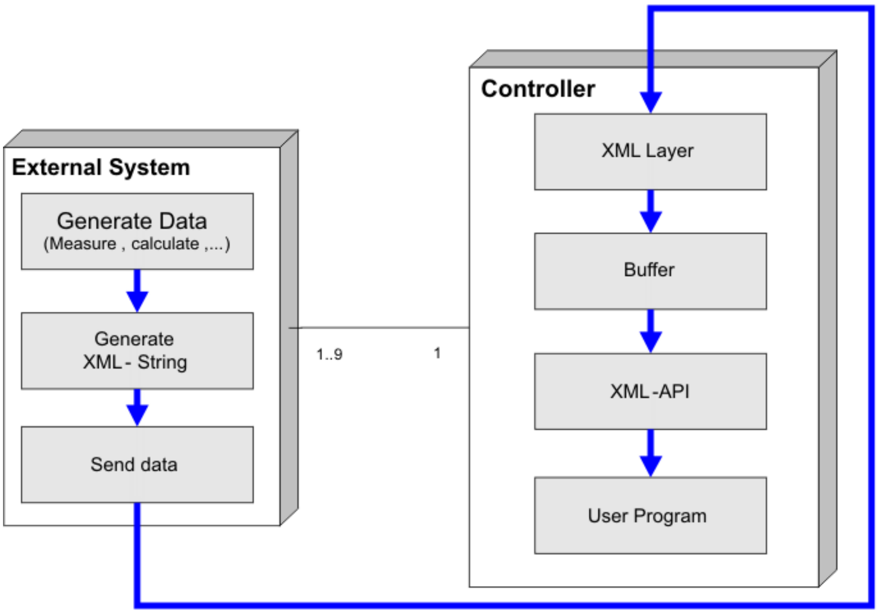


Fig. 2-2: Data transmission sequence

2.2.2 Data processing

Description

The driver layer has the task of ensuring data processing and communication with the external system.

This involves the following steps:

1. Receive data from the interface (Ethernet).
2. Check data for XML conformity (well-formedness). Violation of conformity, e.g. due to defective transmission, is detected here and flagged as an error.
3. Sort data. Every element must be copied into the appropriate buffer. They are held there for further processing.
4. The access functions (>>> 6.2.3 "Access functions" page 54) can be used to access the individual buffers. The value read from the buffer is copied to a KRL variable. This ensures, for example, that an XML string is reliably copied to a KRL string.

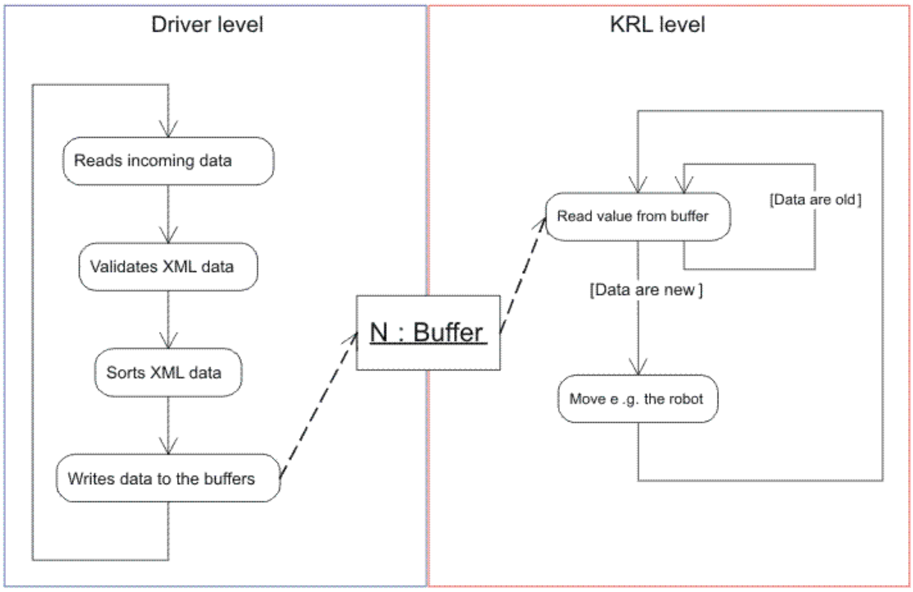


Fig. 2-3: Functionality of the buffer (KRL)

The buffers are arranged as ring memories. Each buffer comprises n memory slots for one specific data type. The number and size of the buffers are defined in a structure file (>>> 5.6 "Defining an XML structure for receiving data" page 47).

Valid data types:

Data type	Range of values
STRUCTTAG	Contains elements or attributes itself
BOOLEAN	TRUE, FALSE
INTEGER	-2147483648 to 2147483647
REAL	-3.4E+38 to +3.4E+38
STRING	Maximum 80 characters
FRAME	Comprises 6 REAL values Syntax: <FrameName XPos="REAL" YPos="REAL" ZPos="REAL" ARot="REAL" BRot="REAL" CRot="REAL"/> ■ The frame name is freely selectable. ■ The attribute names XPos to CRot must be used as shown.

Example

The incoming data are stored in sequence in the buffer. 3 positions with X and Y coordinates are transmitted one after another.

Element notation:

```
<Pos>
   <XPos>0.1</XPos>
   <YPos>0.8</YPos>
</Pos>
<Pos>
   <XPos>0.2</XPos>
   <YPos>0.7</YPos>
</Pos>
<Pos>
   <XPos>0.3</XPos>
   <YPos>0.5</YPos>
</Pos>
```

Attribute notation:

```
<Pos XPos="0.1" YPos="0.8"/>
<Pos XPos="0.2" YPos="0.7"/>
<Pos XPos="0.3" YPos="0.5"/>
```

The end tag "/>" in the attribute notation is analogous to the end tag "</Pos>" in the element notation.

3 buffers are required to enable this data sequence to be stored in the memory.

Buffer	Data type
Pos	STRUCTTAG
Pos.XPos	REAL
Pos.YPos	REAL

The principle used for reading the data from the buffer is defined by means of a parameter in the access functions (>>> 6.2.3 "Access functions" page 54):

Buffer reading principle

- First In First Out

  The values are read from the buffer in the order in which they were saved there.

  Result for XPos:

```
XPos = 0.1
XPos = 0.2
XPos = 0.3
```

- Last In First Out

  The most recently saved value is read from the buffer.

  Result for XPos:

```
XPos=0.3
```

3 Safety

3.1 General

3.1.1 Liability

The device described in these operating instructions is an industrial robot – called “robot system” in the following text – consisting of:

- Robot
- Connecting cables
- Robot controller
- Teach pendant
- Linear unit (optional)
- Positioner (optional)
- Two-axis positioner (optional)
- Top-mounted cabinet (optional)

The robot system is built using state-of-the-art technology and in accordance with the recognized safety rules. Nevertheless, impermissible misuse of the robot system may constitute a risk to life and limb or cause damage to the robot system and to other material property.

The robot system may only be used in perfect technical condition in accordance with its designated use and only by safety-conscious persons who are fully aware of the risks involved in its operation. Use of the robot system is subject to compliance with these operating instructions and with the declaration of incorporation supplied together with the robot system. Any functional disorders affecting the safety of the system must be rectified immediately.

Safety information

Safety information cannot be held against KUKA Roboter GmbH. Even if all safety instructions are followed, this is not a guarantee that the robot system will not cause personal injuries or material damage.

No modifications may be carried out to the robot system without the authorization of KUKA Roboter GmbH. Additional components (tools, software, etc.), not supplied by KUKA Roboter GmbH, may be integrated into the robot system. The user is liable for any damage these components may cause to the robot system or to other material property.

In addition to the Safety chapter, the operating instructions contain further safety instructions. These must be observed.

3.1.2 Representation of warnings and notes

Safety

Warnings marked with this pictogram are relevant to safety and **must** be observed.

Danger!
This warning means that death, severe physical injury or substantial material damage **will** occur, if no precautions are taken.

Warning!
This warning means that death, severe physical injury or substantial material damage **may** occur, if no precautions are taken.

Caution!
This warning means that minor physical injuries or minor material damage **may** occur, if no precautions are taken.

Notes

Notes marked with this pictogram contain tips to make your work easier or references to further information.

Tips to make your work easier or references to further information.

3.1.3 Designated use of the robot system

The robot system is designed exclusively for the specified applications.

Further information is contained in the technical data of the operating instructions for the robot system and its options.

Using the system for any other or additional purpose is considered impermissible misuse. The manufacturer cannot be held liable for any damage resulting from such use. The risk lies entirely with the user.

Operating the robot system and its options within the limits of its designated use also involves continuous observance of the operating instructions with particular reference to the maintenance specifications.

Impermissible misuse

Any use or application deviating from the designated use is deemed to be impermissible misuse; examples of such misuse include:

- Transportation of persons and animals
- Use as a climbing aid
- Operation outside the permissible operating parameters
- Use in potentially explosive environments

3.1.4 EC declaration of conformity and declaration of incorporation

This robot system is a partial or incomplete machine as defined by the EC Machinery Directive. The robot system may only be put into operation if the following preconditions are met:

- The robot system is integrated into an overall system.

  Or: The robot system, together with other machines, constitutes an overall system.

  Or: All safety equipment and safeguards required for operation in the overall system as defined by the EC Machinery Directive have been added to the robot system.
- The overall system complies with the EC Machinery Directive. This has been confirmed by means of an assessment of conformity.

Declaration of conformity

The system integrator must issue a declaration of conformity for the overall system in accordance with the Machinery Directive. The declaration of conformity forms the basis for the CE mark for the system. The robot system must be operated in accordance with the applicable national laws, regulations and standards.

The robot controller is CE certified under the EMC Directive and the Low Voltage Directive.

Declaration of incorporation

A declaration of incorporation is provided for the robot system. This declaration of incorporation contains the stipulation that the robot system must not be commissioned until it complies with the provisions of the Machinery Directive.

3.1.5 Description of the robot system

The robot system consists of the following components:

- Robot
- Robot controller
- KCP teach pendant
- Connecting cables
- External axes, e.g. linear unit, two-axis positioner, positioner (optional)
- Top-mounted cabinet (optional)
- Software
- Options, accessories

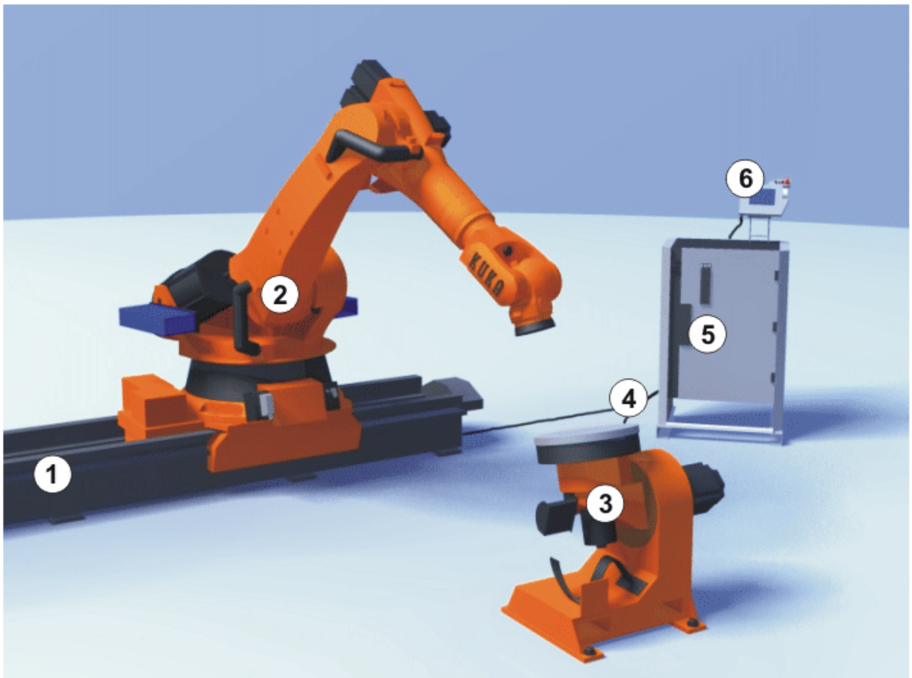


Fig. 3-1: Example of a robot system

1	Linear unit	4	Connecting cables
2	Robot	5	Robot controller
3	Positioner	6	Teach pendant

3.1.6 Terms used

Term	Description
Axis range	Range of each axis, in degrees, within which it may move. The axis range must be defined for each axis that is to be monitored.
Stopping distance	■ Stopping distance = reaction distance + braking distance The stopping distance is part of the danger zone. ■ The braking distance is the distance covered by the robot and any optional external axes after the stop function has been triggered and before the robot comes to a standstill.

Term	Description
Workspace	The robot is allowed to move within its workspace. The workspace is derived from the individual axis ranges.
Operator (User)	The user of the robot system can be the management, employer or delegated person responsible for use of the robot system.
Danger zone	The danger zone consists of the workspace and the stopping distances.
KCP	The KCP (KUKA Control Panel) teach pendant has all the operator control and display functions required for operating and programming the robot system.
Robot system	The robot system consists of the robot controller and robot, together with any options (e.g. KUKA linear unit, two-axis positioner, other positioner, top-mounted cabinet).
Safety zone	The safety zone is situated outside the danger zone.
Stop category 0	The drives are deactivated immediately and the brakes are applied. The robot and any external axes (optional) perform path-oriented braking. **Note:** This stop category is called STOP 0 in this document.
Stop category 1	The robot and any external axes (optional) brake with path-maintaining braking. The drives are deactivated after 1 s and the brakes are applied. **Note:** This stop category is called STOP 1 in this document.
Stop category 2	The drives are not deactivated and the brakes are not applied. The robot and any external axes (optional) are braked with a normal braking ramp. **Note:** This stop category is called STOP 2 in this document.
System integrator (plant integrator)	System integrators are people who safely integrate the robot system into a plant and commission it.
T1	Test mode, Manual Reduced Velocity (<= 250 mm/s)
T2	Test mode, Manual High Velocity (> 250 mm/s permissible)
External axis	Motion axis which is not part of the robot but which is controlled using the robot controller, e.g. KUKA linear unit, two-axis positioner, Posiflex

3.2 Personnel

The following persons or groups of persons are defined for the robot system:

- User
- Personnel

All persons working with the robot system must have read and understood the robot system documentation, including the safety chapter.

User

The user must observe the labor laws and regulations. This includes e.g.:

- The user must comply with his monitoring obligations.
- The user must carry out instruction at defined intervals.

Personnel

Personnel must be instructed, before any work is commenced, in the type of work involved and what exactly it entails as well as any hazards which may exist. Instruction must be repeated after particular incidents or technical modifications.

Personnel includes:

- System integrator
- Operators, subdivided into:
  - Start-up, maintenance and service personnel
  - Operating personnel
  - Cleaning personnel

Installation, exchange, adjustment, operation, maintenance and repair must be performed only as specified in the operating instructions for the relevant component of the robot system and only by personnel specially trained for this purpose.

System integrator

The robot system is safely integrated into a plant by the system integrator.

The system integrator is responsible for the following tasks:

- Installing the robot system
- Connecting the robot system
- Performing risk assessment
- Operating the required safety equipment and safeguards
- Issuing the declaration of conformity
- Attaching the CE mark
- Creating the operating instructions for the overall system

Operator

The operator must meet the following preconditions:

- The operator must be trained for the work to be carried out.
- Work on the robot system must only be carried out by qualified personnel. These are people who, due to their specialist training, knowledge and experience, and their familiarization with the relevant standards, are able to assess the work to be carried out and detect any potential dangers.

Example

The tasks can be distributed as shown in the following table.

Tasks	Operator	Programmer	System integrator
Switch robot controller on/off	x	x	x
Start program	x	x	x
Select program	x	x	x
Select operating mode	x	x	x
Calibration (tool, base)		x	x
Master robot		x	x
Configuration		x	x
Programming		x	x
Start-up			x
Maintenance			x
Repair			x
Shutting down			x
Transportation			x

Work on the electrical and mechanical equipment of the robot system may only be carried out by specially trained personnel.

3.3 Safety equipment and safeguards in the robot system

3.3.1 Overview of the safety equipment and safeguards

The following safety equipment and safeguards are present in the robot system:

- Mode selector switch
  (>>> 3.3.4 "Mode selector switch" page 21)
- Operator safety (= connection for interlocking fixed guards)
  (>>> 3.3.7 "Operator safety" page 23)
- EMERGENCY STOP device
  (>>> 3.3.8 "EMERGENCY STOP device" page 24)
- Enabling device
  (>>> 3.3.9 "Enabling device" page 25)
  (>>> 3.3.10 "Connection for external enabling device" page 25)
- Jog mode in the test modes
  (>>> 3.3.11 "Jog mode" page 26)
- Mechanical limit stops
  (>>> 3.3.12 "Mechanical end stops" page 26)
- Mechanical axis range limitation (optional)
  (>>> 3.3.15 "Mechanical axis range limitation (option)" page 27)
- Axis range monitoring (optional)
  (>>> 3.3.16 "Axis range monitoring (option)" page 27)
- Release device (optional)
  (>>> 3.3.17 "Release device (option)" page 27)
- Labeling of danger areas in the robot system
  (>>> 3.3.20 "Labeling on the robot system" page 29)

The function and triggering of the electronic safety equipment are monitored by the ESC safety logic.

Danger!
In the absence of functional safety equipment and safeguards, the robot system can cause personal injury or material damage. If safety equipment or safeguards are dismantled or deactivated, the robot system may not be operated.

3.3.2 ESC safety logic

The ESC (Electronic Safety Circuit) safety logic is a dual-channel computer-aided safety system. It permanently monitors all connected safety-relevant components. In the event of a fault or interruption in the safety circuit, the power supply to the drives is shut off, thus bringing the robot system to a standstill.

The ESC safety logic triggers different stop reactions, depending on the operating mode of the robot system.

The ESC safety logic monitors the following inputs:

- Operator safety
- Local EMERGENCY STOP (= EMERGENCY STOP button on the KCP)
- External EMERGENCY STOP
- Enabling device
- Drives OFF

- Drives ON
- Operating modes
- Qualifying inputs

3.3.3 Category for safety-related controller components

Safety-related controller components include:

- Local EMERGENCY STOP device (= EMERGENCY STOP button on the KCP)
- External EMERGENCY STOP device
- Enabling device
- External enabling device
- Operator safety
- Qualifying inputs
- Automatic mode
- Test mode

These circuits conform to the requirements of category 3 according to EN 954-1.

3.3.4 Mode selector switch

The robot system can be operated in the following modes:

- Manual Reduced Velocity (T1)
- Manual High Velocity (T2)
- Automatic (AUT)
- Automatic External (AUT EXT)

The operating mode is selected using the mode selector switch on the KCP. The switch is activated by means of a key which can be removed. If the key is removed, the switch is locked and the operating mode can no longer be changed.

If the operating mode is changed during operation, the drives are immediately switched off. The robot and any external axes (optional) are stopped with a STOP 0.

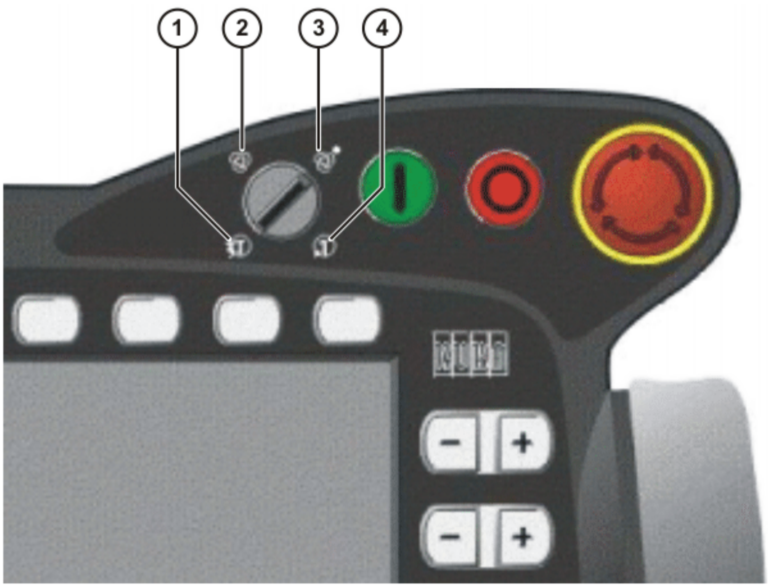


Fig. 3-2: Mode selector switch

1	T2 (Manual High Velocity)
2	AUT (Automatic)
3	AUT EXT (Automatic External)
4	T1 (Manual Reduced Velocity)

Operating mode	Use	Velocities
T1	For test operation, programming and teaching	■ Program verification: Programmed velocity, maximum 250 mm/s ■ Jog mode: Jog velocity, maximum 250 mm/s
T2	For test operation	■ Program verification: Programmed velocity
AUT	For robot systems without higher-level controllers Only possible with a connected safety circuit	■ Program mode: Programmed velocity ■ Jog mode: not possible
AUT EXT	For robot systems with higher-level controllers, e.g. PLC Only possible with a connected safety circuit	■ Program mode: Programmed velocity ■ Jog mode: not possible

3.3.5 Stop reactions

Stop reactions of the robot system are triggered in response to operator actions or as a reaction to monitoring functions and error messages. The following table shows the different stop reactions according to the operating mode that has been set.

STOP 0, STOP 1 and STOP 2 are the stop definitions according to DIN EN 60204-1:2006.

Trigger	T1, T2	AUT, AUT EXT
Safety gate opened	-	STOP 1
EMERGENCY STOP pressed	STOP 0	STOP 1
Enabling withdrawn	STOP 0	-
Start key released	STOP 2	-
“Drives OFF” key pressed	STOP 0	
STOP key pressed	STOP 2	
Operating mode changed	STOP 0	
Encoder error (DSE-RDC connection broken)	STOP 0	

Trigger	T1, T2	AUT, AUT EXT
Motion enable canceled	STOP 2	
Robot controller switched off Power failure	STOP 0	

3.3.6 Workspace, safety zone and danger zone

Workspaces are to be restricted to the necessary minimum size. A workspace must be safeguarded using appropriate safeguards.

The safeguards (e.g. safety gate) must be situated inside the safety zone. If a safeguard is triggered, robot and external axes (optional) are braked and come to a stop within the danger zone.

The danger zone consists of the workspace and the stopping distances of the robot and external axes (optional). It must be safeguarded by means of physical safeguards to prevent danger to persons or the risk of material damage.

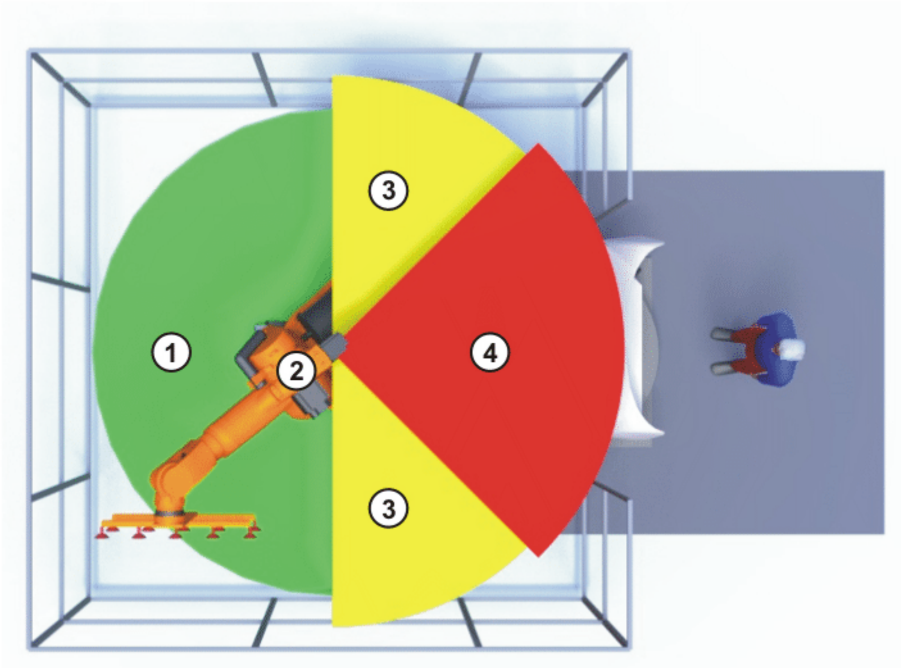


Fig. 3-3: Example of axis range A1

1	Workspace	3	Stopping distance
2	Robot	4	Safety zone

3.3.7 Operator safety

The operator safety input is used for interlocking fixed guards. Safety equipment, such as safety gates, can be connected to the dual-channel input. If nothing is connected to this input, operation in Automatic mode is not possible. Operator safety is not active in the test modes T1 (Manual Reduced Velocity) and T2 (Manual High Velocity).

In the event of a loss of signal during Automatic operation (e.g. safety gate is opened), the drives are deactivated after 1 s and the robot and any external axes (optional) are stopped with a STOP 1. When the signal is applied again at the input (e.g. safety gate closed), Automatic operation can be resumed once the corresponding message has been acknowledged.

The operator safety must be designed in such a way that it is only possible to acknowledge the message from outside.

Operator safety can be connected via the peripheral interface on the robot controller.

3.3.8 EMERGENCY STOP device

The EMERGENCY STOP device for the robot system is the EMERGENCY STOP button on the KCP. The button must be pressed in the event of a hazardous situation or emergency.

Reactions of the robot system if the EMERGENCY STOP button is pressed:

- Manual Reduced Velocity (T1) and Manual High Velocity (T2) modes:

  The drives are switched off immediately. The robot and any external axes (optional) are stopped with a STOP 0.
- Automatic modes (AUT and AUT EXT):

  The drives are switched off after 1 second. The robot and any external axes (optional) are stopped with a STOP 1.

Before operation can be resumed, the EMERGENCY STOP button must be turned to release it and the stop message must be acknowledged.

Fig. 3-4: EMERGENCY STOP button on the KCP

1 EMERGENCY STOP button

Warning!
Tools and other equipment connected to the robot must be integrated into the EMERGENCY STOP circuit on the system side if they could constitute a potential hazard.
Failure to observe this precaution may result in death, severe physical injuries or considerable damage to property.

3.3.9 Enabling device

The enabling devices of the robot system are the enabling switches on the KCP.

There are 3 enabling switches installed on the KCP. The enabling switches have 3 positions:

- Not pressed
- Center position
- Panic position

In the test modes T1 (Manual Reduced Velocity) and T2 (Manual High Velocity), the robot can only be moved if one of the enabling switches is held in the central position. If the enabling switch is released or pressed fully down (panic position), the drives are deactivated immediately and the robot stops with a STOP 0.

Warning!
The enabling switches must not be held down by adhesive tape or other means or manipulated in any other way.
Death, serious physical injuries or major damage to property may result.

Fig. 3-5: Enabling switches on the KCP

1 - 3 Enabling switches

3.3.10 Connection for external enabling device

External enabling devices are required if there is more than one person in the danger zone of the robot system. They can be connected via the peripheral interface on the robot controller.

External enabling devices are not included in the scope of supply of the robot system.

3.3.11 Jog mode

In the operating modes T1 (Manual Reduced Velocity) and T2 (Manual High Velocity), the robot can only execute programs in jog mode. This means that it is necessary to hold down an enabling switch and the Start key in order to execute a program.

If the enabling switch is released or pressed fully down (panic position), the drives are deactivated immediately and the robot and any external axes (optional) stop with a STOP 0.

Releasing only the Start key causes the robot system to be stopped with a STOP 2.

3.3.12 Mechanical end stops

The axis ranges of main axes A 1 to A 3 and wrist axis A 5 of the robot are limited by means of mechanical limit stops with a buffer.

Additional mechanical limit stops can be installed on the external axes.

Danger!
If the robot or an external axis hits an obstruction or a buffer on the mechanical end stop or axis range limitation, this can result in material damage to the robot system. KUKA Roboter GmbH must be consulted before the robot system is put back into operation (>>> 9 "KUKA Service" page 69). The affected buffer must immediately be replaced with a new one. If a robot (or external axis) collides with a buffer at more than 250 mm/s, the robot (or external axis) must be exchanged or recommissioning must be carried out by the KUKA Roboter GmbH.

3.3.13 Software limit switches

The axis ranges of all robot and positioner axes are limited by means of adjustable software limit switches. These software limit switches only serve as machine protection and must be adjusted in such a way that the robot/positioner cannot hit the mechanical limit stops.

The software limit switches are set during commissioning of a robot system.

Further information is contained in the operating and programming instructions.

3.3.14 Overview of operating modes and safety measures

The following table indicates the operating modes in which the safety measures are active.

Safety measures	T1	T2	AUT	AUT EXT
Operator safety	-	-	active	active
EMERGENCY STOP device	active (STOP 0)	active (STOP 0)	active (STOP 1)	active (STOP 1)
Enabling device	active	active	-	-
Reduced velocity during program verification	active	-	-	-

Safety measures	T1	T2	AUT	AUT EXT
Jog mode	active	active	-	-
Software limit switches	active	active	active	active

3.3.15 Mechanical axis range limitation (option)

Most robots can be fitted with mechanical axis range limitation in axes A 1 to A 3. The adjustable axis range limitation systems restrict the working range to the required minimum. This increases personal safety and protection of the system.

In the case of robots that are not designed to be fitted with mechanical axis range limitation, the workspace must be laid out in such a way that there is no danger to persons or material property, even in the absence of mechanical axis range limitation.

If this is not possible, the workspace must be limited by means of photoelectric barriers, photoelectric curtains or obstacles on the system side. There must be no shearing or crushing hazards at the loading and transfer areas.

This option can be retrofitted.

3.3.16 Axis range monitoring (option)

Most robots can be fitted with dual-channel axis range monitoring systems in main axes A 1 to A 3. The positioner axes may be fitted with additional axis range monitoring systems. The safety zone for an axis can be adjusted and monitored using an axis range monitoring system. This increases personal safety and protection of the system.

This option can be retrofitted.

3.3.17 Release device (option)

Description

The release device can be used to move the robot mechanically after an accident or malfunction. The release device can be used for the main axis drive motors and, depending on the robot variant, also for the wrist axis drive motors. It is only for use in exceptional circumstances and emergencies (e.g. for freeing people). After use of the release device, the affected motors must be exchanged.

Warning!
The motors reach temperatures during operation which can cause burns to the skin. Contact should be avoided if at all possible. If necessary, appropriate protective equipment must be used.

Procedure

1. Switch off the robot controller and secure it (e.g. with a padlock) to prevent unauthorized persons from switching it on again.
2. Remove the protective cap from the motor.
3. Push the release device onto the corresponding motor and move the axis in the desired direction.

   The directions are indicated with arrows on the motors. It is necessary to overcome the resistance of the mechanical motor brake and any other loads acting on the axis.

Warning!
Moving an axis with the release device can damage the motor brake. This can result in personal injury and material damage. After using the release device, the affected motor must be exchanged.

Further information is contained in the robot operating instructions.

3.3.18 KCP coupler (optional)

The KCP coupler allows the KCP to be connected and disconnected with the robot controller running.

Warning!
The operator must ensure that decoupled KCPs are immediately removed from the system and stored out of sight and reach of the robot system personnel. This serves to prevent operational and non-operational EMERGENCY STOP facilities from becoming interchanged.
Failure to observe this precaution may result in death, severe physical injuries or considerable damage to property.

Further information is contained in the robot controller operating instructions.

3.3.19 External safety equipment and safeguards

E-STOP

There must be EMERGENCY STOP devices on every operator panel and anywhere else it may be necessary to trigger an EMERGENCY STOP. The system integrator is responsible for ensuring this. External EMERGENCY STOP devices are connected to the peripheral interface of the robot.

Safeguards

The access of persons to the danger zone of the robot must be prevented by means of safeguards.

Physical safeguards must meet the following requirements:

- They meet the requirements of EN 953.
- They prevent access of persons to the danger zone and cannot be easily circumvented.
- They are sufficiently fastened and can withstand all forces that are likely to occur in the course of operation, whether from inside or outside the enclosure.
- They do not, themselves, represent a hazard or potential hazard.
- The prescribed minimum clearance from the danger zone is maintained.

Safety gates (maintenance gates) must meet the following requirements:

- They are reduced to an absolute minimum.
- The interlocks (e.g. safety gate switches) are linked to the operator safety input of the robot controller via safety gate switching devices or safety PLC.
- Switching devices, switches and the type of switching conform to category 3 according to EN 954-1.
- Depending on the risk situation: the safety gate is additionally safeguarded by means of a locking mechanism that only allows the gate to be opened if the robot is safely at a standstill.

- The button for acknowledging the safety gate is located outside the space limited by the safeguards.

Further information is contained in the corresponding standards and regulations. These also include DIN EN 953.

Other safety equipment

Other safety equipment must be integrated into the system in accordance with the corresponding standards and regulations.

3.3.20 Labeling on the robot system

All plates, labels, symbols and marks constitute safety-relevant parts of the robot system. They must not be modified or removed.

Labeling on the robot system consists of:

- Rating plates
- Warning labels
- Safety symbols
- Designation labels
- Cable markings
- Identification plates

Further information can be found in the operating instructions of the robot, linear unit, positioner and robot controller.

3.4 Safety measures

3.4.1 General safety measures

The robot system may only be used in perfect technical condition in accordance with its designated use and only by safety-conscious persons. Operator errors can result in personal injury and damage to property.

It is important to be prepared for possible movements of the robot system even after the robot controller has been switched off and locked. Incorrect installation (e.g. overload) or mechanical defects (e.g. brake defect) can cause the robot or external axes to sag. If work is to be carried out on a switched-off robot system, the robot and external axes must first be moved into a position in which they are unable to move on their own, whether the payload is mounted or not. If this is not possible, the robot and external axes must be secured by appropriate means.

Danger!
In the absence of functional safety equipment and safeguards, the robot system can cause personal injury or material damage. If safety equipment or safeguards are dismantled or deactivated, the robot system may not be operated.

Warning!
The motors reach temperatures during operation which can cause burns to the skin. Contact should be avoided if at all possible. If necessary, appropriate protective equipment must be used.

KCP

If more than one KCP is used in the overall system, it must be ensured that each KCP is unambiguously assigned to the corresponding robot system. There must be no possibility of mixing them up in an emergency situation.

Warning!
The operator must ensure that decoupled KCPs are immediately removed from the system and stored out of sight and reach of the robot system personnel. This serves to prevent operational and non-operational EMERGENCY STOP facilities from becoming interchanged.
Failure to observe this precaution may result in death, severe physical injuries or considerable damage to property.

External keyboard, external mouse

An external keyboard and/or external mouse may only be used if the following conditions are met:

- Start-up or maintenance work is being carried out.
- The drives are switched off.
- There are no persons in the danger zone.

The KCP must not be used as long as an external keyboard and/or external mouse are connected.

The external keyboard and/or external mouse must be removed as soon as the start-up or maintenance work is completed.

Faults

The following tasks must be carried out in the case of faults to the robot system:

- Switch off the robot controller and secure it (e.g. with a padlock) to prevent unauthorized persons from switching it on again.
- Indicate the fault by means of a label with a corresponding warning (tagout).
- Keep a record of the faults.
- Eliminate the fault and carry out a function test.

Modifications

After modifications to the robot system, checks must be carried out to ensure the required safety level. The valid national or regional work safety regulations must be observed for this check. The correct functioning of all safety circuits must also be tested.

3.4.2 Transportation

Robot

The prescribed transport position of the robot must be observed. Transportation must be carried out in accordance with the robot operating instructions.

Robot controller

The robot controller must be transported and installed in an upright position. Avoid vibrations and impacts during transportation in order to prevent damage to the robot controller.

Transportation must be carried out in accordance with the operating instructions for the robot controller.

External axis (optional)

The prescribed transport position of the external axis (e.g. KUKA linear unit, two-axis positioner, etc.) must be observed. Transportation must be carried out in accordance with the operating instructions for the external axis.

3.4.3 Start-up and recommissioning

Before starting up systems and devices for the first time, a check must be carried out to ensure that the systems and devices are complete and operational, that they can be operated safely and that any damage is detected.

The valid national or regional work safety regulations must be observed for this check. The correct functioning of all safety circuits must also be tested.

The passwords for logging onto the KUKA System Software as “Expert” and “Administrator” must be changed before start-up and must only be communicated to authorized personnel.

Danger!
The robot controller is preconfigured for the specific robot system. If cables are interchanged, the robot and the external axes (optional) may receive incorrect data and can thus cause personal injury or material damage. If a system consists of more than one robot, always connect the connecting cables to the robots and their corresponding robot controllers.

Caution!
If the internal cabinet temperature of the robot controller differs greatly from the ambient temperature, condensation can form, which may cause damage to the electrical components. Do not put the robot controller into operation until the internal temperature of the cabinet has adjusted to the ambient temperature.

Function test

The following tests must be carried out before start-up and recommissioning:

General test

It must be ensured that:

- The robot system is correctly installed and fastened in accordance with the specifications in the documentation.
- There are no foreign bodies or loose parts on the robot system.
- All required safety equipment is correctly installed and operational.
- The power supply ratings of the robot system correspond to the local supply voltage and mains type.
- The ground conductor and the equipotential bonding cable are sufficiently rated and correctly connected.
- The connecting cables are correctly connected and the connectors are locked.

Test of safety-oriented circuits

A function test must be carried out for the following safety-oriented circuits to ensure that they are functioning correctly:

- Local EMERGENCY STOP device (= EMERGENCY STOP button on the KCP)
- External EMERGENCY STOP device (input and output)
- Enabling device (in the test modes)
- Operator safety (in the automatic modes)
- Qualifying inputs (if connected)

Test of reduced velocity control

This test is to be carried out as follows:

1. Program a straight path with the maximum possible velocity.
2. Calculate the length of the path.
3. Execute the path in T1 mode with the override set to 100% and time the motion with a stopwatch.
4. Calculate the velocity from the length of the path and the time measured for execution of the motion.

   The velocity thus calculated must not exceed 250 mm/s.

Machine data

It must be ensured that the rating plate on the robot controller has the same machine data as those entered in the declaration of incorporation. The machine data on the rating plate of the robot and the external axes (optional) must be entered during start-up.

Warning!
The robot must not be moved if incorrect machine data are loaded. Death, severe physical injuries or considerable damage to property may otherwise result. The correct machine data must be loaded.

3.4.4 Virus protection and network security

The user of the robot system is responsible for ensuring that the software is always safeguarded with the latest virus protection. If the robot controller is integrated into a network that is connected to the company network or to the Internet, it is advisable to protect this robot network against external risks by means of a firewall.

For optimal use of our products, we recommend that our customers carry out a regular virus scan. Information about security updates can be found at www.kuka.com.

3.4.5 Manual mode

Manual mode is the mode for setup work. Setup work is all the tasks that have to be carried out on the robot system to enable automatic operation. Setup work includes:

- Jog mode
- Teaching
- Programming
- Program verification

The following must be taken into consideration in manual mode:

- If the drives are not required, they must be switched off to prevent the robot or the external axes (optional) from being moved unintentionally.

  New or modified programs must always be tested first in Manual Reduced Velocity mode (T1).
- The robot, tooling or external axes (optional) must never touch or project beyond the safety fence.
- Components, tooling and other objects must not become jammed due to the motion of the robot system, nor must they lead to short-circuits or be liable to fall off.
- All setup work must be carried out, where possible, from outside the safeguarded area.

If the setup work has to be carried out inside the safeguarded area, the following must be taken into consideration:

In **Manual Reduced Velocity mode (T1)**:

- If it can be avoided, there must be no other persons inside the safeguarded area.

  If it is necessary for there to be several persons inside the safeguarded area, the following must be observed:
  - Each person must have an enabling device.
  - All persons must have an unimpeded view of the robot system.
  - Eye-contact between all persons must be possible at all times.
- The operator must be so positioned that he can see into the danger area and get out of harm's way.

In **Manual High Velocity mode (T2)**:

- This mode may only be used if the application requires a test at a velocity higher than Manual Reduced Velocity.
- Teaching and programming are not permissible in this operating mode.
- Before commencing the test, the operator must ensure that the enabling devices are operational.
- The operator must be positioned outside the danger zone.
- There must be no other persons inside the safeguarded area. it is the responsibility of the operator to ensure this.

3.4.6 Simulation

Simulation programs do not correspond exactly to reality. Robot programs created in simulation programs must be tested in the system in Manual Reduced Velocity mode (SSTEP T1). t may be necessary to modify the program.

3.4.7 Automatic mode

Automatic mode is only permissible in compliance with the following safety measures:

- All safety equipment and safeguards are present and operational.
- There are no persons in the production line.
- The defined working procedures are adhered to.

If the robot or an external axis (optional) comes to a standstill for no apparent reason, the danger zone must not be entered until an EMERGENCY STOP has been triggered.

3.4.8 Maintenance and repair

After maintenance and repair work, checks must be carried out to ensure the required safety level. The valid national or regional work safety regulations must be observed for this check. The correct functioning of all safety circuits must also be tested.

The purpose of maintenance and repair work is to ensure that the system is kept operational or, in the event of a fault, to return the system to an operational state. Repair work includes troubleshooting in addition to the actual repair itself.

The following safety measures must be carried out when working on the robot system:

- Carry out work outside the danger zone. If work inside the danger zone is necessary, the user must define additional safety measures to ensure the safe protection of personnel.
- Switch off the robot controller and secure it (e.g. with a padlock) to prevent unauthorized persons from switching it on again. If it is necessary to carry out work with the robot controller switched on, the user must define additional safety measures to ensure the safe protection of personnel.
- If it is necessary to carry out work with the robot controller switched on, this may only be done in operating mode T1.
- Label the system with a sign indicating that work is in progress. This sign must remain in place, even during temporary interruptions to the work.

- The EMERGENCY STOP systems must remain active. If safety equipment or safeguards are deactivated during maintenance or repair work, they must be reactivated immediately after the work is completed.

Faulty components must be replaced using new components with the same article numbers or equivalent components approved by KUKA Roboter GmbH for this purpose.

Cleaning and preventive maintenance work is to be carried out in accordance with the operating instructions.

Robot controller

Even when the robot controller is switched off, parts connected to peripheral devices may still carry voltage. The external power sources must therefore be switched off if work is to be carried out on the robot controller.

The ESD regulations must be adhered to when working on components in the robot controller.

Voltages in excess of 50 V (up to 600 V) can be present in the KPS 600 (KUKA Power Supply), the KSDs (KUKA Servo Drives) and the intermediate-circuit connecting cables several minutes after the robot controller has been switched off. To prevent life-threatening injuries, no work may be carried out on the robot system in this time.

Water and dust must be prevented from entering the robot controller.

Counterbalancing system

Some robot variants are equipped with a hydropneumatic, spring or gas cylinder counterbalancing system.

The hydropneumatic and gas cylinder counterbalancing systems are pressure equipment and, as such, are subject to obligatory equipment monitoring. Depending on the robot variant, the counterbalancing systems correspond to category II or III, fluid group 2, of the Pressure Equipment Directive.

The user must comply with the applicable national laws, regulations and standards pertaining to pressure equipment.

Inspection intervals in Germany in accordance with Industrial Safety Order, Sections 14 and 15. Inspection by the user before commissioning at the installation site.

The following safety measures must be carried out when working on the counterbalancing system:

- The robot assemblies supported by the counterbalancing systems must be secured.
- Work on the counterbalancing systems must only be carried out by qualified personnel.

Hazardous substances

The following safety measures must be carried out when handling hazardous substances:

- Avoid prolonged and repeated intensive contact with the skin.
- Avoid breathing in oil spray or vapors.
- Clean skin and apply skin cream.

To ensure safe use of our products, we recommend that our customers regularly request up-to-date safety data sheets from the manufacturers of hazardous substances.

3.4.9 Decommissioning, storage and disposal

The robot system must be decommissioned, stored and disposed of in accordance with the applicable national laws, regulations and standards.

3.4.10 Safety measures for “single point of control”

Overview

If certain components in the robot system are operated, safety measures must be taken to ensure complete implementation of the principle of “single point of control”.

Components:

- Submit interpreter
- PLC
- OPC server
- Remote control tools
- External keyboard/mouse

> The implementation of additional safety measures may be required. This must be clarified for each specific application; this is the responsibility of the system integrator, programmer or user of the system.

Since only the system integrator knows the safe states of actuators in the periphery of the robot controller, it is his task to set these actuators to a safe state, e.g. in the event of an EMERGENCY STOP.

Submit interpreter, PLC

If motions, (e.g. drives or grippers) are controlled with the Submit interpreter or the PLC via the I/O system, and if they are not safeguarded by other means, then this control will take effect even in T1 and T2 modes or while an EMERGENCY STOP is active.

If variables that affect the robot motion (e.g. override) are modified with the Submit interpreter or the PLC, this takes effect even in T1 and T2 modes or while an EMERGENCY STOP is active.

Safety measures:

- Do not modify safety-relevant signals and variables (e.g. operating mode, EMERGENCY STOP, safety gate contact) via the Submit interpreter or PLC.
- If modifications are nonetheless required, all safety-relevant signals and variables must be linked in such a way that they cannot be set to a dangerous state by the Submit interpreter or PLC.

OPC server, remote control tools

These components can be used with write access to modify programs, outputs or other parameters of the robot controller, without this being noticed by any persons located inside the system.

Safety measures:

- KUKA stipulates that these components are to be used exclusively for diagnosis and visualization.

  Programs, outputs or other parameters of the robot controller must not be modified using these components.

External keyboard/mouse

These components can be used to modify programs, outputs or other parameters of the robot controller, without this being noticed by any persons located inside the system.

Safety measures:

- Only use one operator console at each robot controller.
- If the KCP is being used for work inside the system, remove any keyboard and mouse from the robot controller beforehand.

3.5 Applied norms and regulations

Name	Definition	Edition
2006/95/EC	Low Voltage Directive: Directive of the European Parliament and the Council of 12 December 2006 on the harmonization of the laws of Member States relating to electrical equipment designed for use within certain voltage limits	2006
89/336/EEC	EMC Directive: Council Directive of 3 May 1989 on the approximation of the laws of the Member States relating to electromagnetic compatibility	1993
97/23/EC	Pressure Equipment Directive: Directive of the European Parliament and of the Council of 29 May 1997 on the approximation of the laws of the Member States concerning pressure equipment	1997
98/37/EC	Machinery Directive: Directive of the European Parliament and of the Council of 22 June 1998 on the approximation of the laws of the Member States relating to machinery	1998
EN ISO 13850	Safety of machinery: E-STOP - Principles for design	2007
EN ISO 13732-1	Ergonomics of thermal environment: Methods for the assessment of human responses to contact with surfaces - Part 1: Hot surfaces	2006
EN 614-1	Safety of machinery: Ergonomic design principles – Part 1: Terms and general principles	2006
EN 954-1	Safety of machinery: Safety-related parts of control systems - Part 1: General principles for design	1997
EN 60204-1	Safety of machinery: Electrical equipment of machines - Part 1: General requirements	2007
EN 61000-6-2	Electromagnetic compatibility (EMC): Part 6-2: Generic standards; Immunity for industrial environments	2002
EN 61000-6-4	Electromagnetic compatibility (EMC): Part 6-4: Generic standards; Emission standard for industrial environments	2002
EN ISO 10218-1	Industrial robots: Safety	2006

Name	Definition	Edition
EN ISO 12100-1	Safety of machinery: Basic concepts, general principles for design - Part 1: Basic terminology, methodology	2004
EN ISO 12100-2	Safety of machinery: Basic concepts, general principles for design - Part 2: Technical principles	2004

4 Installation

4.1 System requirements

Hardware

- Robot controller:
  - KR C2
  - KR C2 ed05
  - KR C2 sr
  - KR C3
- External system: processor-supported system with network card
- Network cable for switch, hub or crossed network cable for direct connection

Software

- KUKA System Software 5.2, 5.3, 5.4, 5.5, 5.6 or 7.0

External system

- Communication via TCP/IP protocol
- XML parser for generating XML strings with the data for the robot controller
- Sending and receiving of XML structures via a port

The robot controller does not always send XML structures to the external system in a packet.

This is for the following reasons:

- The KRL can send character strings as a single item only if they are shorter than 80 characters.
- The Nagel algorithm incorporated in the TCP/IP protocol optimizes the packet size for data throughput on the bus. It is thus not possible to predict the size of the packets sent on the bus.

The application on the external system must itself determine whether an XML structure is complete, based on the well-formedness. This can be achieved, for instance, by waiting for the end tag of the root element.

If the external system is only intended to transmit data to the robot controller, the implementation requires very little effort. It is merely necessary to send well-formed XML strings to the robot controller.

If the external system is also to receive data from the robot controller, it is necessary, in most cases, to integrate an XML parser into the server application. A parser simplifies the reading of data and avoids complex string operations.

Recommendation

XML parser:

- Microsoft .Net XML parser
- Gnome parser, SuSE LINUX

4.2 PCI slot assignment

Overview

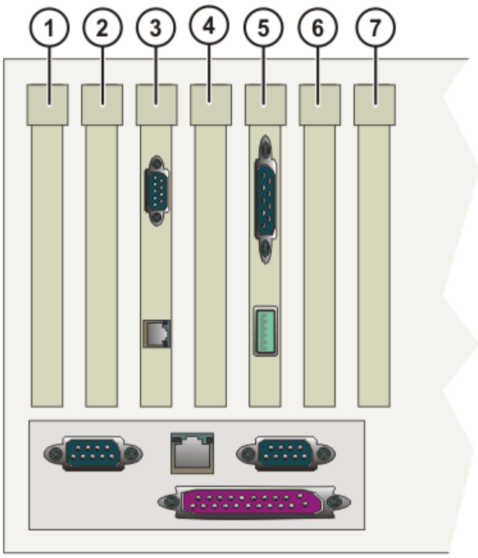


Fig. 4-1: PCI slots

The PC slots can be fitted with the following plug-in cards:

Slot	Plug-in card
1	■ Interbus card (FOC) (optional) ■ Interbus card (copper) (optional) ■ LPDN scanner card (optional) ■ Profibus master/slave card (optional) ■ CN_EthernetIP card (optional)
2	■ LPDN scanner card (optional)
3	KVGA card
4	DSE-IBS-C33 AUX card (optional)
5	MFC3 card
6	■ Network card (optional) ■ LPDN scanner card (optional) ■ Profibus master/slave card (optional) ■ LIBO-2PCI card (optional) ■ KUKA modem card (optional)
7	Not assigned

4.3 Installing KUKA.Ethernet KRL XML

Precondition

- User group “Expert”
- Windows interface (CTRL+ESC)
- Installed KUKA network card

During installation, the network card is automatically assigned to the Vx-Works kernel. The Windows driver is deleted.
This function is not supported by the following configurations:

- KR C2 ed05 with KSS 5.2 and KUKA network card
- KR C2 ed05 without optional KUKA network card
- KR C2 sr without optional KUKA network card

For these configurations, the network connection under VxWorks must be established using KUKA.Router. (>>> 5.1.2 "Network connection under Vx-Works with KUKA.Router" page 44)

Procedure

1. Start the program **Setup.exe** from the CD-ROM. A window is opened, requesting the network address.

If no window requesting a network address is opened, a network interface has already been installed (>>> 8.1 "Diagnosis with Telnet" page 67).

2. Enter the network address of the robot controller. The files are copied onto the hard drive.

The IP address range 192.0.1.x is reserved and is disabled for applications. Configuring the VxWorks network connection with this address range results in a system error in the KUKA system software. It is no longer possible to boot the robot controller.

3. Confirm the reboot prompt with **OK**.
4. Reboot the robot controller.

LOG file

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

4.4 Uninstalling KUKA.Ethernet KRL XML

Precondition

- KUKA.Ethernet KRL XML is installed.
- User group “Expert”
- Windows interface (CTRL+ESC)

Procedure

1. Start the **UnInstall.exe** program in the directory C:\KRC_OPTION\ETH-ERNETKRLXML\UNINST. Uninstallation is prepared.
2. Confirm the reboot prompt with **OK**.
3. Reboot the robot controller.

LOG file

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

4.5 Reinstalling KUKA.Ethernet KRL XML

Precondition

- KUKA.Ethernet KRL XML has been uninstalled.
- User group “Expert”
- Windows interface (CTRL+ESC)

Procedure

1. Start the **ReInstall.exe** program in the directory C:\KRC_OPTION\ETH-ERNETKRLXML\REINST. Setup is prepared.
2. Confirm the reboot prompt with **OK**.
3. Reboot the robot controller.

LOG file

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

4.6 Installing KUKA.Router

Precondition

- Network connection in Windows between robot controller and external PC

Further information about KUKA.Router can be found in the KUKA_Router directory on the CD-ROM.

Procedure

1. Start the program **Setup.exe** in the KUKA_Router directory on the CD-ROM.
2. Accept the license agreement and press **Next**.

   The Router will then be installed.
3. Reboot the robot controller.

5 Configuration

Overview

Step	Description
1	Establish the physical network connections (>>> 5.1 "Network connections" page 43)
2	Installation; assign or modify the IP address as required (>>> 5.2 "Modifying the IP address for KSS 5.x" page 45) (>>> 5.3 "Modifying the IP address for KSS 7.0" page 46)
3	Optional: Configure KUKA.Router (>>> 5.4 "Configuring KUKA.Router" page 46)
4	Configure the communications parameters (>>> 5.5 "Configuring communication parameters" page 46)
5	Define the XML structure (>>> 5.6 "Defining an XML structure for receiving data" page 47) (>>> 5.7 "Defining an XML structure for sending data" page 48)
6	Reconfigure the I/O driver (>>> 5.8 "Reconfiguring the I/O driver" page 49)
7	Create the KRL program (>>> 6 "Programming" page 51)

5.1 Network connections

Overview

Various network connections are possible on the robot controller:

- Network connection under Windows
- Network connection under VxWorks with KUKA.Router
- Network connection under VxWorks

5.1.1 Network connection under Windows

Description

2 operating systems run simultaneously on the robot controller:

- MS Windows is responsible for the visualization.
- VxWorks is responsible for the real-time control of the robot system.

The data traffic between the two operating systems takes place via the shared memory driver (TCP/IP).

Addresses of the shared memory driver:

- 192.0.1.1 for VxWorks
- 192.0.1.2 for Windows

These two addresses are preset on every robot controller.

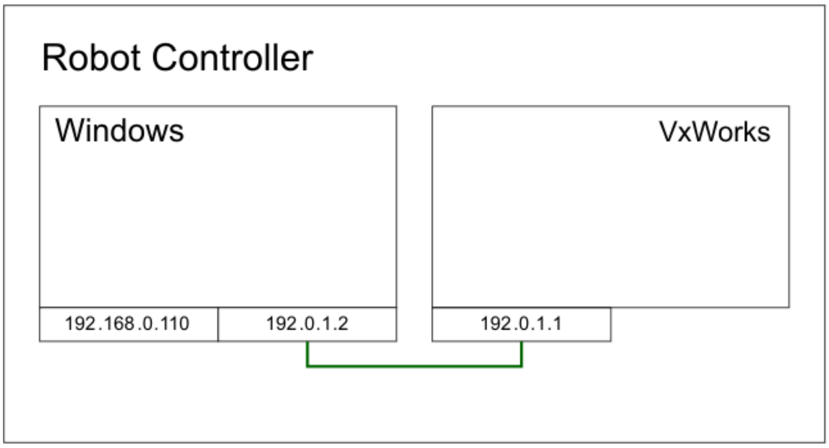


Fig. 5-1: Shared memory network

Robot controllers are supplied as standard with a network card:

- KR C2: Network card integrated into the MFC2
- KR C2 ed05: Network card integrated into the motherboard
- KR C2 sr: Network card integrated into the motherboard

Optionally, a further KUKA network card can also be used.

In the figure, the network card is assigned to Windows and has the address "192.168.0.110" (example only). This address can be used to connect the controller to a network.

5.1.2 Network connection under VxWorks with KUKA.Router

Description

In this example, the robot controller is connected to a network via a hub or switch. The sensor system and the external computer represent devices with which the robot controller is in communication. The network card is operated under Windows. Windows cannot perform routing, i.e. it cannot forward data packets between the addresses "192.168.0.110" and "192.0.1.2". KUKA.Router must be used to perform forwarding.

If KR C2 ed05 is used with KSS 5.2, or KR C2 ed05 and KR C2 sr without an optional KUKA network card, the data packets must always be forwarded using KUKA.Router.

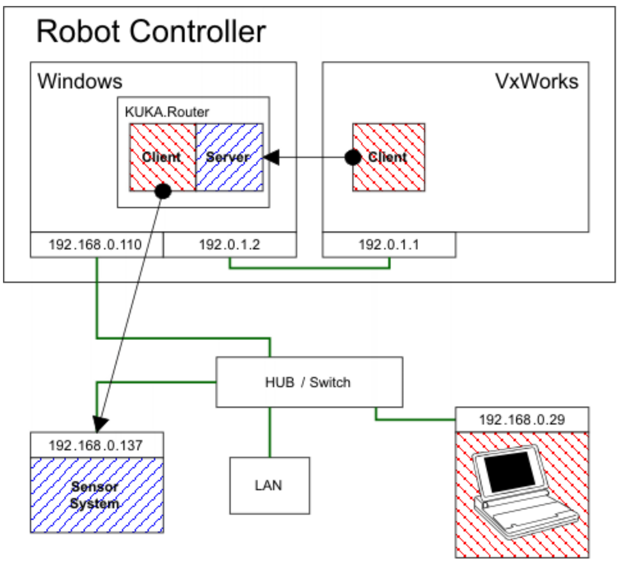


Fig. 5-2: Network connection to VxWorks with KUKA.Router

The figure shows schematically how the KUKA.Router program works. The target server here is any sensor system. The router appears to VxWorks as a server. For the external system, the router represents the client. The Ethernet KRL XML software running on VxWorks is thus connected as a client to the server in the router and the client in the router is connected to the actual destination server (here a sensor system).

Disadvantages

- Data exchange between KUKA.HMI and VxWorks runs in parallel to the data exchange between the external system and VxWorks using the shared memory network. Where large volumes of data are being handled, this can lead to poor performance.
- Additional configuration work, due to the configuration of KUKA.Router. (>>> 5.4 "Configuring KUKA.Router" page 46)

Advantages

- Windows and VxWorks can be connected to a network at the same time, using only a single network card. It is not necessary to install an additional network or driver.

5.1.3 Network connection under VxWorks

Description

Direct data exchange with VxWorks allows greater performance during data transfer.

- KR C2 ed05: requires a KUKA network card (option).
- KR C2 sr: requires a KUKA network card (option).
- KR C2: the network connection can be made via the MFC2.

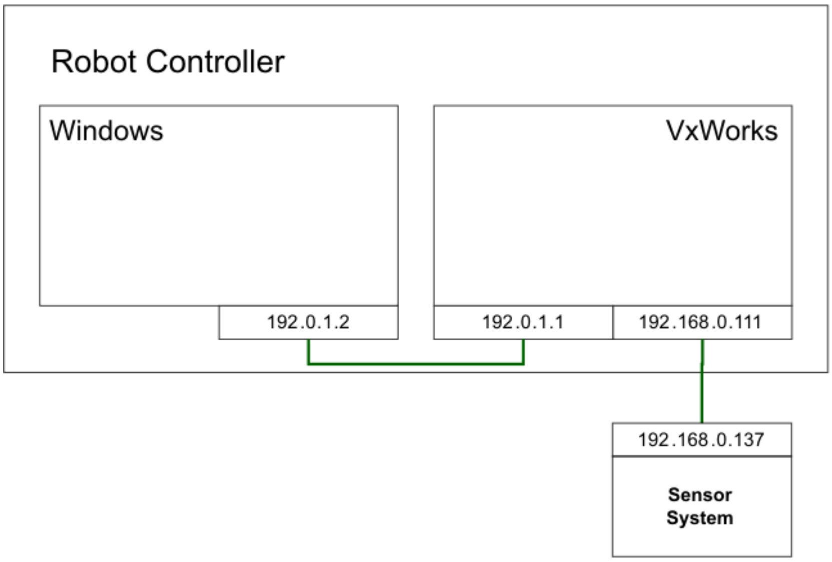


Fig. 5-3: Network connection under VxWorks

Advantages

- The direct connection to VxWorks reduces the load on the robot controller, since data exchange no longer uses the shared memory network.
- Bypassing the shared memory network shortens the transmission distance, thus enabling faster data transmission.
- With a point-to-point connection there is no risk of data collisions on the bus, thus enabling faster data transmission.

5.2 Modifying the IP address for KSS 5.x

Precondition

- User group “Expert”
- Windows interface (CTRL+ESC)

Procedure

1. Open the file C:\Windows\vxWin.ini.
2. Modify the IP address under e={......}.
3. Save and close.
4. Reboot the robot controller.

5.3 Modifying the IP address for KSS 7.0

Precondition

- User group “Expert”
- Windows interface (CTRL+ESC)

Procedure

1. Open the file C:\KRC\ROBOTER\INIT\progress.ini.
2. Modify the IP address under IPADDR_ELPCI.
3. Save and close.
4. Reboot the robot controller.

5.4 Configuring KUKA.Router

Precondition

- User group “Expert”
- Windows interface (CTRL+ESC)

Procedure

1. On the Windows interface, open the program KUKA.Router by selecting the menu sequence **Start** > **Programs** > **Startup** > **Router**.
2. Add a new route using the menu sequence **File** > **New**.

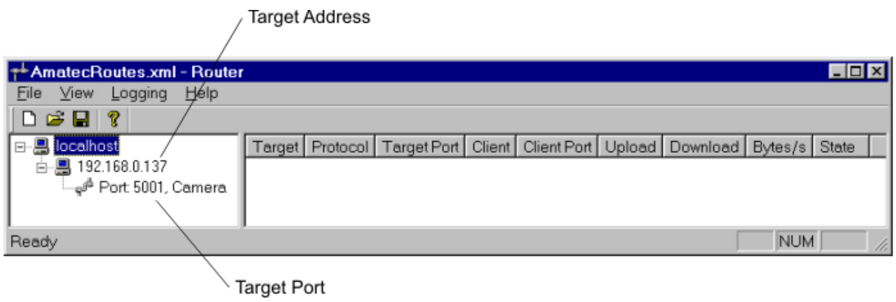


Fig. 5-4: KUKA.Router program

3. Using the arrow keys, select the IP address and press the Enter key.
   The **Configure target** window is opened.
4. In the **Target host** box, enter the target address and press **OK**.
5. Using the arrow keys, select the port and press the Enter key.
   The **Configure route** window is opened.
6. Enter the source port in the **Source Port** box.
7. Enter the target port in the **Target Port** box, and press **OK**.
8. Minimize the **KUKA.Router** window.

Do not close the program, as otherwise no data will be sent.

5.5 Configuring communication parameters

Description

The communication parameters are defined in the file C:\KRC\ROBOTER\INIT\XmlApiConfig.XML.

```
<?xml version="1.0"?>
<!--  NAME                                      -->
<!--    C:\Krc\Roboter\Init\XmlApiConfig.XML    -->
<!--                                            -->
<XmlApiConfig>
  <!--  -->
  <!--  -->
  <XmlApiParam InitOnce="false"/>
  <!--  -->
  <!--  -->
  <Channel SensorName="StackCam" SensorType="dvt600">
     <TCP_IP IP="192.168.0.29" Port="5001" Route="true" Map-
Port="5001"/>
  </Channel>
</XmlApiConfig>
```

No spaces or tabstops may be inserted after the </Channel> tag.

Parameter	Attribute	Description
XmlApi-Param	InitOnce	Defines when the initialization files are read ■ TRUE: Only once after I/O reconfiguration ■ FALSE: Every time a channel is opened (default)
Channel		Up to 9 channels are possible; this corresponds to the maximum number of external systems that can be connected
	Sensor-Name	The name of the external system that is connected
	SensorType	No function
TCP/IP	IP	Target address (server)
	Port	Target port (server)
	Route	Defines how the network connection to the external system is established ■ TRUE: with KUKA.Router ■ FALSE: directly via the network card under VxWorks (default)
	MapPort	No function

5.6 Defining an XML structure for receiving data

Description

In order for the robot controller to be able to receive data from the external system, a structure file (XML structure) for receiving data must be defined in the C:\KRC\ROBOTER\INIT directory for each channel.

The name of the file must correspond to the name of the channel or sensor, e.g. SampleSensor.xml.

Example

```
<Elements>
  <Element Tag="ExData" Type="STRUCTTAG" Stacksize="5" />
 <Element Tag="ExData.TString" Type="STRING" Size="80" Stacksize="5"
/>
  <Element Tag="ExData.Position" Type="STRUCTTAG" Stacksize="5" />
  <Element Tag="ExData.Position.XPos" Type="REAL" Stacksize="5" />
  <Element Tag="ExData.Position.YPos" Type="REAL" Stacksize="5" />
  <Element Tag="ExData.Position.ZPos" Type="REAL" Stacksize="5" />
 <Element Tag="ExData.Temperature" Type="STRUCTTAG" Stacksize="5" />
  <Element Tag="ExData.Temperature.Cpu" Type="REAL" Stacksize="5" />
  <Element Tag="ExData.Temperature.Fan" Type="REAL" Stacksize="5" />
  <Element Tag="ExData.Ints" Type="STRUCTTAG" Stacksize="5" />
  <Element Tag="ExData.Ints.AState" Type="INTEGER" Stacksize="5" />
  <Element Tag="ExData.Ints.BState" Type="INTEGER" Stacksize="5" />
  <Element Tag="ExData.Boolean" Type="STRUCTTAG" Stacksize="5" />
  <Element Tag="ExData.Boolean.CState" Type="BOOLEAN" Stacksize="5" /
>
  <Element Tag="ExData.Frames" Type="STRUCTTAG" Stacksize="5" OnTag-
SetPort="1" />
  <Element Tag="ExData.Frames.XFrame" Type="FRAME" Stacksize="10" />
</Elements
```

Parameter	Description
Tag	Name of the tag that is to be written
Type	Data type of the tag that is to be sent Permissible data types are: ■ STRUCTTAG ■ BOOLEAN ■ INTEGER ■ REAL ■ STRING ■ FRAME
Size	In the case of a STRING, the parameter defines the maximum size of the string.
OnTagSetPort	Defines that an output is set every time a tag is received by the robot controller The output must be reset by the KRL program. The output is automatically set as long as tags are being received. In other words, if the output is reset, the output is immediately set again if there are still tags present. This function is only permissible with a STRUCTTAG.
Stacksize	Size of the ring buffer Recommended value: 100

A distinction is made between upper and lower case.

5.7 Defining an XML structure for sending data

Description

A predefined XML structure can be used for sending complex data structures from the robot controller to the external system (>>> 7.6.5 "Sending complex XML structures" page 65).

For this purpose, a structure file (XML structure) for sending data can be created in the C:\KRC\ROBOTER\INIT directory for each channel.

The name of the file must correspond to the name of the channel or sensor and is followed by “+”, e.g. SampleSensor+.xml.

The XML structure can be defined as a mere structure without contents, as is the case in the example. The tags can, however, also be predefined with data.

Example

```
<Robot>
 <Data>
  <ActPos   X="" Y="" Z="" A="" B="" C="" />
  <LastPos  X="" Y="" Z="" A="" B="" C="" />
 </Data>
 <Status />
 <Mode />
 <Complex>
  <Tickcount />
 </Complex>
 <RobotType>
  <Robot>
   <Type></Type>
   <Serial></Serial>
  </Robot>
 </RobotType>
 <Any A="" B="" C="" />
</Robot>
```

5.8 Reconfiguring the I/O driver

Precondition

- User group "Expert".
- Operating mode T1 or T2.

Procedure

Warning!
All outputs are reset!

1. Select the menu sequence **Configure** > **I/O Driver** > **Reconfigure I/O Driver**.
2. Acknowledge messages.

6 Programming

6.1 Communication sequence in KRL

Description

This sequence serves as an example for most KRL programs.

1. The KRL program opens the communications channel to an external system.
2. The KRL program waits for data and informs the external system of this (trigger signal).
3. The KRL program waits until the external system has transferred the data.
4. The KRL program reads the received data from the buffer.

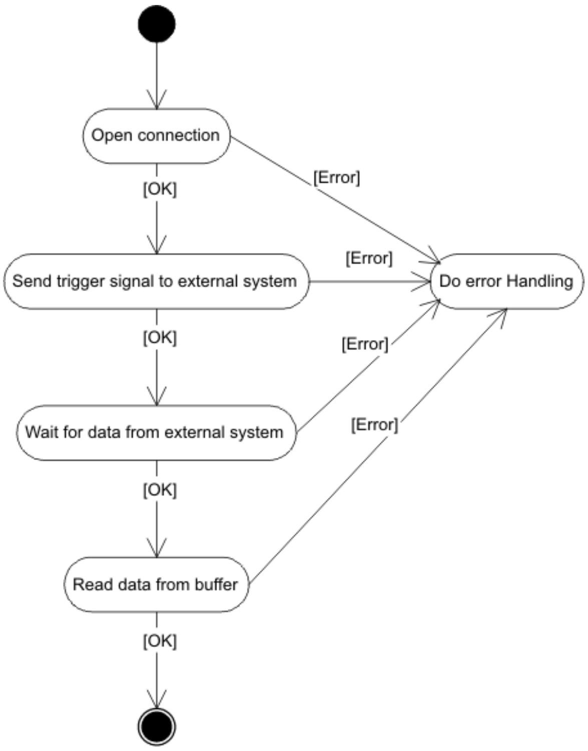


Fig. 6-1: Communication sequence in KRL

6.2 Overview of the functions

Overview

KUKA.Ethernet KRL XML provides the following functions for organizing the exchange of data between the robot controller and an external system:

- **Functions for establishing and terminating a connection**

Functions
INT=EKX_Open(CHAR[])
BOOL=EKX_Close(CHAR[])

Functions in detail: (>>> 6.2.1 "Functions for establishing and terminating a connection" page 52)

- **Send functions**

Functions
INT=EKX_Write(CHAR[], CHAR[])
INT=EKX_WriteLine(CHAR[], CHAR[])
INT=EKX_Send(CHAR[])
INT=EKX_WriteReal(CHAR[], REAL)
INT=EKX_WriteInteger(CHAR[], INTEGER)
INT=EKX_WriteBool(CHAR[], BOOL)
INT=EKX_WriteString(CHAR[], CHAR[])

Functions in detail: (>>> 6.2.2 "Send functions" page 53)

- **Access functions**

Functions
INT=EKX_GetNextElementName(CHAR[], CHAR[])
BOOL=EKX_GetStructElement(INT, CHAR[], BOOL)
BOOL=EKX_GetBoolElement(INT, CHAR[], BOOL, BOOL)
BOOL=EKX_GetIntegerElement(INT, CHAR[], INT, BOOL)
BOOL=EKX_GetRealElement(NT, CHAR[], REAL, BOOL)
BOOL=EKX_GetStringElement(INT, CHAR[], BOOL)
BOOL=EKX_GetFrameElement(INT, CHAR[], FRAME, BOOL)
BOOL=EKX_WaitForSensorData(INT, CHAR[], INT)

Functions in detail: (>>> 6.2.3 "Access functions" page 54)

- **Error handling function**

Function
EKX_HandleError(INT)

Function in detail: (>>> 6.2.4 "Error handling function" page 58)

6.2.1 Functions for establishing and terminating a connection

Description

RET=EKX_Open(CHAR[])	
Function	Opens and initializes a sensor channel This channel must be defined in the file XmlApiCon-fig.xml.
Parameter	CHAR array Name of the channel
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_Close(CHAR[])	
Function	Closes the channel specified in the parameter
Parameter	CHAR array Name of the channel
RET	BOOL, return value TRUE: channel closed successfully FALSE: error on closing channel

6.2.2 Send functions

Description

- Sending strings

RET=EKX_Write(CHAR[], CHAR[])	
Function	Writes a string to an open channel and sends it
Parameter 1	CHAR array Name of the channel
Parameter 2	CHAR array String to be sent
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_WriteLine(CHAR[], CHAR[])	
Function	Writes a string to an open channel and sends it with a line break
Parameter 1	CHAR array Name of the channel
Parameter 2	CHAR array String to be sent
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

- Sending XML structures

RET=EKX_Send(CHAR[])	
Function	Sends a defined XML structure
Parameter 1	CHAR array ■ Name of the channel ■ Name of the channel with position of the XML node
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_WriteReal(CHAR[], REAL)	
Function	Writes a value of type REAL to a defined XML structure
Parameter 1	CHAR array Name of the channel with position of the XML node or with position of the XML attribute
Parameter 2	REAL Value to be sent
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_WriteInteger(CHAR[], INTEGER)	
Function	Writes a value of type INTEGER to a defined XML structure
Parameter 1	CHAR array Name of the channel with position of the XML node or XML attribute
Parameter 2	INTEGER Value to be sent
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_WriteBool(CHAR[], BOOL)	
Function	Writes a value of type BOOL to a defined XML structure
Parameter 1	CHAR array Name of the channel with position of the XML node or XML attribute
Parameter 2	BOOL Value to be sent
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_WriteString(CHAR[], CHAR[])	
Function	Writes a string to a defined XML structure
Parameter 1	CHAR array Name of the channel with position of the XML node or XML attribute
Parameter 2	CHAR array String to be sent
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

6.2.3 Access functions

Description

RET=EKX_GetNextElementName(CHAR[], CHAR[])	
Function	Reads the next string from the buffer The strings are transferred to the robot controller in the order in which they arrive.
Parameter 1	CHAR array Name of the channel
Parameter 2	CHAR array, return value Variable to which the string read from the buffer is written
RET	INTEGER, return value (>>> 6.2.4 "Error handling function" page 58)

RET=EKX_GetStructElement(INT, CHAR[], BOOL)	
Function	Writes in parameter 3 whether the value of type STRUCTTAG has already been read
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. Sensor1.Position.Values
Parameter 3	BOOL, return value TRUE: The read value is new. FALSE: The read value has already been read.
RET	BOOL, return value TRUE: value transmitted successfully FALSE: error during transmission of value

RET=EKX_GetBoolElement(INT, CHAR[], BOOL, BOOL)	
Function	Writes values of type BOOL to parameter 3
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. Sensor1.Position.yesno
Parameter 3	BOOL, return value Variable to which the value read from the buffer is writ-ten
Parameter 4	BOOL, return value TRUE: The read value is new. FALSE: The read value has already been read.
RET	BOOL, return value TRUE: value transmitted successfully FALSE: error during transmission of value

RET=EKX_GetIntegerElement(INT, CHAR[], INT, BOOL)	
Function	Writes values of type INTEGER to parameter 3
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)

RET=EKX_GetIntegerElement(INT, CHAR[], INT, BOOL)	
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. Sensor1.Status.Integer
Parameter 3	INTEGER, return value Variable to which the value read from the buffer is written
Parameter 4	BOOL, return value TRUE: The read value is new. FALSE: The read value has already been read.
RET	BOOL, return value TRUE: value transmitted successfully FALSE: error during transmission of value

RET= EKX_GetRealElement(INT, CHAR[], REAL, BOOL)	
Function	Writes values of type REAL to parameter 3
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. Sensor1.Temperature.Flange
Parameter 3	REAL, return value Variable to which the value read from the buffer is written
Parameter 4	BOOL, return value TRUE: The read value is new. FALSE: The read value has already been read.
RET	BOOL, return value TRUE: value transmitted successfully FALSE: error during transmission of value

RET=EKX_GetStringElement(INT, CHAR[], BOOL)	
Function	Writes strings to parameter 2
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. Sensor1.Weekday The name of the element is overwritten with the value read from the buffer.

RET=EKX_GetStringElement(INT, CHAR[], BOOL)	
Parameter 3	BOOL, return value TRUE: The read value is new. FALSE: The read value has already been read.
RET	BOOL, return value TRUE: value transmitted successfully FALSE: error during transmission of value

RET=EKX_GetFrameElement(INT, CHAR[], FRAME, BOOL)	
Function	Writes values of type FRAME to parameter 3
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. Sensor1.Position.Tool
Parameter 3	FRAME, return value Variable to which the value read from the buffer is written
Parameter 4	BOOL, return value TRUE: The read value is new. FALSE: The read value has already been read.
RET	BOOL, return value TRUE: value transmitted successfully FALSE: error during transmission of value

RET=EKX_WaitForSensorData(INT, CHAR[], INT)	
Function	Waits for a new set of sensor data, i.e. an XML structure with new data
Parameter 1	INTEGER Determines whether the next value or the current value is to be read from the buffer INT=0: the next value is read (First In First Out) INT=1: the current value is read (Last In First Out)
Parameter 2	CHAR array The name of the element preceded by the name of the channel, e.g. SampleSensor.ExternalData

RET=EKX_WaitForSensorData(INT, CHAR[], INT)	
Parameter 3	INTEGER Timeout in milliseconds
RET	BOOL, return value TRUE: The end tag of the XML structure has been transferred. FALSE: error during data transfer: ■ timeout has been exceeded ■ an element not defined in the XML structure has been read

6.2.4 Error handling function

Description

EKX_HandleError(INT)	
Function	Handles possible errors when a channel is opened The cause of the error is displayed in the message window.
Parameter	INTEGER Return value to be handled for the following functions: ■ EKX_Open(CHAR[]) ■ EKX_Write(CHAR[], CHAR[]) ■ EKX_WriteLine(CHAR[], CHAR[]) ■ EKX_WriteReal(CHAR[], REAL) ■ EKX_WriteInteger(CHAR[], INTEGER) ■ EKX_WriteBool(CHAR[], BOOL) ■ EKX_WriteString(CHAR[], CHAR[]) ■ EKX_Send(CHAR[]) ■ EKX_GetNextElementName(CHAR[], CHAR[])

7 Example

7.1 Sample application

Description

KUKA.Ethernet KRL XML includes a sample application that can be used to establish communication between a server program and the robot controller. The software is located on the CD-ROM, in the C:\EthernetKRLXML\Demo directory.

The application consists of the following components:

- Server program Server_EKX.exe
- XML files for the external system
- KRL program EKXDemo.src
- Sample source code in C#

7.2 Implementing the sample application

Procedure

1. Copy the contents of the EthernetKRLXML\Demo\Server_app directory to a Windows system with installed .NET Framework.

> The current version of Windows .NET Framework can be obtained free of charge from: http://www.microsoft.com/downloads/.

2. Copy the KRL program ERXDemo.src in the EthernetKRLXML\Demo\SRC_KRL\Program directory to the C:\KRC\ROBOTER\Program directory of the robot controller via the user interface.
3. Copy the initialization files in the directory EthernetKRLXML\Demo\SRC_KRL\INIT to the directory C:\KRC\ROBOTER\INIT of the robot controller.
4. Start the server program Server_EKX.exe on the external system.
5. Press the **Listen!** button (>>> Fig. 7-1) to display the IP address of the server program.
6. Replace the IP address in **XmlApiConfig.xml** with the IP address of the server program in order to establish a connection to the external system.
7. Reboot the robot controller.

> If no external system is available, communication can be carried out via the "Shared Memory" of the robot controller. In the server application, the network adapter (NetcardIndex) is set in such a way that the network address "192.0.1.2" is displayed.
> In this case, the server program Server_EKX.exe is started in the Windows interface of the robot controller.

7.3 Server program Server_EKX.exe

Description

The server program Server_EKX.exe makes it possible to test the connection between the external system and the robot controller by establishing permanent communication to the robot controller.

Functions:

- Stable communication: sending and receiving of data
- Display of the received data
- Display of the transmitted data

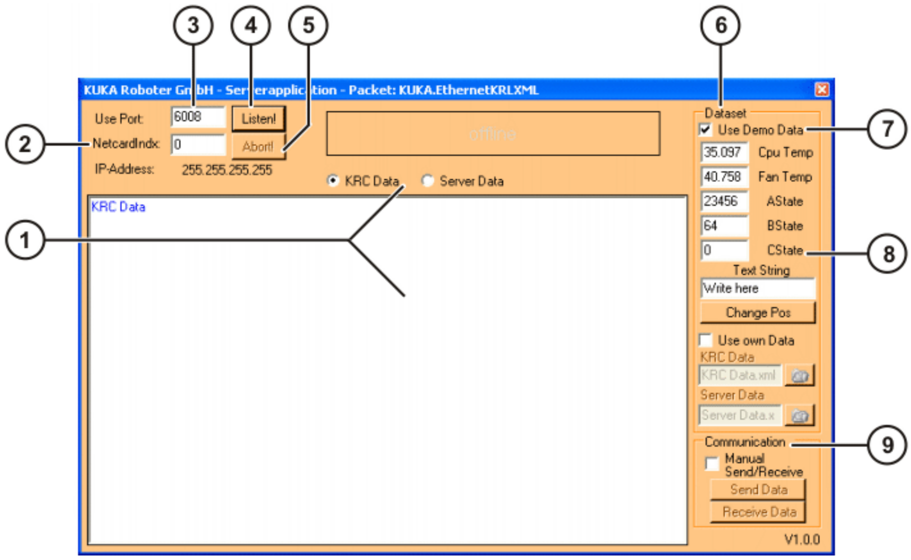


Fig. 7-1: Server program

Item	Description
1	Display box ■ If the radio button **KRC Data** is selected, the data that have just been received are displayed ■ If the radio button **Server Data** is selected, the data that have just been sent are displayed
2	The **NetcardIndex** box refers to the numbering of the system network adapter that has been found.
3	The port used for the socket connection is set in the **Use Port** box. The server computer awaits the connection request from the robot controller at this port. A free number that is not assigned as a standard service must be selected.
4	The **Listen!** button is used to switch the program to listen mode. The first incoming connection request is connected and used as a communications adapter. The **Listen!** button is also used to display the IP address of the server program.
5	If the **Abort!** button is pressed, the program immediately terminates the communication and resets the server.
6	The data sets to be used can be selected in the **Dataset** group. ■ Data set of the KRL program EKX.Demo.src ■ Own data sets

Item	Description
7	If the check box **Use Demo Data** is activated, the program communicates with the data defined in the KRL program EKX-Demo.src. The KRL program obtains the information about the received data from the definition of the XML structure (>>> 5.6 "Defining an XML structure for receiving data" page 47). The communication is carried out in so-called ping pong mode, i.e. both sides respond to received data. The data set can be changed in the input boxes beneath the **Use Demo Data** check box. If the user is using his own data sets, the **Use Own Data** check box must be activated. There are then 2 functions available to the user for importing sent and received data. The information about the data sent and received must be available to the server program as an XML structure.
8	The KRL program reacts sensitively to the variable ‘CState’. ‘CState’ is interpreted as a Boolean value in every cycle. ■ ‘CState’ = 0: A new send/receive cycle is being started. ■ ‘CState’ = 1: It is interpreted as TRUE and the KRL program is terminated.
9	Ping pong mode can be deactivated in the **Communication** group. This makes it possible to wait for or send a data packet manually. The forms of the data set used are subject to this mode.

7.4 KRL program EKXDemo.src

Description

The KRL program EKXDemo.src is a simple KRL structure for sending and receiving data. A data set is formed and transferred to the server system in conformity with XML. A timer for the number of data packets sent is integrated into the data set.

The received data are transferred to local variables and not evaluated further.

7.5 Sample source code for server application

Description

Sample source code, written in the C# programming language, can be found in the directory EthernetKRLXML\Demo\SRC_Server.

This module illustrates the programming of a network connection to the robot controller. Simple integration is possible, for example, with a console project. For this, the member function “private static void anyfunction()” must be called.

The program generates a second process which communicates with the controller, independently of the application.

Port 6008 and network card index 0 are set by default. The data to be sent are loaded via the XML model class. The file ExternalData.xml must then be inserted into the project; this file can be found in the directory of the server program Server_EKX.exe.

7.6 Example of programming with EKXDemo.src

Overview

Section	Description
1	Opening a communications channel (>>> 7.6.1 "Opening a communications channel" page 62)
2	Structure of the data sets (>>> 7.6.2 "Structure of the data sets" page 63)
3	Sending a character string directly (>>> 7.6.3 "Sending character strings directly" page 63)
4	Waiting for data, reading data types, evaluating return values (>>> 7.6.4 "Waiting for data, reading data types, evaluating return values" page 64)
5	Sending complex XML structures (>>> 7.6.5 "Sending complex XML structures" page 65)
6	Closing a communications channel (>>> 7.6.6 "Closing a communications channel" page 65)

7.6.1 Opening a communications channel

Program

```
1   sensorname[]="SampleSensor"
2   errint = EKX_open(sensorname[])
3   EKX_handleerror(errint)
```

Description

Line	Description
1	Sensor name
2	The communications channel is opened
3	Evaluation of the return value of the function EKX_open If an error value is returned, the KRL program is stopped and an error message is generated in the message window.

7.6.2 Structure of the data sets

Program

```
1  sendtext[1].s[] = "<Rob Type=~KUKA~>"
2  sendtext[2].s[] = "<Data>"
3  sendtext[3].s[] = "<ComStatus>continuous</ComStatus>"
4  sendtext[4].s[] = "<Tick>"
5  sendtext[5].s[] = "0"
6  sendtext[6].s[] = "</Tick>"
7  sendtext[7].s[] = "</Data>"
8  sendtext[8].s[] = "</Rob>"
9  rectext[1].s[] = "SampleSensor.ExternalData.TString"
10 rectext[2].s[] = "SampleSensor.ExternalData.Position.XPos"
11 rectext[3].s[] = "SampleSensor.ExternalData.Position.YPos"
12 rectext[4].s[] = "SampleSensor.ExternalData.Position.ZPos"
13 rectext[5].s[] = "SampleSensor.ExternalData.Temperature.Cpu"
14 rectext[6].s[] = "SampleSensor.ExternalData.Temperature.Fan"
15 rectext[7].s[] = "SampleSensor.ExternalData.Ints.AState"
16 rectext[8].s[] = "SampleSensor.ExternalData.Ints.BState"
17 rectext[9].s[] = "SampleSensor.ExternalData.Boolean.CState"
18 rectext[10].s[] = "SampleSensor.ExternalData"
19 counter = 0
20 cstate = false
```

Description

Line	Description
1...8	Definition of the XML structure to be sent Within KRL strings, quotation marks (") must be replaced with tildes (~). These are converted back to quotations marks on transmission.
9...18	Definition of the XML tags to be read

7.6.3 Sending character strings directly

Program

```
1  FOR i=1 TO 3
2  errint = EKX_writeline(sensorname[], sendtext[i].s[])
3  IF errint == eioc_error THEN
4    HALT
5  ENDIF
6  ENDFOR
7  errint = EKX_write(sensorname[], sendtext[4].s[])
8  IF errint == eioc_error THEN
9    HALT
10 ENDIF
11 errint = EKX_write(sensorname[], sendtext[5].s[])
12 IF errint == eioc_error THEN
13   HALT
14 ENDIF
15 FOR i=6 TO 8
16 errint = EKX_writeline(sensorname[], sendtext[i].s[])
17 IF errint == eioc_error THEN
18   HALT
19 ENDIF
20 ENDFOR
21 counter = counter + 1
22 sendtext[5].s[] = "0"
23 offset = 0
24 SWRITE(sendtext[5].s[], state,offset,"%d",counter)
```

Description

Line	Description
1...2	Lines 1 to 3 of the XML structure are transmitted with carriage returns
3...6	The program is stopped if an error occurs.
7...11	Lines 4 and 5 of the XML structure are transmitted without carriage returns, i.e. the strings are run together into one line

Line	Description
15...16	Lines 6 to 8 of the XML structure are transmitted with carriage returns
21	The communication cycle counter is incremented

7.6.4 Waiting for data, reading data types, evaluating return values

Program

```
1   errbl = EKX_WaitForSensorData(0, rectext[11].s[], 10000)
2   IF errbl == FALSE THEN
3     HALT
4   ENDIF
5   ; get string
6   dummy = rectext[1]
7   errbl = EKX_GetStringElement(0, dummy.s[], bNew)
8   IF errbl == FALSE THEN
9     HALT
10  ENDIF
11  ;get position
12  FOR i=1 TO 3
13  errbl = EKX_GetRealElement(0, rectext[1+i].s[], position[i], bNew)
14  ...
15  ; get Ints
16  errbl = EKX_GetIntegerElement(0, rectext[7].s[], astate, bNew)
17  IF errbl == FALSE THEN
18    HALT
19  ENDIF
20  ...
21  ; get bool
22  errbl = EKX_GetBoolElement(0, rectext[9].s[], cstate, bNew)
23  IF errbl == FALSE THEN
24    HALT
25  ENDIF
26  ; get frame until the stack is empty!
27  bNew = TRUE
28  WHILE bNew == TRUE
29   errbl = EKX_GetFrameElement( 0,  rectext[10].s[], extframe, bNew)
30  IF errbl == FALSE THEN
31    HALT
32  ENDIF
33    ;FOLD Holdon
34    IF holdon == TRUE THEN
35     HALT
36    ENDIF
37    ;ENDFOLD
38    ; Take a look in the view of variables and show extframe
39    ; Recognise the OUT[1] is setted
40  ENDWHILE
```

Description

Line	Description
1	Wait for a set of new sensor data
2...4	The program is stopped if an error occurs. Possible causes of errors when waiting for data: ■ Channel not open ■ Sensor name not available ■ Buffer name not available ■ Buffer name is not a STRUCTTAG buffer ■ Timeout has been exceeded
5...7	A string is read from the buffer

Line	Description
8...10	The program is stopped if an error occurs. Possible causes of errors when reading data from the buffer: ■ Channel not open ■ Sensor name not available ■ Buffer name not available ■ Incorrect access function: data type does not match
11...13	Values of type REAL are read from the buffer
15...16	A value of type INTEGER is read from the buffer
21...22	A value of type BOOL is read from the buffer
26...29	A value of type FRAME is read from the buffer Values continue being read from the frame until it no longer contains new data.

7.6.5 Sending complex XML structures

Program

```
1   errint = EKX_WriteReal("SampleSensor.Robot.Data.ActPos.X",
$pos_act.x)
2   EKX_handleerror(errint)
3
4   ...
5
6   errint = EKX_WriteString("SampleSensor.Robot.Data.LastPos.X",
"0.1")
7   EKX_handleerror(errint)
8   errint = EKX_WriteInteger("SampleSensor.Robot.Data.LastPos.Y",
15)
9   EKX_handleerror(errint)
10
11  ...
12
13  errint = EKX_WriteBool("SampleSensor.Robot.Mode", $peri_rdy)
14  EKX_handleerror(errint)
15
16  ...
17
18  ; Send document
19  errint = EKX_Send("SampleSensor")
20  EKX_handleerror(errint)
21
22  ...
23
24  errint = EKX_Send("SampleSensor.Robot.RobotType.Robot")
25  EKX_handleerror(errint)
```

Description

Line	Description
1...14	Values are assigned to the XML structure
19...20	The complete XML structure is sent
24...25	Part of the XML structure is sent

7.6.6 Closing a communications channel

Program

```
errbl = EKX_close(sensorname[])
IF errbl == false THEN
  HALT
ENDIF
```

Description

Line	Description
1	The communications channel is closed
2...4	Evaluation of the return value of the function EKX_close If an error value is returned, the KRL program is stopped and an error message is generated in the message window.

8 Diagnosis

8.1 Diagnosis with Telnet

Description

Telnet can be used to check the configuration and communication with Vx-Works.

Procedure

1. Click on the Windows **Start** button.
2. Select the menu option **Run...**.
3. In the **Open** box, enter the command **telnetk 192.0.1.1** and press OK. The Telnet window is opened.

Displaying the IP address

The IP address can be checked using the **version** command. The address is displayed in the "Boot line" under **e=...**.

> It is only possible to display the IP address with Telnet if using KUKA System Software (KSS) 5.x.

```
-> version
VxWorks (for VxWin RTAcc) version 5.4.2.
Kernel: WIND version 2.5.
Made on Mar  7 2005, 11:13:54.
Boot line:
esmc(0,1)pc:vxworks h=192.0.1.2 b=192.0.1.1 e=160.160.62.118 u=target
pw=vxworks
value = 92 = 0x5c = '\'
```

Testing the network card

The command **ping xxx.xxx.xxx.xxx** can be used to check the communication of the VxWorks system network card with the remote station. The command can be aborted by closing the Telnet window.

- Connection present:

```
-> ping "192.0.1.2"
PING 192.0.1.2: 56 data bytes
64 bytes from pc (192.0.1.2): icmp_seq=0. time=0. ms
64 bytes from pc (192.0.1.2): icmp_seq=1. time=0. ms
64 bytes from pc (192.0.1.2): icmp_seq=2. time=0. ms
64 bytes from pc (192.0.1.2): icmp_seq=3. time=0. ms
...
```

- No connection:

```
-> ping "123.123.45.2"
PING 123.123.45.2: 56 data bytes
no answer from 123.123.45.2
```

9 KUKA Service

9.1 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.

Faults leading to production downtime should be reported to the local KUKA subsidiary within one hour of their occurrence.

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)
- Version of the KUKA System Software
- Optional software or modifications
- Archive of the software
- Application used
- Any external axes used
- Description of the problem, duration and frequency of the fault

9.2 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

Marand Precision Engineering Pty. Ltd. (Agency)
153 Keys Road
Moorabbin
Victoria 31 89
Australia
Tel. +61 3 8552-0600
Fax +61 3 8552-0605
robotics@marand.com.au

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
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 Flexible Manufacturing Equipment (Shanghai) Co., Ltd. Shanghai Qingpu Industrial Zone No. 502 Tianying Rd. 201712 Shanghai P.R. China Tel. +86 21 5922-8652 Fax +86 21 5922-8538 Franz.Poeckl@kuka-sha.com.cn www.kuka.cn

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
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
Hungary	KUKA Robotics Hungaria Kft. Fö út 140 2335 Taksony Hungary Tel. +36 24 501609 Fax +36 24 477031 info@kuka-robotics.hu
India	KUKA Robotics, Private Limited 621 Galleria Towers DLF Phase IV 122 002 Gurgaon Haryana India Tel. +91 124 4148574 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
Tokio
135-0091
Japan
Tel. +81 3 6380-7311
Fax +81 3 6380-7312
info@kuka.co.jp

Korea

KUKA Robot Automation Korea Co. Ltd.
4 Ba 806 Sihwa Ind. Complex
Sung-Gok Dong, Ansan City
Kyunggi Do
425-110
Korea
Tel. +82 31 496-9937 or -9938
Fax +82 31 496-9939
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
Bryggeveien 9
2821 Gjövik
Norway
Tel. +47 61 133422
Fax +47 61 186200
geir.ulsrud@kuka.no

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
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
Spain	KUKA Sistemas de Automatización S.A. Pol. Industrial Torrent de la Pastera Carrer del Bages s/n 08800 Vilanova i la Geltrú (Barcelona) Spain Tel. +34 93 814-2353 Fax +34 93 814-2950 Comercial@kuka-e.com www.kuka-e.com
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 Riedstr. 7 8953 Dietikon Switzerland Tel. +41 44 74490-90 Fax +41 44 74490-91 info@kuka-roboter.ch www.kuka-roboter.ch
Taiwan	KUKA Robot Automation Taiwan Co. Ltd. 136, Section 2, Huanjung E. Road Jungli City, Taoyuan Taiwan 320 Tel. +886 3 4371902 Fax +886 3 2830023 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
UK	KUKA Automation + Robotics Hereward Rise Halesowen B62 8AN UK Tel. +44 121 585-0800 Fax +44 121 585-0900 sales@kuka.co.uk
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

Index