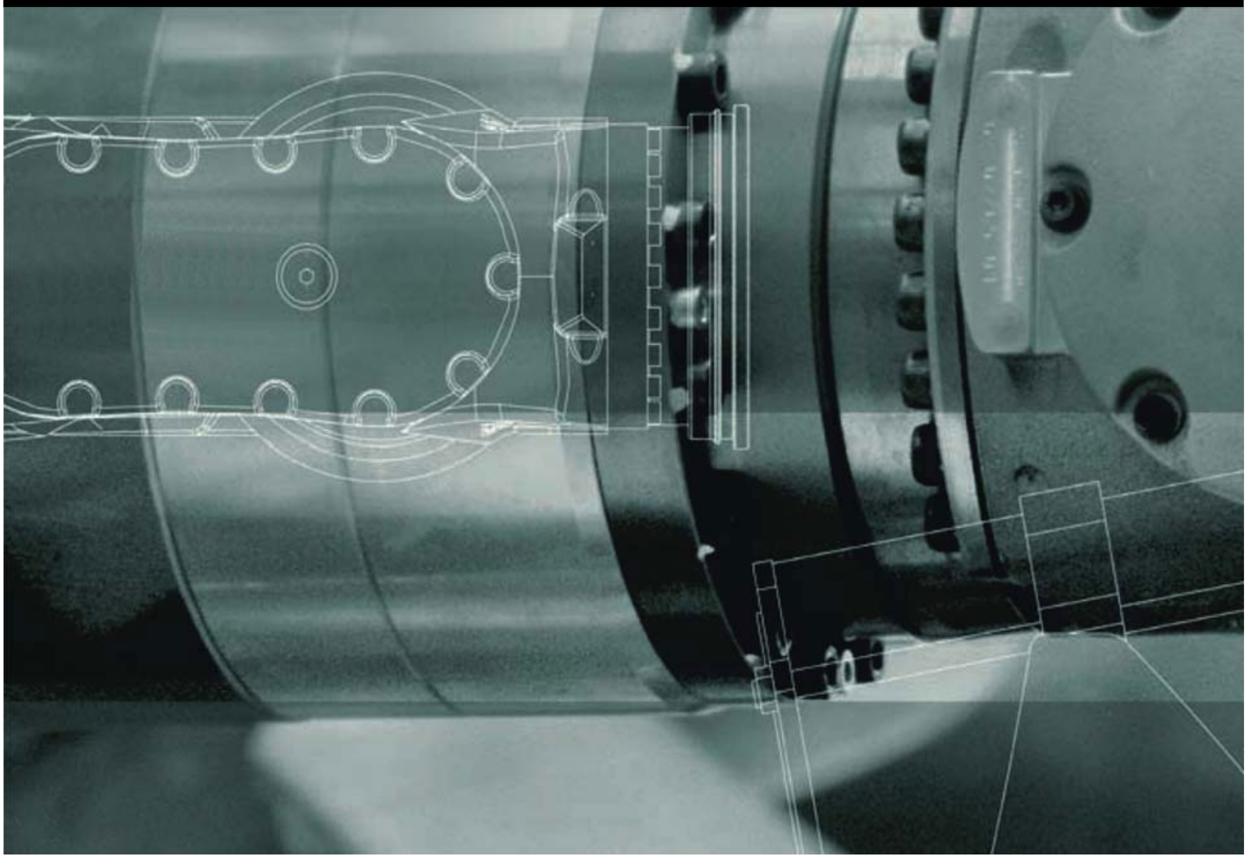
KUKA System Technology

KUKA Roboter GmbH

KUKA.FastResearchInterface 1.0

For KUKA System Software 5.6 lr



Issued: 16.09.2011

Version: KUKA.FRI 1.0 V2 en

KUKA Roboter GmbH
Zugspitzstraße 140
D-86165 Augsburg
Germany

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

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

Subject to technical alterations without an effect on the function.

Translation of the original documentation

KIM-PS5-DOC

Publication:	Pub KUKA.FRI 1.0 en
Bookstructure:	KUKA.FRI 1.0 V2.1
Version:	KUKA.FRI 1.0 V2 en

Contents

1 Introduction

1.1 Industrial robot documentation

The industrial robot documentation consists of the following parts:

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

Each of these sets of instructions is a separate document.

1.2 Representation of warnings and notes

Safety

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

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

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

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

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

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

Hints

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

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

1.3 Terms used

Term	Description
CIRC	Circular motion
FRI	Fast Research Interface
Host	Defined by an IP address and port numbers.
KCP	KUKA Control Panel
KR C	KUKA Robot Controller
KRL	KUKA Robot Language
LWR	Lightweight robot
LIN	Linear motion
PTP	Point-to-point motion

Term	Description
SDK	Software Development Kit
Socket	Software interface for Ethernet connections.
UDP	User Datagram Protocol Ethernet protocol via sockets.
Watchdog	Functionality that monitors the max. allocated processing/response time.

1.4 Trademarks

VxWorks is a trademark of Wind River Systems Inc.

Windows is a trademark of Microsoft Corporation.

2 Purpose

2.1 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 network connections
- Basic knowledge of C++ programming
- Knowledge of the “Configuration” and “System variables” chapters in the operating and programming instructions of the KUKA System Software (KSS) for the LWR.

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.

2.2 Intended use

Use

FRI is a software package for experiments with the LWR in research laboratories.

Operation in accordance with the intended use also involves continuous observance of the following documentation:

- Assembly or operating instructions of the robot
- Assembly or operating instructions of the robot controller
- Operating and programming instructions of the KUKA System Software for system integrators

Misuse

Any use or application of the LWR deviating from the intended 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
- Operation without additional safeguards
- Outdoor operation
- Leaning on the robot arm

NOTICE Any application of the FRI other than that described in this documentation could result in damage to the LWR. This nullifies guarantee and liability claims relating to the LWR.

3 Product description

3.1 Overview of KUKA.FastResearchInterface

Functions

KUKA.FastResearchInterface is an add-on software option with the following functions:

- Configuration of a cycle time for data transfer: range from 1 ... 100 ms
- Configuration of signal processing in the real-time system
- Monitoring of the robot motions from an external computer
- Data access via an external computer
- Transfer of robot controller messages to an external computer
- Reading and writing of system and program variables of the LWR
- Superposition of paths
- Execution of customer programs from the external computer

Properties

- Monitor mode
- Command mode
- Interaction with KRL programs
- UDP socket communication with integrated SDK
- Binary data transfer

Field of application

- Experiments in the research laboratory with the LWR

3.2 FRI modes

Description

Data transfer between the robot controller and an external computer is carried out using the following modes:

- Monitor mode

  Cyclical communication with transfer of robot data to an external computer.
- Command mode

  Cyclical communication with transmission of commands from an external computer to the robot controller.

  In Command mode, robot motion can also be controlled externally.

The following figure shows the interaction of these modes:

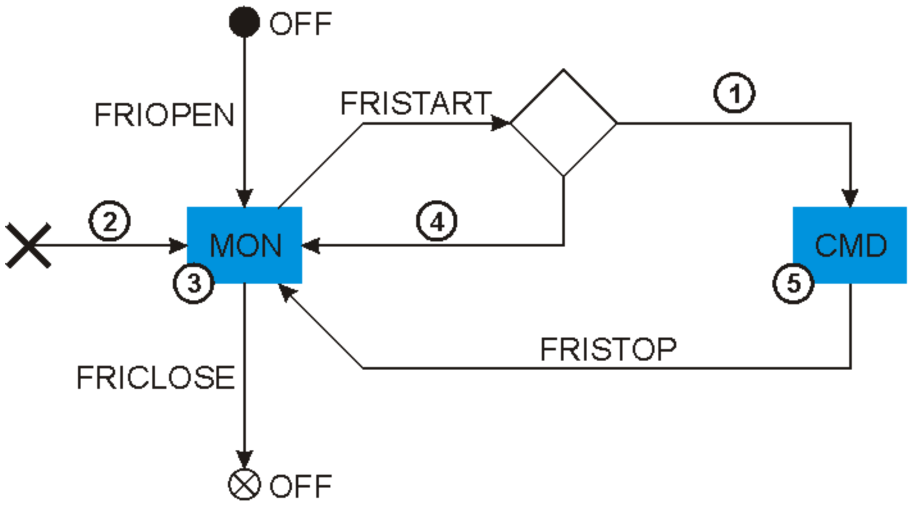


Fig. 3-1: Status graph for Monitor and Command modes

1 Data packet set in the specified timeframe
2 Fault
3 Monitor mode (MON)
4 Quality of the data transfer is not sufficient
5 Command mode (CMD)

The state can be polled at any time using the system variable `$FRISTATE`.

```
ENUM FRISTATE OFF, MON, CMD, INVALID
DECL FRISTATE $FRISTATE
```

Functional principle

FRIOPEN activates the FRI connection between the robot controller and the external computer. At this point in time Monitor mode is active. In this mode, there is cyclical data exchange between the KRL and the robot. The quality of the data transfer is also determined and classified.

FRISTART starts Command mode if the quality of the data transfer has been classified as “Good” or “Perfect” (>>> 3.3.1 "UDP timing quality" Page 11). Data exchange between the robot controller and the robot and determination of the quality of the data transfer continue. In addition, it is now possible for robot motions to be superposed and for programs created by the customer to be executed from an external computer.

FRISTOP stops Command mode and switches to Monitor mode. At this point in time, selected Start-Stop (FRISTART, FRISTOP) loops can be executed.

FRICLOSE closes the FRI connection. The connection is closed completely.

If a fault occurs in Command mode, Command mode is deactivated immediately and the FRI switches to Monitor mode. The switch can be triggered by the following situations:

- The quality of the data transfer is not sufficient (“Good” or “Perfect”).
- More than one UDP data packet has been lost.
- The sequence count has been violated, i.e. the UDP data packets are not in the correct sequence, for example.
- The command values are outside the limit values, e.g. positions, velocity, acceleration and/or torque limits have been exceeded.

3.3 UDP communication

Description

The external computer must provide a socket server for UDP communication. Data exchange between the robot controller and an external computer is implemented with the 2 following data telegrams:

- tFriMsrData
- tFriCmdData

Further information about the data telegrams can be found in the chapter “Programming”.

Both data telegrams contain a header that ensures communication between the robot controller and external computer. The header contains the following information:

- Sequence count mechanism
- Binary data size of the data packet
- Data structure ID

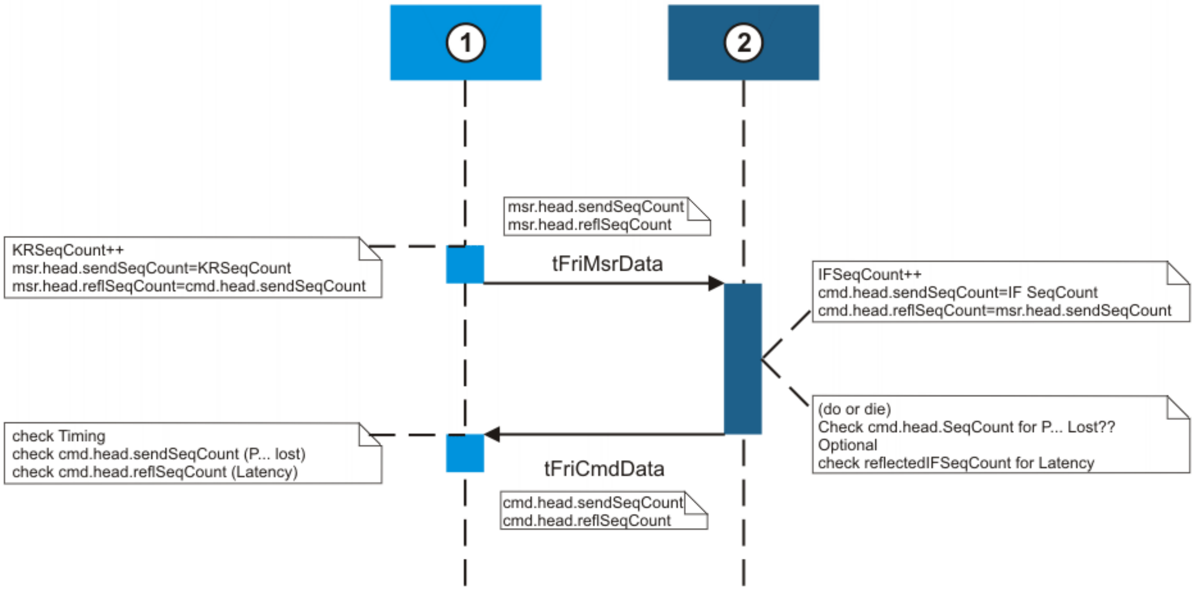


Fig. 3-2: Interaction of the data telegrams, example

1	Robot controller	2	External computer

During the handshake between the robot controller and the external computer, the sequence counts are carried out on both sides. The data packet loss and latency time are determined on the basis of the sequence counts on both sides. For this reason, the external computer must reflect the sequence counts of the KRC side.

3.3.1 UDP timing quality

The quality of the data transfer is classified as follows:

- Perfect
- Good
- Unacceptable
- Bad

The quality of the connection can be polled using the system variable `$FRIQUALITY`. The command `FRISHOW[]` can be used for a more detailed analysis.

```
ENUM FRIQUALITY UNACCEPTABLE, BAD, OK, PERFECT, INVALID
DECL FRIQUALITY $FRIQUALITY
```

It is possible to switch to Command mode with FRISTART when $FRIQUALITY has acquired the status “OK”.

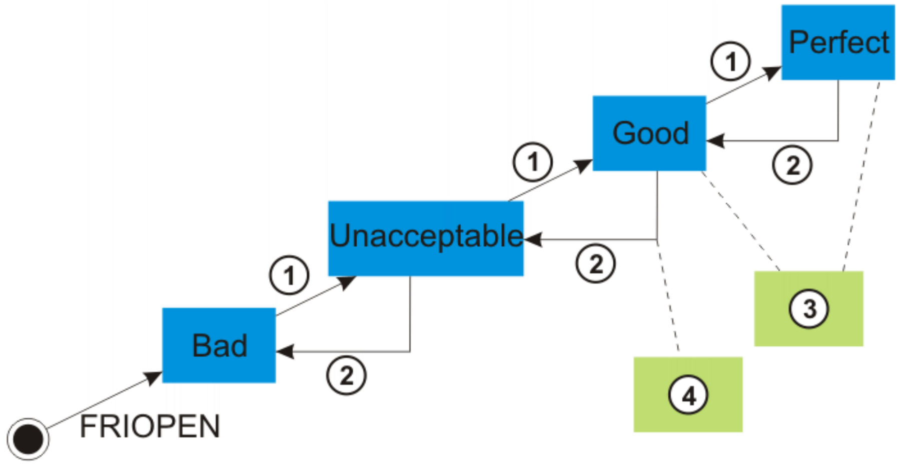


Fig. 3-3: Quality of the data transfer

1 Handshaking is carried out
2 UPD data packet lost
3 Command mode can be used
4 Command mode is deactivated; Monitor mode is activated

When the FRI connection is first established, the quality of the data transfer is always classified as “Bad”. A successful handshake improves the quality of the connection from “Bad” to “Unacceptable” and then from “Good” to “Perfect”.

Commands from the external computer to the robot controller are only accepted if the quality of the data transfer is “Perfect” or “Good”.

If, during operation in Command mode, the quality of the data transfer is classified as “Unacceptable” or “Bad”, Command mode is deactivated and Monitor mode is activated. Additionally, a message is generated.

4 Safety

This documentation contains safety instructions which refer specifically to the product described here. The fundamental safety information for the industrial robot can be found in the “Safety” chapter of the following documentation:

- Operating or assembly instructions of the robot
- Operating and programming instructions for system integrators

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

5 Installation

5.1 System requirements

Hardware

- KR C2 lr robot controller
- 3COM or Intel PRO 1000/GT network card (Ethernet card)
- External computer: processor-supported system with network card
- Network cable for direct connection
  Recommendation: direct connection to minimize latency time.

Software

Robot controller:

- KUKA System Software 5.6 lr

External system

- Communication via UDP protocol

5.1.1 PCI slot assignment

Overview

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

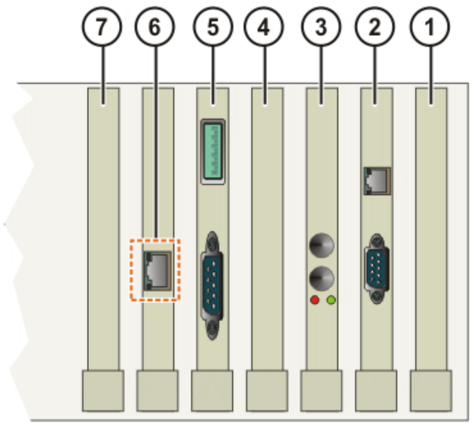


Fig. 5-1: PCI slots on the KR C2 lr controller hardware

Item	Slot	Plug-in card
1	1	Profibus master/slave card (optional)
2	2	KVGA card
3	3	Sercos FOC card
4	4	free
5	5	MFC3 card
6	6	Ethernet card ■ 3COM card or ■ Intel PRO 1000/GT
7	7	free

5.2 Installing or updating FastResearchInterface (KSS)

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

Precondition

- "Expert" user group

- Windows interface (CTRL+ESC)

Procedure

1. Connect KUKA USB stick.

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

2. Start the program **Setup.exe** from the USB stick. The files are copied onto the hard drive.
3. Confirm the reboot prompt with **OK**.
4. Reboot the robot controller. The installation is resumed.

   The window **KUKA - Kernel System Network Setup** is opened.
5. Enter the IP address of the operating system in the **KUKA - Kernel System Network Setup** window and confirm with **OK**.

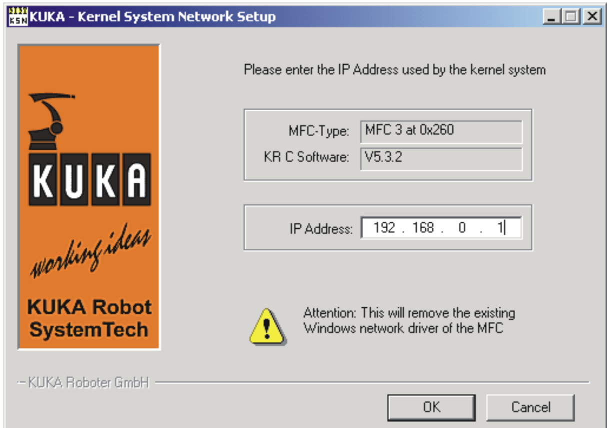


Fig. 5-2: “KUKA - Kernel System Network Setup” window

6. The installation is completed.

LOG file

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

5.3 Installing or updating FastResearchInterface (external computer)

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

Precondition

- Windows interface (CTRL+ESC)

Procedure

1. Connect the KUKA USB stick.

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

2. Copy the FRI_RemoteSample folder to the external computer.

5.4 Creating a license request

A license key is required for licensing KUKA.FastResearchInterface.

To obtain the license key it is necessary to contact KUKA Roboter GmbH.

5.5 Activating FRI

Preconditions

- FRI is installed.

- 3COM or Intel PRO 1000/GT network card (Ethernet card) is installed.
- A license key is available (>>> 5.4 "Creating a license request" Page 16).

Procedure

1. Open the file DLRRC.INI in the directory C:\KRC\ROBOTER\INIT using a text editor.
2. After the line 'IMMEDIATE_STARTUP=1', insert the following lines (>>> Fig. 5-3):
   - `FRIHOST=xxx.xxx.xx.xx`
     Specify the IP address of the host PC
   - `FRISOCK=49938.0`
     Specify the port address of the host PC (default value: 49938.0)
   - `FRIKEY=xx:xxxxx:xxxxxxx`
     Specify the FRI license key
3. Reboot the robot controller.
4. Select the **FRI** status key in the left-hand status key bar.

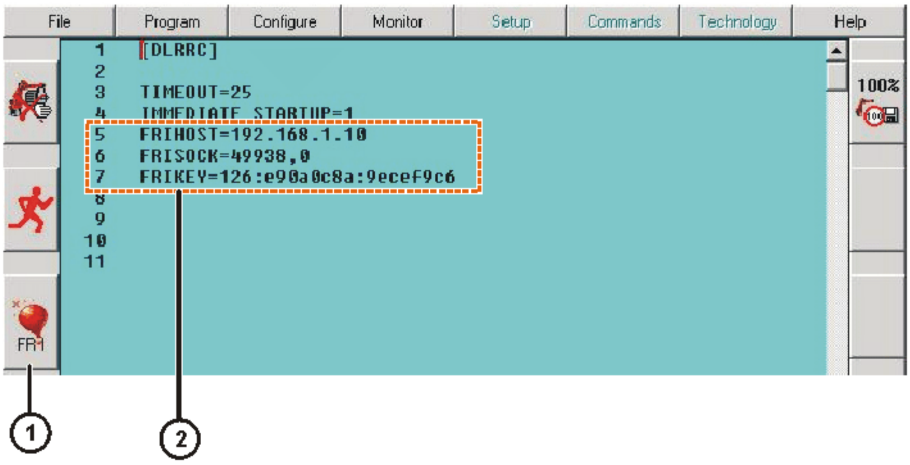


Fig. 5-3: Activating FRI

1	**FRI** status key	2	Lines: FRIHOST, FRISOCK, FRIKEY

5. Select the **FRISHOW** softkey.
6. If the license key is valid, the message **st: not initialized** is displayed.
   If the license key is invalid, the message **NO FRI License** is displayed.

5.6 Uninstalling FastResearchInterface (KSS)

Precondition

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

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

Procedure

1. Start the UnInstall.EXE program in the directory C:\KRC_OPTION\FRI\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.

5.7 Uninstalling FastResearchInterface (external computer)

Precondition

- Windows interface (CTRL+ESC)

Procedure

1. Delete the FRI_RemoteSample folder on the external computer.
2. Reboot the external computer.

6 Configuration

6.1 Configuring the field bus

Description

Customer-defined variables for configuration of the I/O system can be mapped directly to the I/O system without the need for a connection to the robot controller.

Further information is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

The following configurations are available:

- Static field bus configuration with the file DIrrc.INI
  (>>> 6.1.1 "Static field bus configuration" Page 19)
- Dynamic field bus configuration during the runtime
  (>>> 6.1.2 "Dynamic field bus configuration" Page 20)

6.1.1 Static field bus configuration

Description

The following parameters are permissible in the file DIrrc.INI for I/O mapping:

Parameter	Field bus variable	FRI variable
FRIMAPANAIN	$ANAIN[]	$FRI_TO_REA[]
FRIMAPANAOUT	$ANAOUT[]	$FRI_FRM_REA[]
FRIMAPDIGIN	$DIGIN[]	$FRI_TO_INT[]
FRIMAPDIGOUT	$DIGOUT[]	$FRI_FRM_INT[]
FRIMAPBOOLIN	$IN[ ]	$FRI_TO_BOOL[]
FRIMAPBOOLOUT	$OUT[]	$FRI_FRM_BOOL[]

These parameters have the following syntax:

```
<keyword>=<int1><int2><int3><int4><int5><int6>
                <int7><int8><int9><int10><int11><int12>
                <int13><int14><int15><int16>
```

The 16 INTEGER values stand for the indices of the field bus I/O numbers. If an INTEGER value is 0 (zero), no mapping is carried out.

Example

The configuration:

```
FRIMAPANAIN=0 0 13 14 15 16 17 18 19 0 0 0 0 0 0 0
```

Itemized mapping:

```
$FRI_TO_REA[3]= $ANAIN[13]
$FRI_TO_REA[4]= $ANAIN[14]
$FRI_TO_REA[5]= $ANAIN[15]
$FRI_TO_REA[6]= $ANAIN[16]
$FRI_TO_REA[7]= $ANAIN[17]
$FRI_TO_REA[8]= $ANAIN[18]
$FRI_TO_REA[9]= $ANAIN[19]
```

Exactly 16 INTEGER values must be written, with exactly one space between them.
INTEGER value 0 (zero) means that no mapping is carried out.

The update rate for this mapping can be configured with the password `FRIMAPDELAY=<int>`. By default, the update rate is 4 ms.

6.1.2 Dynamic field bus configuration

Description

If the static field bus configuration is modified during the runtime, the following KRL commands are available:

KRL command	Field bus variable	FRI variable
friMapAnaIn	$ANAIN[]	$FRI_TO_REA[]
friMapAnaOut	$ANAOUT[]	$FRI_FRM_REA[]
friMapDigIn	$DIGIN[]	$FRI_TO_INT[]
friMapDigOut	$DIGOUT[]	$FRI_FRM_INT[]
friMapBoolIn	$IN[ ]	$FRI_TO_BOOL[]
friMapBoolOut	$OUT[]	$FRI_FRM_BOOL[]
friMapDelay	Configuration of the update rate	

These mapping commands are called by means of 16 INTEGER values. The contents of these commands are the same as described in static configuration.

(>>> 6.1.1 "Static field bus configuration" Page 19)

7 Start-up and recommissioning

7.1 Softkeys and status keys

Overview

The following status keys are available:

Status key	Description
FRI	Activates the FRI softkey bar.

The following softkeys are available:

Softkey	Description
FRIOPEN	The connection is opened.
FRISTART	Data transfer is activated.
FRISTOP	Data transfer is deactivated.
FRICLOSE	The connection is closed.
FRISHOW	Displays the current status of the FRI.
Cancel	The standard softkey bar is displayed.

7.2 KRL commands

Overview

The FRI is controlled by means of the following commands:

FRIOPEN 1	(>>> 7.2.1 "KRL command FRIOPEN" Page 21)
FRIOPEN 2	(>>> 7.2.2 "KRL command FRIOPEN2" Page 22)
FRISTART	(>>> 7.2.3 "KRL command FRISTART" Page 23)
FRISTOP	(>>> 7.2.4 "KRL command FRISTOP" Page 25)
FRICLOSE	(>>> 7.2.5 "KRL command FRICLOSE" Page 25)
FRISHOW	(>>> 7.2.6 "KRL command FRISHOW" Page 26)
FRISETUP	(>>> 7.2.7 "KRL command FRISETUP" Page 27)

7.2.1 KRL command FRIOPEN

Syntax

```
FRISTATE FRIOPEN( INT sendTimeInMsec)
```

Explanation of the syntax

Parameter	Description	Value
sendTimeInMsec	Transmission time A data packet is sent from the robot controller to the external computer within the time specified.	1 … 100 ms

Returns

Return	Description
INVALID	Error, e.g. No license available
MON	Monitor mode is activated

Description

The sending and receiving of data packets is symmetrical. This command acts like FRIOPEN2 (sendTimeInMsec,1).

(>>> 7.2.2 "KRL command FRIOPEN2" Page 22)

The following functions are executed:

- FRI connection is activated

- UDP socket to external computer is opened
- Monitor mode is activated
- Cycle time is set with this command (symmetrical time behavior)
  Remote station responds in the same timeframe.
- Data exchange between robot controller and external computer is carried out

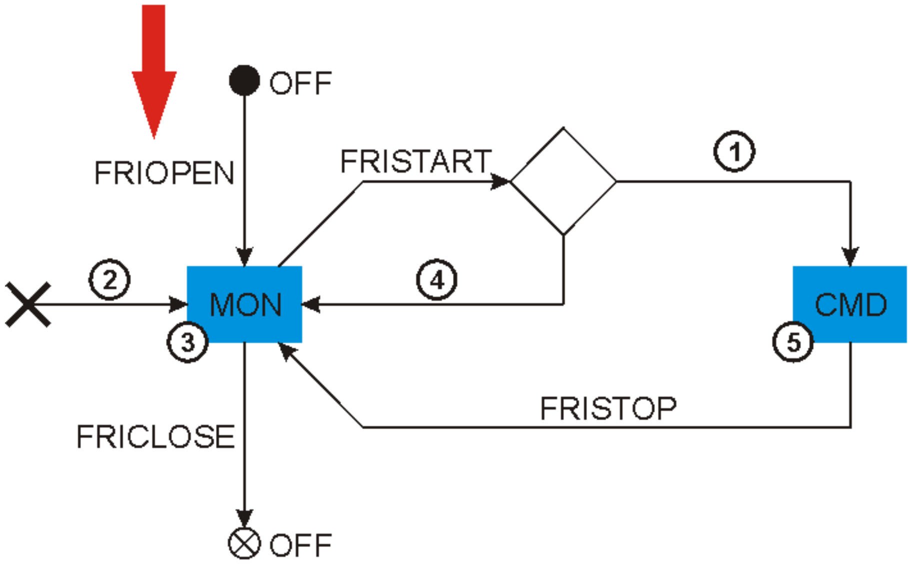


Fig. 7-1: FRIOPEN

1	Data packet set in the specified timeframe
2	Error
3	Monitor mode
4	Quality of the data transfer is not sufficient
5	Command mode

7.2.2 KRL command FRIOPEN2

Syntax

```
FRISTATE friOpen2( Int sendTimeInMsec, Int receiveMultiple)
```

Explanation of the syntax

Parameter	Description	Value
sendTimeInMsec	Transmission time A data packet is sent from the robot controller to the external computer within the time specified.	1 … 100 ms
receiveMultiple	Multiplier The multiplier is multiplied by the transmission time. The resulting value specifies the interval at which a check-back signal is sent from the external computer to the robot controller.	1 … x Recommended value: 1

The product calculated in "sendTimeInMsec * receiveMultiple" must not exceed 100 ms.

Returns

Return	Description
INVALID	Error, e.g. No license available
MON	Monitor mode is activated

Description

The following functions are executed:

- FRI connection is activated
- UDP socket to external computer is opened
- Monitor mode is activated
- A cycle time of 1 … 100 ms can be set (asymmetrical time behavior)
  The remote station responds more slowly.
- Data exchange between robot controller and external computer is carried out

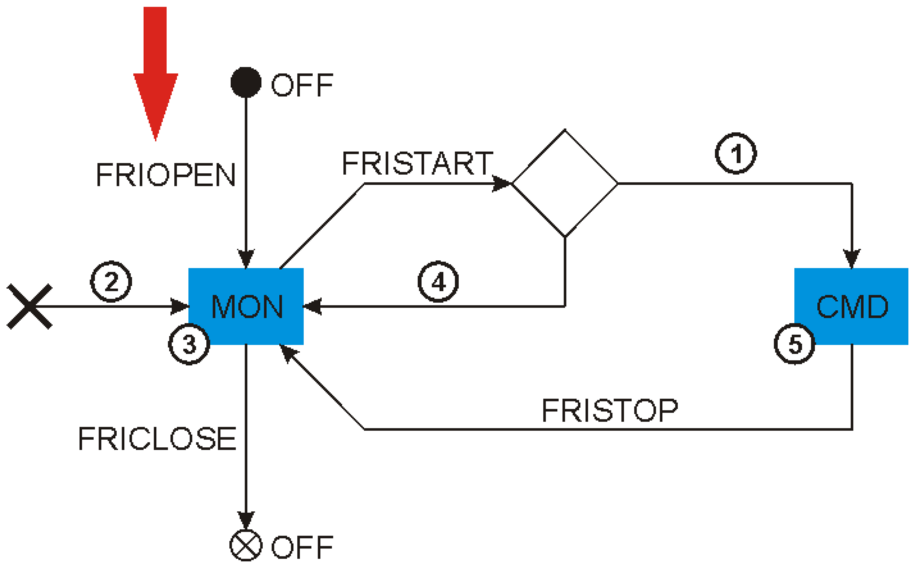


Fig. 7-2: FRIOPEN

1 Data packet set in the specified timeframe
2 Error
3 Monitor mode
4 Quality of the data transfer is not sufficient
5 Command mode

Example

Transmission time = 3 ms

Multiplier = 3

Calculation: 3*3=9

A data packet is sent from the robot controller to the external computer every 3 ms. A checkback signal is sent from the external computer to the robot controller every 9 ms.

7.2.3 KRL command FRISTART

Syntax

```
FRISTATE friStart( real safetyVal)
```

Explanation of the syntax

Parameter	Description	Value
safetyVal	The SafetyVal factor restricts the permissible limit ranges (e.g. velocity, acceleration).	0.0 ... 1.0 Default: 1.0

Returns

Return	Description
INVALID	Error, e.g. No license available
MON	Monitor mode is activated
CMD	Command mode is activated

Description

The SafetyVal factor is used in experiments in which a new algorithm is introduced on the external computer. This allows the program created by the customer to be executed slowly and tested.

The value of SafetyVal must be between 0.0 and 1.0. The default value is 1.0, i.e. the limit ranges predefined by KUKA apply.

Command mode is activated if:

- FRI connection is active.
- Quality of the data transfer is “Good” or “Perfect”.
- Load data are entered correctly, i.e. the message “Verify load data” is not displayed.

This instruction causes the following to be executed:

- Monitor mode is deactivated
- Command mode is activated

  Commands from the external computer are implemented
- The following controllers can be used:
  - Position controller
  - Cartesian stiffness controller
  - Axis-specific stiffness controller
- Set limit values (e.g. velocity, acceleration, etc.) with the SafetyVal factor

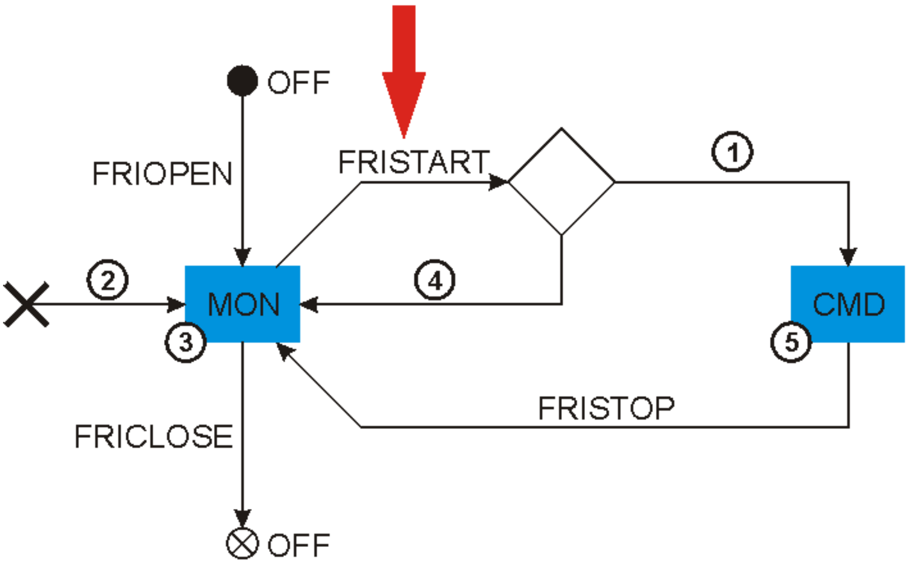


Fig. 7-3: FRISTART

1 Data packet set in the specified timeframe
2 Error
3 Monitor mode
4 Quality of the data transfer is not sufficient
5 Command mode

7.2.4 KRL command FRISTOP

Syntax

```
FRISTATE FRISTOP()
```

Returns

Return	Description
INVALID	Error, e.g. No license available
MON	Monitor mode is activated until the conditions change

Description

This instruction causes the following to be executed:

- Command mode is deactivated
- Monitor mode is activated

  Any robot motions are stopped.
- Multiple Start-Stop (FRISTART, FRISTOP) loops can be executed.

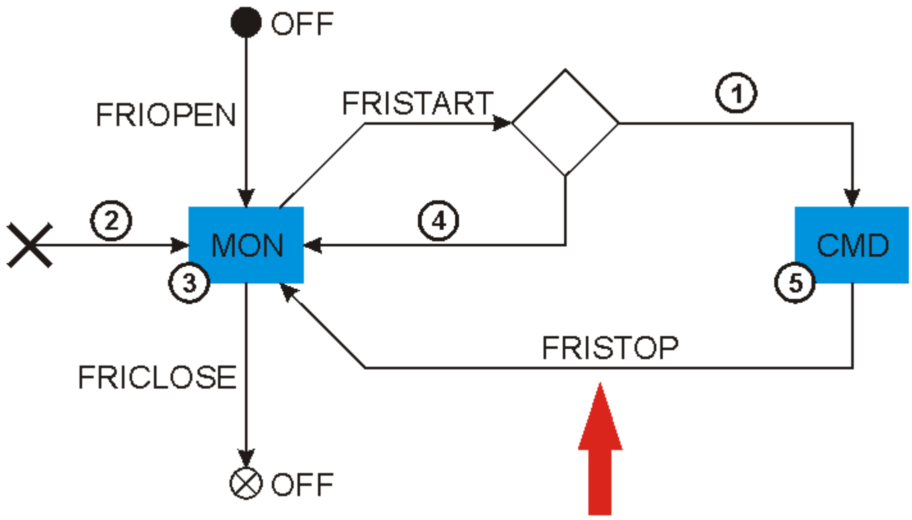


Fig. 7-4: FRISTOP

1 Data packet set in the specified timeframe
2 Error
3 Monitor mode
4 Quality of the data transfer is not sufficient
5 Command mode

7.2.5 KRL command FRICLOSE

Syntax

```
FRISTATE FRICLOSE()
```

Returns

Return	Description
INVALID	Error, e.g. No license available
OFF	FRI connection is deactivated

Description

This instruction causes the following to be executed:

- UDP socket and associated tasks are closed
- Open sequences are closed
- FRI connection is deactivated

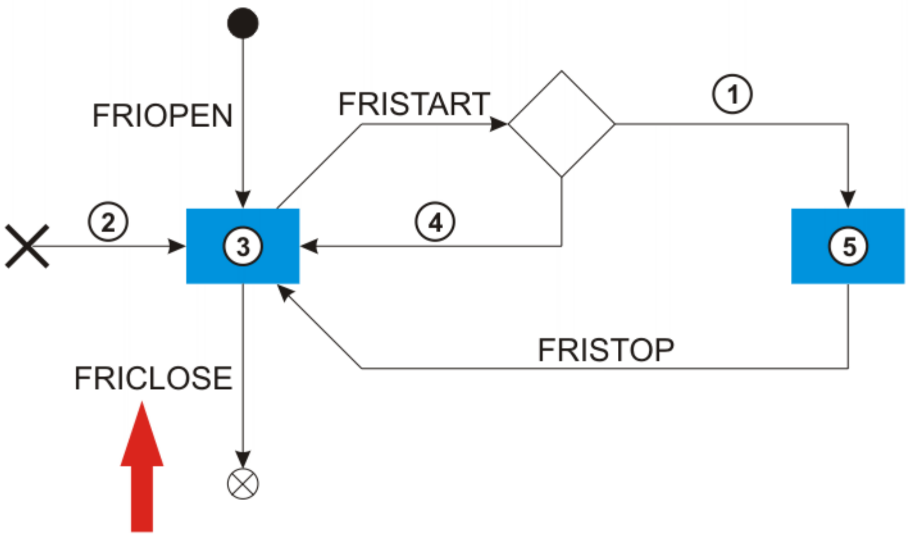


Fig. 7-5: FRICLOSE

1 Data packet set in the specified timeframe
2 Error
3 Monitor mode
4 Quality of the data transfer is not sufficient
5 Command mode

7.2.6 KRL command FRISHOW

Syntax

```
FRISTATE FRISHOW()
```

Returns

Return	Description
INVALID	Error, e.g. No license available
OFF	FRI connection is deactivated
MON	Monitor mode is activated
CMD	Command mode is activated

Description

This command can be called as follows:

- Press the softkey on the user interface of the robot controller.
  or
- Enter the following string in the variable display (Monitor > Variable > Single):

```
=friShow
```

Pressing the softkey FRISHOW displays the following information about the status of the FRI:

- IP host
- PortHost
- PortRecieve
- Communication quality
- Cycle time
- Multiplier
- Latency time
- Transfer mode

If there is no valid license available, the message “NO FRI License” appears.

7.2.7 KRL command FRISETUP

Syntax

```
FRISTATE FRISETUP( CHAR HOSTNAME[99], int friToPort, int friFrmPort)
```

Explanation of the syntax

Parameter	Description	Value
Char Hostname	IP address	Must correspond to at least IPv4.
Int friToPort	Socket port number of the external computer	-
Int friFrmPort	KRC port number of the robot controller	Default: Zero

Returns

Return	Description
INVALID	Error, e.g. No license available
OFF	FRI connection is deactivated
MON or CMD	If the FRI is in Monitor/Command mode, the command has no effect until the next FRIOPEN command.

Description

This command is used to modify the predefined host and port addresses. The FRI must be closed in order to allow the changes to be made during the runtime.

If the preset address has been specified in the configuration file DLR-RC.INI, it is not necessary to change the address.

Example

```
DECL FRISTATE RetVal
retVal=friSetup("192.168.0.20", 49938,0)
```

The IP address 192.168.0.20 is specified with port 49938. The IP address must correspond to IPv4.

If only the port is to be modified and the IP address is to remain the same, “no” can be entered for the IP address.

```
retVal=friSetup("no", 49940,0)
```

7.3 Creating a program

Description

The following must be taken into consideration when creating your own programs:

- Integrate the following programs into the application:
  - friudp.H
  - friudp.CPP
  - friComm.H
- Disconnect critical real-time operations from UDP socket interactions

Recommendation

The programs friRemote.H and friRemote.CPP are not necessary for independent program creation, but simplify the interaction of the data structures.

8 System variables

8.1 System variables

These variables can be monitored in the KRL program or viewed via the variable display.

Further information is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

8.2 FRI variables

Specific FRI variables are defined on the robot controller. These can be monitored in the KRL program or viewed via the variable display.

Variable	KRL access	Description	C structure
$FRI_TO_INT[16]	Read/write	Type: INT[16] Send buffer: 16 INTEGER values to the external computer.	msr.krl.intData
$FRI_FRM_INT[16]	Read only	Type: INT[16] Receive buffer: 16 INTEGER values from the external computer.	cmd.krl.intData
$FRI_TO_REA[16]	Read/write	Type: REAL[16] Send buffer: 16 REAL values to the external computer.	msr.krl.realData
$FRI_FRM_REA[16]	Read only	Type: REAL[16] Receive buffer: 16 REAL values from the external computer.	cmd.krl.realData
$FRI_TO_BOOL[16]	Read/write	Type: BOOL[16] Send buffer: 16 BOOLEAN values to the external computer. **Note:** The bits are grouped together as integers.	msr.krl.boolData
$FRI_FRM_BOOL[16]	Read only	Type: BOOL[16] Receive buffer: 16 BOOLEAN values from the external computer.	cmd.krl.boolData

Variable	KRL access	Description	C structure
$FRIQUALITY		Type: ENUM FRIQUALITY Variable indicates the quality of the UDP connection. ■ PERFECT ■ OK ■ BAD ■ UNACCEPTABLE ■ INVALID (>>> 3.3.1 "UDP timing quality" Page 11)	
$FRISTATE		Type: ENUM FRISTATE Variable indicates the FRI state. ■ OFF ■ MON ■ CMD ■ INVALID (>>> 3.2 "FRI modes" Page 9)	

The indexing of the arrays begins with 1 in KRL and 0 on the external computer. $FRI_TO_INT[1] thus corresponds to msr.krl.intData[0], for example.

9 Programming

9.1 Data exchange by means of data telegrams

Description

In the case of data exchange between the robot controller and the external computer, the following data telegrams are transferred:

- Msr data telegrams (tFriMsr.msr)
  (>>> 9.1.1 "Msr data telegram" Page 31)
- Cmd data telegrams (tFriCMD.cmd)
  (>>> 9.1.2 "Cmd data telegram" Page 35)

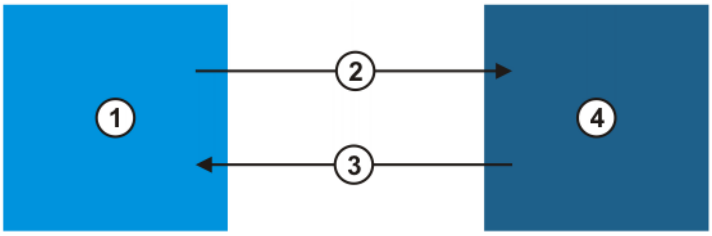


Fig. 9-1: Data exchange by means of data telegrams

1 Robot controller
2 Msr data telegram
3 Cmd data telegram
4 External computer

Data telegram	Description
Msr data telegram	Sent from the robot controller to the external computer.
Cmd data telegram	Sent from the external computer to the robot controller.

9.1.1 Msr data telegram

Description

This data telegram contains the measured values sent from the robot controller to the external computer.

Component	KRL variable	Description	Data structure
Handshake (header) `tFriHeader`			`msr.head`
		Sent sequence count	`msr.head.sendSeqCount`
		Reflected sequence count	`msr.head.reflSeqCount`
		Size of the data telegram in bytes `sizeof(tFriMsrData)`	`msr.head.paketSize`
		ID of data telegram `FRI_DATAGRAM_ID_MSR`	`msr.head.datagramId`

Component	KRL variable	Description	Data structure
FRI state `tFriIntfState`		Statistics, configurations	`msr.intf`
		Time stamp in seconds	`msr.intf.timestamp`
		State of the FRI (Monitor mode, Command mode) `FRI_STATE`	`msr.intf.state`
		Quality of the UDP communication (and latency time)	`msr.intf.quality`
		Msr cycle time requested by means of **friOpen**	`msr.intf. desiredMsrSampleTime`
		Cmd cycle time requested by means of **friOpen**	`msr.intf. desiredCmdSampleTime`
		Restrictions of the limit ranges requested by means of **friStart** (SafetyVal)	`msr.intf. safetyLimits`
FRI statistics based on sequence count `tFriIntfStatistics`			`msr.intf.stat`
		Average answer rate	`msr.intf.stat.answerRate`
		Latency time in seconds	`msr.intf.stat.latency`
		Jitter in seconds	`msr.intf.stat.jitter`
		Average number of data packets lost (due to UDP and real-time errors)	`msr.intf.stat.missRate`
		Absolute number of data packets lost after the command **friOpen**	`msr.intf.stat.missCounter`
Robot state `tFriRobotState`		Information about the robot	`msr.robot`
		Drives on or off	`msr.robot.power`
	$STIFF_STRAT_ACT[] $STIFFNESS.STRATEGY[]	Active controller strategy	`msr.robot.control`
		An error is displayed in the message window.	`msr.robot.error`
		A warning is displayed in the message window.	`msr.robot.warning`
		Operating temperature	`msr.robot.temperature`

Component	KRL variable	Description	Data structure
Robot data `tFriRobotData`			`msr.data`
	$AXIS_ACT_MES[]	Measured axis position	`msr.data.msrJntPos`
	$AXIS_ACT_CMD[]	Programmed axis position* *The setpoint position currently used is the sum of the following elements: ▪ Programmed axis position `msr.data.cmdJntPos` ▪ Programmed offset of FRI `msr.data.cmdJntPosFriOffset`	`msr.data.cmdJntPos`
		Programmed offset of FRI* *The setpoint position currently used is the sum of the following elements: ▪ Programmed axis position `msr.data.cmdJntPos` ▪ Programmed offset of FRI `msr.data.cmdJntPosFriOffset`	`msr.data.cmdJntPosFriOffset`

Component	KRL variable	Description	Data structure
Robot data `tFriRobotData`	$POS_ACT_MSR[]	Measured Cartesian position of the TCP Defined by the following variables: ■ $STIFFNESS.BASE ■ $STIFFNESS.TOOL **Note:** ■ $TOOL and $BASE are required for $POS_ACT_MSR[]. ■ $STIFFNESS.BASE and $STIFFNESS.TOOL are required for `msr.data.msrCartPos`.	`msr.data.msrCartPos`
	$POS_ACT_CMD[]	Programmed Cartesian position of the TCP **Note:** ■ $TOOL and $BASE are required for $POS_ACT_MSR. ■ $STIFFNESS.BASE and $STIFFNESS.TOOL are required for msr.data.msrCartPos.	`msr.data.cmdCartPos`
		Programmed offset of FRI	`msr.data.cmdCartPosFriOffset`

Component	KRL variable	Description	Data structure
Robot data tFriRobotData	$TORQUE_AXIS_ACT[]	Axis-specific torque	msr.data.msrJntTrq
	$TORQUE_AXIS_EST[]	Current external axis-specific torque	msr.data.estExtJntTrq
	$TORQUE_TCP_EST[]	Force measurement of the torque sensors	msr.data.estExtTcpFT
		Numeric Jacobian matrix	msr.data.jacobian
		Numeric mass matrix	msr.data.massMatrix
		Numeric gravitation compensation	msr.data.gravity
fFriKrlData		KRL interaction	msr.krl
	$FRI_TO_REA[]	16 REAL values to the external FRI computer	msr.krl.realData
	$FRI_TO_INT[]	16 INTEGER values	msr.krl.intData
	$FRI_TO_BOOL[]	16 BOOLEAN values	msr.krl.boolData

9.1.2 Cmd data telegram

Description

This data telegram contains the header information (>>> 3.3 "UDP communication" Page 11).

Additional commanded information is coded in the KRL and command sections.

Component	KRL variable	Description	Data structure
Handshake (header) tFriHeader			Cmd.head
		Sent sequence count	Cmd.head.sendSeqCount
		Reflected sequence count	Cmd.head.reflSeqCount
		Size of the data telegram in bytes sizeof(tFriCmdData)	Cmd.head.packetSize
		ID of data telegram FRI_DATAGRAM_ID_CMD	Cmd.head.datagramId

KRL interaction variables:

Component	KRL variable	Description	Data structure
tFriKrlData		KRL interaction	Cmd.krl
	$FRI_FRM_REA[]	16 REAL values to the external FRI computer	Cmd.krl.realData
	$FRI_FRM_INT[]	16 INTEGER values	Cmd.krl.intData
	$FRI_FRM_BOOL[]	16 BOOLEAN values	Cmd.krl.boolData

The command section: every data structure is specified with the array `Cmd.data.cmdFlags`. The command settings contained in this array are specified by the external computer. It is possible to specify only selected settings, e.g. only the stiffness values in the stiffness controller. For all non-specified settings, the default values of the KRL program are accepted (system variable $STIFFNESS).

> Further information is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

Parameter	Descrip-tion	Data structure	Controller strategies		
			Position controller	Cartesian stiffness	Axis-specific stiffness
Command structure `tFriRobotCommand`		`Cmd.data`	10	20	30
	Specified settings (>>> 9.2 "Controller strategy" Page 37)	`Cmd.data.cmdFlags`			
`FRI_CMD_JNTPOS` q_{FRI}	Axis-specific setpoint position	`Cmd.data.jntPos`	x	y(1)	x
`FRI_CMD_CARTPOS` x_{FRI}	Cartesian setpoint position	`Cmd.data.cartPos`		x(1)	
`FRI_CMD_JNTTRQ` τ_{FRI}	Axis-specific torque	`Cmd.data.addJntTrq`			x
`FRI_CMD_TCPFT` FT_{FRI}	Application of an external force	`Cmd.data.addTcpFT`		x	
`FRI_CMD_JNTSTIFF` k_j	Axis-specific setpoint stiffness	`Cmd.data.jntStiffness`		y	x
`FRI_CMD_JNTDAMP` d_j	Axis-specific setpoint damping	`Cmd.data.jntDamping`		y	x
`FRI_CMD_CARTSTIFF` k_c	Cartesian setpoint stiffness	`Cmd.data.cartStiffness`		x	
`FRI_CMD_CARTDAMP` d_c	Cartesian setpoint damping	`Cmd.data.cartDamping`		x	

- x: Available in these controller strategies.
- y: Available in these controller strategies. Further information is required, however.

(1) In Cartesian stiffness, the Cartesian setpoint position (cartPos) can be specified with an additional zero space (jntPos). Alternatively, the setpoint position can be specified by means of the axis position (jntPos) if no Cartesian setpoint position (cartPos) has been specified. In this case, the Cartesian setpoint position is generated by means of the forward kinematic.

> The settings in `Cmd.data.cmdFlags` can only be modified in Monitor mode.

Example 1

Procedure for modification of the settings in `Cmd.data.cmdFlags`:

1. Select FRISTOP.
2. Modify settings in `Cmd.data.cmdFlags`.
3. Select FRISTART.

Example 2

Procedure for modification of the settings in $STIFFNESS and `Cmd.data.cmdFlags`:

1. Select FRISTOP.
2. Modify configuration of $STIFFNESS.
3. Modify settings in `Cmd.data.cmdFlags`.
4. Select FRISTART.

For applications that require fast modification of the interactive behavior, the stiffness controller can be used with the variable $STIFFNESS (Cartesian and axis-specific stiffness controller).

9.2 Controller strategy

Overview

The following controller strategies can be used:

- Controller strategy 10 (position controller)

  (>>> 9.2.1 " Controller strategy 10 (position controller)" Page 37)
- Controller strategy 20 (Cartesian stiffness controller)

  (>>> 9.2.2 "Controller strategy 20 (Cartesian stiffness controller)" Page 38)
- Controller strategy 30 (axis-specific stiffness controller)

  (>>> 9.2.3 "Controller strategy 30 (axis-specific stiffness controller)" Page 39)

> Further information is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

9.2.1 Controller strategy 10 (position controller)

Description

The control law of the position controller is:

$$q_{LBR} = q_{FRI}$$

Only the array `cmd.data.jntPos` is valid and applicable.

The axis-specific setpoint position is commanded during operation in Command mode.

In order to be able to switch to Command mode, the external FRI side must return `msr.data.cmdJntPos` and `msr.data.cmdJntPosFriOffset`.

In the case of a cycle time of 1 ms, the following applies:

$$q_{LBR} = q_{FRI}$$

Parameter	Description
q_{FRI}	Absolute axis-specific position

If the cycle time is greater than 1 ms, fine interpolation is carried out as follows:

$$\Delta q_{i,\,n\,ms} = \frac{(q_{FRI} - q_{FRI,\,i\text{-}i})}{n\ ms}$$

$$q_{LBR}\,(j) = q_{FRI,\,i\text{-}1} + j * \Delta q_{i,\,n\,ms}$$

$$j = [1...n]$$

9.2.2 Controller strategy 20 (Cartesian stiffness controller)

Description

The control law of the Cartesian stiffness controller is:

$$\tau_{Cmd} = J^T\,(k_c(x_{FRI} - x_{msr}) + F_{FRI}) + D(d_c) + f_{dynamics}\,(q,\dot{q},\ddot{q})$$

This calculation rule is defined by the transposed Jacobian matrix and consists of the following parameters. Most parameters can be modified by the customer using the FRI.

- Cartesian setpoint position:

  x_{FRI}

- Cartesian stiffness (represents a virtual spring):

  k_c

  $k_c(x_{FRI} - x_{msr})$

> **⚠ CAUTION**
> The robot can execute unexpected motions if:
> - The dynamic model is incorrect and the stiffness values are low.
>
>   or
> - The status message 480 “Verify load data” is displayed.
>
> Customer programs must be tested in T1 mode. Severe physical injuries or considerable damage to property may otherwise result.

- Damping term, dependent on the standardized damping value:

  $D(d_c)$

- Dynamic model (cannot be modified by the external computer):

  $f_{dynamics}\,(q,\dot{q},\ddot{q})$

- Cartesian force:

  F_{FRI}

- The zero space configuration is defined by the axis-specific setpoint position (`FRI_CMD_JNTPOS`) which must correspond to the current situation. An unexpected control response may otherwise occur.
- Alternatively, the Cartesian setpoint position can be specified by the axis position. In this case, `FRI_CMD_CARTPOS` must be omitted.

9.2.3 Controller strategy 30 (axis-specific stiffness controller)

Description

The control law of the axis-specific stiffness controller is:

$$\tau_{Cmd} = k_j(q_{FRI} - q_{msr}) + D(d_j) + \tau_{FRI} + f_{dynamics}(q,\dot{q},\ddot{q})$$

The axis-specific torque consists of the following parameters. Most parameters can be modified by the customer using the FRI.

- Stiffness:

  k_j
- Virtual spring in the jointed arm:

  $k_j(q_{FRI} - q_{msr})$

> **⚠ CAUTION** The robot can execute unexpected motions if:
> - The dynamic model is incorrect and the stiffness values are low.
>
> or
> - The status message 480 “Verify load data” is displayed.
>
> Customer programs must be tested in T1 mode. Severe physical injuries or considerable damage to property may otherwise result.

- Damping term, dependent on the standardized damping value:

  $D(d_j)$
- Dynamic model:

  $f_{dynamics}(q,\dot{q},\ddot{q})$
- Axis-specific torque:

  τ_{FRI}

9.3 Programmed axis position (cmd.data.JntPos)

Description

The axis positions can be programmed in all controller strategies, particularly in controller strategies 10 (position controller) and 30 (axis-specific stiffness controller).

The case: FRI_CMD_JNTPOS:

- The axis-specific setpoint position is commanded during operation in Command mode.
- In order to be able to switch to Command mode, the external FRI side must reflect `msr.data.cmdJntPos` and `msr.data.cmdJntPosFriOffset`.

In the case of a cycle time of 1 ms, the following applies:

$$q_{LBR} = q_{FRI}$$

Parameter	Description
q_{FRI}	Absolute axis-specific position

If the cycle time is greater than 1 ms, fine interpolation is carried out as follows:

$$\Delta q_{i,\, n\, ms} = \frac{(q_{FRI} - q_{FRI,\, i\text{-}i})}{n\ ms}$$

$$q_{LBR}\,(j) = q_{FRI,\, i\text{-}1} + j * \Delta q_{i,\, n\, ms}$$

$$j = [1...n]$$

9.4 Axis-specific torque (cmd.data.addJntTrq)

Description

The axis-specific torque is available in controller strategy 30 (axis-specific stiffness controller). In order to be able to execute Command mode, zero must be commanded in Monitor mode.

$\Delta\tau_{FRI}$	FRI_CMD_JNTTRQ	Axis-specific torque	cmd.data.addJnt-Trq	Controller strategy 30

This functional principle is analogous to that of DESIREDFORCE. Further information about DESIREDFORCE is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

The additional axis-specific torque is superposed directly over the axis-specific stiffness controller. For further information see:

(>>> 9.2.2 "Controller strategy 20 (Cartesian stiffness controller)" Page 38)

(>>> 9.2.3 "Controller strategy 30 (axis-specific stiffness controller)" Page 39)

In order to be able to make use of this property, the field $STIFFNESS.AXISMAXDELTA must be defined with a positive value.

9.5 Axis-specific setpoint stiffness

Description

Parameter	Description	Data structure	Controller strategies		
			Position controller	Cartesian stiffness	Axis-specific stiffness
FRI_CMD_JNTSTIFF k_j	Axis-specific setpoint stiffness	Cmd.data.jnt-Stiffness		y	x
FRI_CMD_JNTDAMP d_j	Axis-specific setpoint damping	Cmd.data.jnt-Damping		y	x

If the axis-specific stiffness and damping parameters are not configured, the default settings of the system variables $STIFFNESS.AXISSTIFFNESS and $STIFFNESS.AXISDAMPING are applied.

> **i** Command mode cannot be activated until the virtual spring has no deviation from the setpoint position.

9.6 Cartesian setpoint position (cmd.data.cartPos)

Description

In order for Cartesian setpoint position to be available in controller strategy 20 (Cartesian stiffness controller), the following 2 arrays are required.

- `cmd.data.cmdCartPos[12]`
- `msr.data.cmdCartPos[12]`

The 12 values are interpreted as a 3x4 matrix. This layout is used as most compilers are able to transform 12 vectors into a 3x4 matrix.

$$X_{3x4} = \begin{bmatrix} R_{3x3} & P_3 \end{bmatrix}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 \\ r_{21} & r_{22} & r_{23} & p_2 \\ r_{31} & r_{32} & r_{33} & p_3 \end{bmatrix}$$

$$X_{12} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 & r_{21} & r_{22} & r_{23} & p_2 & r_{31} & r_{32} & r_{33} & p_3 \end{bmatrix}$$

The orientation presentation of matrix R is orthonormal.

9.7 Superposed exertion of force

Description

The superposed exertion of force is available in controller strategy 20 (Cartesian stiffness controller). In order to be able to switch from Monitor mode to Command mode, zero must be entered during operation in Monitor mode.

Parameter	Descrip-tion	Data struc-ture	Controller strategies		
			Position controller	Cartesian stiffness	Axis-specific stiffness
`FRI_CMD_TCPFT` FT_{FRI}	Application of an external force	`Cmd.data.addTcpFT`		x	

Representation (Cartesian vector):

$$FT_{FRI} = \begin{bmatrix} F_x & F_y & F_z & T_z & T_y & T_x \end{bmatrix}$$

Symbol in formula	Description
F	Force along the coordinate axis
T	Torque along the coordinate axis

> **i** This function is analogous to the DESIREDFORCE function. Further information is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

The additional axis-specific torque is superposed directly over the axis-specific stiffness controller. For further information see:

(>>> 9.2.2 "Controller strategy 20 (Cartesian stiffness controller)" Page 38)

(>>> 9.2.3 "Controller strategy 30 (axis-specific stiffness controller)" Page 39)

In order to be able to use this function, the array $STIFFNESS.CP-MAXDELTA must be defined with a positive value.

⚠ CAUTION With this function, it must be taken into consideration that the defined forces are exerted immediately and the robot thus executes its motions very quickly.

⚠ CAUTION When this function is used for the first time, the defined force must be small and tested in T1 mode.

9.8 Cartesian stiffness parameters

Description

Parameter	Description	Data structure	Controller strategies		
			Position controller	Cartesian stiffness	Axis-specific stiffness
`FRI_CMD_CARTSTIFF` k_c	Cartesian setpoint stiffness	`Cmd.data.cartStiffness`		x	
`FRI_CMD_CARTDAMP` d_c	Cartesian setpoint damping	`Cmd.data.cartDamping`		x	

$$k_c = [k_x \;\; k_y \;\; k_z \;\; k_{A(Rz)} \;\; k_{B(Ry)} \;\; k_{C(Rx)}]$$

$$d_c = [d_x \;\; d_y \;\; d_z \;\; d_{A(Rz)} \;\; d_{B(Ry)} \;\; d_{C(Rx)}]$$

If the stiffness and damping parameters are not configured, the default settings of the system variables $STIFFNESS are applied.

Command mode cannot be activated until the virtual spring has no deviation from the setpoint position.

9.9 FRI_GuiApp

Description

FRI_GuiApp serves as a graphical user interface for the FRI interface. A prerequisite for using it is the Windows operating system.

FRI_GuiApp is generally stable with a data transfer rate of 20 ms. Higher data transfer rates are possible, but dependent on the installation (e.g. firewall, virus scanner, network configurations).

FRI_GuiApp is provided by KUKA Roboter, but correct operation is not guaranteed.

10 Examples

10.1 Introductory program 1

Description

This introductory program is intended to illustrate the functional principle of the FRI. It demonstrates the communication between the robot controller and an external computer so that the user can subsequently create a program independently.

In this introductory program, the robot first moves between the HOME position and a taught position. Thereafter, the robot performs a loop between two positions.

This introductory program illustrates the following functions:

- Monitor and Command modes
- Interaction between KRL and FRI using the following system variables:
  - $FRI_FRM_INT[]
  - $FRI_TO_INT[]
  - $FRIFRM_REA[]
  - $FRI_TO_REA[]
- The following commands on the robot controller
  - FRISTART
  - FRIOPEN
  - FRISTOP
  - FRICLOSE

Preconditions

- Operating mode T1
- friSample1.SRC is collision-free

NOTICE The positions must be taught in such a way that the program can be executed without collisions and the robot can move without collisions. Damage to property may otherwise result.

Procedure

1. Compile the following files to an executable program on the external computer:

   From the directory InstallOnCD\FRI_RemoteSample\FRI_FirstApp

   - friFirstApp.CPP

   From the directory InstallOnCD\FRI_Remote

   - friremote.CPP

     A library for simplifying data transfer.
   - friudp.CPP

     Contains the direct UDP socket processing.
2. Start the program friFirstApp.CPP on the external computer.
3. Select the program friSample1.SRC on the robot controller.
4. Press and hold down the enabling switch and Start key.

   The robot executes the first loop as a dry run in Monitor mode. No external commands are activated. The FRI is opened.

   In the second loop, consisting of 3 cycles, the FRI is started. The interaction is displayed as a superposed sine wave.

   If the Start key is released, the superposed motion is resumed from the external computer.
5. Move the robot as far as the next stop.

   To do so, press the Start key again. The interface is closed.

The following messages appear:

On the external computer

- **fri...active**

On the robot controller

- **fri quality good**

> If these messages do not appear, there is a fault that must be eliminated.
> (>>> 11 "Troubleshooting" Page 45)

6. If the data transfer quality is classified as “Good”, a value is sent from the external computer to the robot controller using the system variable $FRI_FRM_INT[1]. This value indicates that Command mode is activated.
7. The previous path is executed again and superposed with commands from the external computer.
8. The FRI is stopped and the interface is closed. This is indicated on the robot controller.

10.2 Introductory program 2

Description

This introductory program illustrates the use of FRI in relation to stiffness controllers.

Preconditions

- Operating mode T1

Procedure

1. Select the application fri_SecondApp.CPP.
2. Compile the following files to an executable program:
   - FRI_SecondApp.CPP
   - friremote.CPP
   - friudp.CPP
3. Start the application fri_SecondApp.CPP on the external computer.
4. Select and start the program friSample2.SRC on the robot controller.

In this example, the control scheme (controller strategies) `friInst.getCurrentControlScheme()` is polled. The control scheme can only be modified on the robot controller in Monitor mode.

If it is necessary to modify a controller quickly for an application, the system variable $STIFFNESS must be used for this with controllers 20 (Cartesian stiffness controller) and 30 (axis-specific stiffness controller).

11 Troubleshooting

Fault	Possible cause	Remedy
No FRI status message is displayed in the message window.	FRI license is invalid.	Call **FRISHOW** and check the license.
	Invalid host/socket specification.	Call **FRISHOW** and check the host and socket settings. (>>> 7.2.7 "KRL command FRISETUP" Page 27)
	Ethernet connection is faulty.	Ping from external PC.
Command mode is not executed.	The UDP data transfer is classified as Unacceptable or Bad.	Check the quality of the data transfer. Increase cycle time, check real-time system.
	The dynamic model is faulty.	Adapt the system variables $GRAVITATION[ ] and $LOAD and check them with the interactive GravComp mode.
	License version FRI_light.	Call **FRISHOW** and check the license.
	Status message 480 “Verify load data” is displayed.	Eliminate the cause of the status message 480 “Verify load data”. Further information about the load data is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.

The message “Interpolation Error” appears.	The setpoint command from the external computer does not match the system status. In general, convergence criteria governing the transfer of rights have been violated. These must be observed particularly when switching from Monitor mode to Command mode, but also when activating the drives (T1/T2).	If the drives are switched off, the sum of the internal setpoint positions in the KRC and the external setpoint positions must generally be specified as a setpoint position by the external computer.
		
		`cmd.data.cmdJntPos`
		`=msr.data.cmdJntPos`
		`+msr.data.cmdJntPosOffset`

FRI switches from Command mode back to Monitor mode.	The UDP data transfer is classified as Unacceptable or Bad.	1. Check the quality of the data transfer on the KRC side with `FRISHOW`. 2. Check the statistics array of the msr structure.
	Errors occurred, e.g. violation of the permissible operating parameters.	Check the displayed error message.
In AUT and AUT EXT modes, the robot switches off and the brakes are applied.	The watchdog is activated. Variable $BRK_DEL_PRO in R1/$machine.DAT	Increase the interactive watchdog time. Increase the value of the variable $BRK_DEL_PRO in R1/$machine.DAT. Occasionally execute KRL instructions, such as PTP, LIN, CIRC (restart the watchdog).
System crash on calling **FRIOPEN**.	The Ethernet card is managed via Windows.	Check the connected hardware in the Device Manager.

Fault	Possible cause	Remedy
The size of the structures is wrong.	The compiler “Alignment” is not set to 4 bytes.	Check that the compiler “Alignment” has a size of 4 bytes (see operating instructions of the compiler). Use the following validation macros to check the structure size: ■ FRI_CHECK_SIZES_OK ■ FRI_PREPARE_CHECK_BYTE_ ORDER ■ FRI_CHECK_BYTE_ORDER
	Incorrect interface version.	Use the correct file for the robot controller: friComm.H
No connection is established between the robot controller and the external computer.	The IP address specification is incorrect.	Check IP addresses: 1. Call **FRISHOW**. 2. Check IP addresses and correct if necessary. 3. Open the file DIrrc.INI in the directory C:\KRC\ROBOTER\INIT.
	Program does not run on external computer.	Start program on external computer.
	External computer has active firewall.	Set firewall correctly on external computer.
	Ethernet cable is not connected to the Ethernet card.	Connect Ethernet cable to the correct PCI slot. (>>> 5.1.1 "PCI slot assignment" Page 15)
Status message 480 “Verify load data” is displayed.	■ The load data do not match the tool mounted on the flange. ■ The installation site of the robot is not suitable (inclination).	Eliminate the cause of the status message 480 “Verify load data”. ■ Check system variables $GRAVITATION[] and $ROBROOT. ■ Select correct tool. ■ Check system variable $LOAD for the current load data. ■ Configure tool load data (LOAD_DATA[].m and LOAD_DATA[].cm). Further information about the load data is contained in the operating and programming instructions of the KUKA System Software (KSS) for the lightweight robot.
The values of the FRI-specific variables ($FRI_TO_INT[], …, $FRI_RFM_REA[]) do not reach the external computer program.	The indexing of the KRL and external computer differ by 1. (>>> 8.2 "FRI variables" Page 29)	Adapt the numbering of the KRL program and the external computer (C++ program).

12 KUKA Service

12.1 Requesting support

Introduction

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

Information

The following information is required for processing a support request:

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

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

12.2 KUKA Customer Support

Availability

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

Argentina

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

Australia

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

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

France	KUKA Automatisme + Robotique SAS Techvallée 6, Avenue du Parc 91140 Villebon S/Yvette France Tel. +33 1 6931660-0 Fax +33 1 6931660-1 commercial@kuka.fr www.kuka.fr
India	KUKA Robotics India Pvt. Ltd. Office Number-7, German Centre, Level 12, Building No. - 9B DLF Cyber City Phase III 122 002 Gurgaon Haryana India Tel. +91 124 4635774 Fax +91 124 4635773 info@kuka.in www.kuka.in
Italy	KUKA Roboter Italia S.p.A. Via Pavia 9/a - int.6 10098 Rivoli (TO) Italy Tel. +39 011 959-5013 Fax +39 011 959-5141 kuka@kuka.it www.kuka.it
Japan	KUKA Robotics Japan K.K. Daiba Garden City Building 1F 2-3-5 Daiba, Minato-ku Tokyo 135-0091 Japan Tel. +81 3 6380-7311 Fax +81 3 6380-7312 info@kuka.co.jp
Korea	KUKA Robotics Korea Co. Ltd. RIT Center 306, Gyeonggi Technopark 1271-11 Sa 3-dong, Sangnok-gu Ansan City, Gyeonggi Do 426-901 Korea Tel. +82 31 501-1451 Fax +82 31 501-1461 info@kukakorea.com

Malaysia	KUKA Robot Automation Sdn Bhd South East Asia Regional Office No. 24, Jalan TPP 1/10 Taman Industri Puchong 47100 Puchong Selangor Malaysia Tel. +60 3 8061-0613 or -0614 Fax +60 3 8061-7386 info@kuka.com.my
Mexico	KUKA de Mexico S. de R.L. de C.V. Rio San Joaquin #339, Local 5 Colonia Pensil Sur C.P. 11490 Mexico D.F. Mexico Tel. +52 55 5203-8407 Fax +52 55 5203-8148 info@kuka.com.mx
Norway	KUKA Sveiseanlegg + Roboter Bryggeveien 9 2821 Gjövik Norway Tel. +47 61 133422 Fax +47 61 186200 geir.ulsrud@kuka.no
Austria	KUKA Roboter Austria GmbH Vertriebsbüro Österreich Regensburger Strasse 9/1 4020 Linz Austria Tel. +43 732 784752 Fax +43 732 793880 office@kuka-roboter.at www.kuka-roboter.at
Poland	KUKA Roboter Austria GmbH Spółka z ograniczoną odpowiedzialnością Oddział w Polsce Ul. Porcelanowa 10 40-246 Katowice Poland Tel. +48 327 30 32 13 or -14 Fax +48 327 30 32 26 ServicePL@kuka-roboter.de

Portugal	KUKA Sistemas de Automatización S.A. Rua do Alto da Guerra n° 50 Armazém 04 2910 011 Setúbal Portugal Tel. +351 265 729780 Fax +351 265 729782 kuka@mail.telepac.pt
Russia	OOO KUKA Robotics Rus Webnaja ul. 8A 107143 Moskau Russia Tel. +7 495 781-31-20 Fax +7 495 781-31-19 kuka-robotics.ru
Sweden	KUKA Svetsanläggningar + Robotar AB A. Odhners gata 15 421 30 Västra Frölunda Sweden Tel. +46 31 7266-200 Fax +46 31 7266-201 info@kuka.se
Switzerland	KUKA Roboter Schweiz AG Industriestr. 9 5432 Neuenhof Switzerland Tel. +41 44 74490-90 Fax +41 44 74490-91 info@kuka-roboter.ch www.kuka-roboter.ch
Spain	KUKA Robots IBÉRICA, S.A. Pol. Industrial Torrent de la Pastera Carrer del Bages s/n 08800 Vilanova i la Geltrú (Barcelona) Spain Tel. +34 93 8142-353 Fax +34 93 8142-950 Comercial@kuka-e.com www.kuka-e.com

South Africa	Jendamark Automation LTD (Agency) 76a York Road North End 6000 Port Elizabeth South Africa Tel. +27 41 391 4700 Fax +27 41 373 3869 www.jendamark.co.za
Taiwan	KUKA Robot Automation Taiwan Co., Ltd. No. 249 Pujong Road Jungli City, Taoyuan County 320 Taiwan, R. O. C. Tel. +886 3 4331988 Fax +886 3 4331948 info@kuka.com.tw www.kuka.com.tw
Thailand	KUKA Robot Automation (M)SdnBhd Thailand Office c/o Maccall System Co. Ltd. 49/9-10 Soi Kingkaew 30 Kingkaew Road Tt. Rachatheva, A. Bangpli Samutprakarn 10540 Thailand Tel. +66 2 7502737 Fax +66 2 6612355 atika@ji-net.com www.kuka-roboter.de
Czech Republic	KUKA Roboter Austria GmbH Organisation Tschechien und Slowakei Sezemická 2757/2 193 00 Praha Horní Počernice Czech Republic Tel. +420 22 62 12 27 2 Fax +420 22 62 12 27 0 support@kuka.cz
Hungary	KUKA Robotics Hungaria Kft. Fö út 140 2335 Taksony Hungary Tel. +36 24 501609 Fax +36 24 477031 info@kuka-robotics.hu

USA

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

UK

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

Index