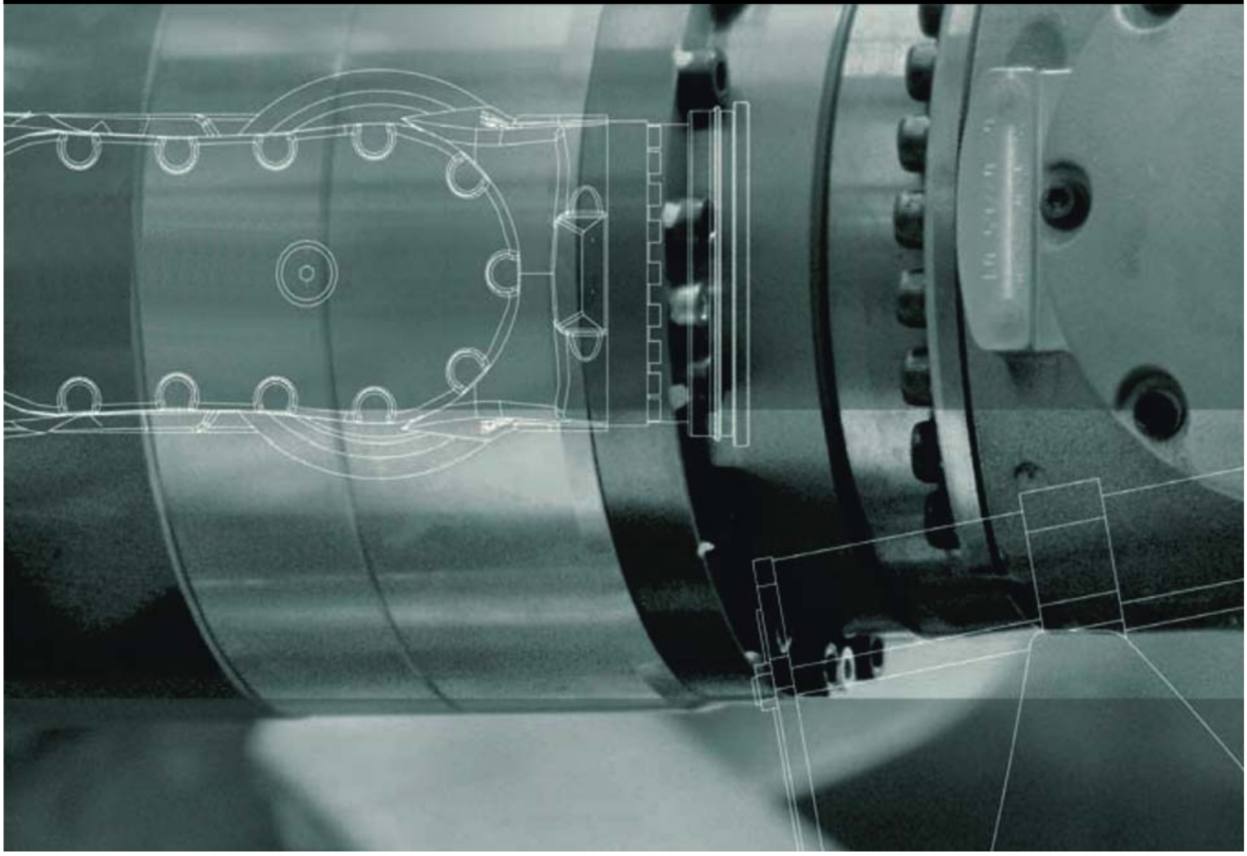
KUKA System Technology

KUKA Roboter GmbH

KUKA.PLC ProConOS 4-1 4.0

For KUKA System Software 8.2



Issued: 07.07.2011

Version: KST PLC ProConOS 4.0 V2 en

KUKA Roboter GmbH
Zugspitzstraße 140
D-86165 Augsburg
Germany

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

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

Subject to technical alterations without an effect on the function.

Translation of the original documentation

KIM-PS5-DOC

Publication: Pub KST PLC ProConOS 4.0 en
Bookstructure: KST PLC ProConOS 4.0 V2.1
Label: KST PLC ProConOS 4.0 V2 en

Contents

1 Introduction

1.1 Target group

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

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

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

1.2 Industrial robot documentation

The industrial robot documentation consists of the following parts:

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

Each of these sets of instructions is a separate document.

1.3 Representation of warnings and notes

Safety

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

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

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

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

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

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

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.

1.4 Trademarks

ProConOS and **MULTIPROG** are trademarks of KW-Software GmbH.

VxWorks is a trademark of Wind River Systems Inc.

Windows and **Windows XP** are trademarks of Microsoft Corporation.

1.5 Terms used

Term	Description
KLI	KUKA Line Interface. Connection to higher-level control infrastructure (PLC, archiving)
Exception	Exceptional treatment for a specific event.
KUKA.PLC Multiprog 5-35	Soft PLC for use in the robot controller.
WorkVisual	Software that serves as an offline engineering system for the software of KR C4-controlled robot systems.
SPOC	Single Point of Control
Retentive data	Data that are retained even when the PLC is reset.
Time slice	Allocated time slot in which an application takes over the CPU processing time.
Watchdog	Function that monitors the max. allocated processing/ response time.

2 Product description

2.1 Overview of KUKA.PLC ProConOS 4-1

Description

KUKA.PLC ProConOS 4-1 is a runtime system that is used by the technology package KUKA.PLC Multiprog 5-35 to execute PLC applications on the robot controller. ProConOS is compatible with the IEC standard 61131-3, and thus employs standardized syntax and semantics. ProConOS can be configured by means of an initialization file, and can be adapted flexibly to the requirements of the robot system.

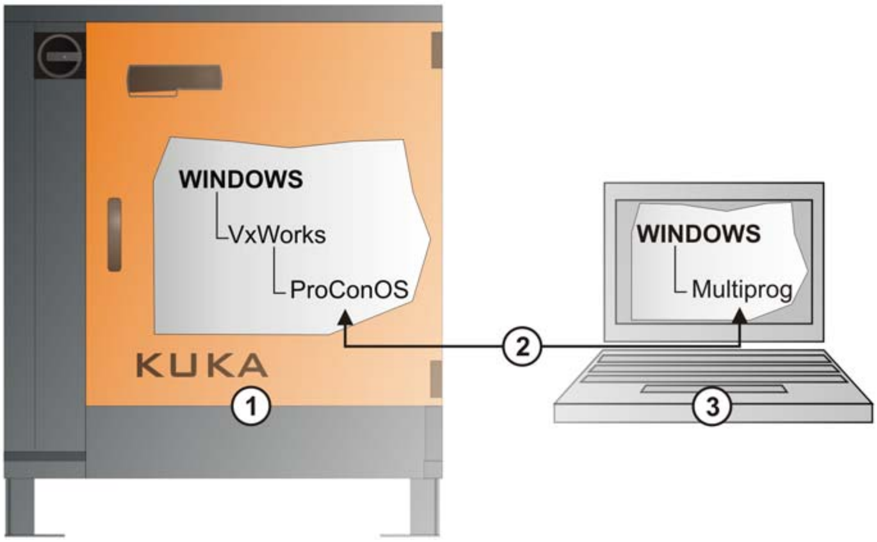


Fig. 2-1: Configuration overview

Item	Description
1	KR C4: ■ Windows operating system ■ VxWorks real-time operating system ■ ProConOS runtime system
2	Connection: KLI
3	External PC/laptop: ■ Windows operating system ■ MULTIPROG PLC development environment ■ WorkVisual offline engineering system

3 Safety

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

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

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

> ⚠ **WARNING** Signal states can be changed by downloading the Multiprog project, via the control dialog in Multiprog or by transferring them out of WorkVisual. It must be ensured that potentially hazardous signals (e.g. the opening/closing of a gun) can only be executed if AUT or AUT EXT mode is set and the safety gate is closed. For this, the signals must be mapped accordingly to the variables **bSPOC_UserSafetyActive** and **bSPOC_MotionEnabled**.

3.1 Safety measures for “single point of control”

Overview

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

Components:

- Submit interpreter
- PLC
- OPC Server
- Remote control tools
- Tools for configuration of bus systems with online functionality
- KUKA.RobotSensorInterface
- 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.

T1, T2

In the test modes, the components referred to above (with the exception of the external keyboard/mouse) may only access the industrial robot if the following signals have the following states:

Signal	State required for SPOC
$USER_SAF	TRUE
$SPOC_MOTION_ENABLE	TRUE

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:

- In the test modes, the system variable $OV_PRO must not be written to by the Submit interpreter or the PLC.
- 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.
- If these components are used, outputs that could cause a hazard must be determined in a risk assessment. These outputs must be designed in such a way that they cannot be set without being enabled. This can be done using an external enabling device, for example.

Tools for configuration of bus systems

If these components have an online functionality, they 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.

- WorkVisual from KUKA
- Tools from other manufacturers

Safety measures:

- In the test modes, 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.

4 Installation

4.1 System requirements

Overview

- KR C4
- PROFINET, PROFIBUS or Interbus
- Network connections (network switch, network cable, 100 Mbit network card)

4.2 Installing or updating KUKA.PLC ProConOS 4-1

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

Precondition

- Expert user group
- Software on KUKA.USB data stick

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

> ℹ ProConOS OPC Server 2.1 is automatically installed during the installation of KUKA.PLC ProConOS 4-1.

Procedure

1. Plug in USB stick.
2. Select **Start-up** > **Install additional software** in the main menu.
3. Press **New software**. If a software package that is on the USB stick is not displayed, press **Refresh**.
4. Mark the entry **ProConOS 4-1** and press **Install**. Reply to the request for confirmation with **Yes**. The files are copied onto the hard drive.
5. Repeat step 4 if another software package is to be installed from this stick.
6. Remove USB stick.
7. It may be necessary to reboot the controller, depending on the additional software. In this case, a corresponding prompt is displayed. Confirm with **OK** and reboot the robot controller. Installation is resumed and completed.

LOG file

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

4.3 Uninstalling KUKA.PLC ProConOS 4-1

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

Precondition

- Expert user group

Procedure

1. Select **Start-up** > **Install additional software** in the main menu. All additional programs installed are displayed.
2. Mark the entry **ProConOS 4-1** and press **Uninstall**. Reply to the request for confirmation with **Yes**. Uninstallation is prepared.
3. Reboot the robot controller. Uninstallation is resumed and completed.

LOG file

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

5 Operation

5.1 Controlling ProConOS by means of KRL

Description

ProConOS can be controlled by means of KRL applications using the function **cIoCtl**. The following functions can be executed:

- Start ProConOS
- Stop ProConOS
- Call up status of ProConOS

The function call returns a value of data type INT.

Syntax

```
cIoCtl($Soft_Plc_Cmd, Function, Parameter, StrParameter, pRet)
```

Parameter	Description
Function	Function number
Parameter	Function parameter (data type INT)
StrParameter	Additional string parameter
pRet	Return status (data type INT)

5.1.1 Calling up the status of ProConOS

Syntax

```
cIoCtl($Soft_Plc_Cmd, 1003, 0, " ", tmpInt)
```

The value returned by the function is binary coded. However, at no time will more than one bit be set.

> ⚠ The variable tmpInt must not be a runtime variable. It must be declared in a .dat file.

Return values:

Value	Designation	Description
1	POWER ON	ProConOS is started and no PLC program is loaded.
2	RUN	The PLC program is started.
4	STOP	The PLC program is loaded and not started.
8	HALT	The PLC program has been stopped, e.g. by a breakpoint.
32	LOADING	The PLC program is being loaded.

5.1.2 Powerfail status

Syntax

```
cIoCtl($Soft_Plc_Cmd, 1005, 0, " ", tmpInt)
```

> ⚠ The variable tmpInt must not be a runtime variable. It must be declared in a .dat file.

Return values:

Value	Description
1	Powerfail status is not active.
2	Powerfail status is active.

5.1.3 Starting ProConOS

Syntax

```
cIoCtl($Soft_Plc_Cmd, 1013, argument, " ", tmpInt)
```

⚠ The variable tmpInt must not be a runtime variable. It must be declared in a .dat file.

Argument	Designation	Description
0	Cold start	All variable contents are deleted and re-initialized.
1	Warm start	All variable contents are deleted and re-initialized, except for retentive data.
2	Hot start	Variable contents are not deleted or re-initialized.

Return value:

Value	Description
0	ProConOS has been started.

5.1.4 Stopping ProConOS

Syntax

```
cIoCtl($Soft_Plc_Cmd, 1014, 0, " ", tmpInt)
```

⚠ The variable tmpInt must not be a runtime variable. It must be declared in a .dat file.

Return value:

Value	Description
0	ProConOS has been stopped.

6 Configuration

ProConOS is configured and operational following installation. Ways of modifying the existing configuration are described below.

6.1 Configuring a task priority

Description

The ProConOS priority model describes the fundamental relationships between the priorities and the various ProConOS tasks. The priority indicates a precedence in the allocation of CPU computing time and the use of system resources. The priority of tasks is defined in the initialization file **...Config\User\Common\ProConOS.xml**.

Each task is assigned to a task range. The assignment is preset and cannot be changed. Each task range is assigned a priority. The defined priority is assigned to the allocated task in this range. Each task range is represented by a parameter in the initialization file. The value of the parameter defines the priority of the tasks located in this range.

The following task ranges are available:

- **ProConOS Supervisor tasks (high-level tasks):** This task range contains all tasks which have a higher priority than the highest-priority user task, for example robot tasks.

  Default value of Supervisor tasks: 141

  (>>> 6.3.1 "PrioSvisor parameter" Page 16)
- **ProConOS User tasks and ProConOS Default task (ProConOS-level tasks):** This task range contains the PLC tasks and the Default task.

  Default value of User tasks: 142 to 149

  (>>> 6.3.2 "PRIOUSER parameter" Page 17)
- **ProConOS Internal tasks:**

  Default value of Internal tasks: 200

  (>>> 6.3.4 "PrioIdle parameter" Page 18)
- **ProConOS System tasks (low-level tasks):** Low Level tasks have a lower priority than the Default task.

  Default value of System tasks: 218 to 223

  (>>> 6.3.3 "PRIOSYSTEM parameter" Page 17)

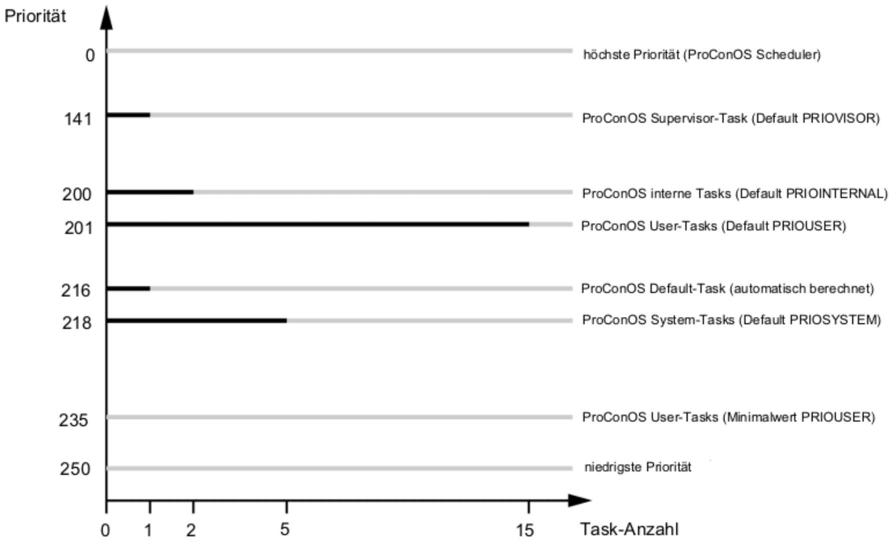


Fig. 6-1

6.2 Configuring the ProConOS/Multiprog interface

Description

A connection to ProConOS can be made via a network card in the robot controller and an external Windows network. This connection can be established using KLI: KLI can route any TCP/IP ports for the KR C, for VxWorks and for external PCs. It is thus possible to access ProConOS from an external PC.

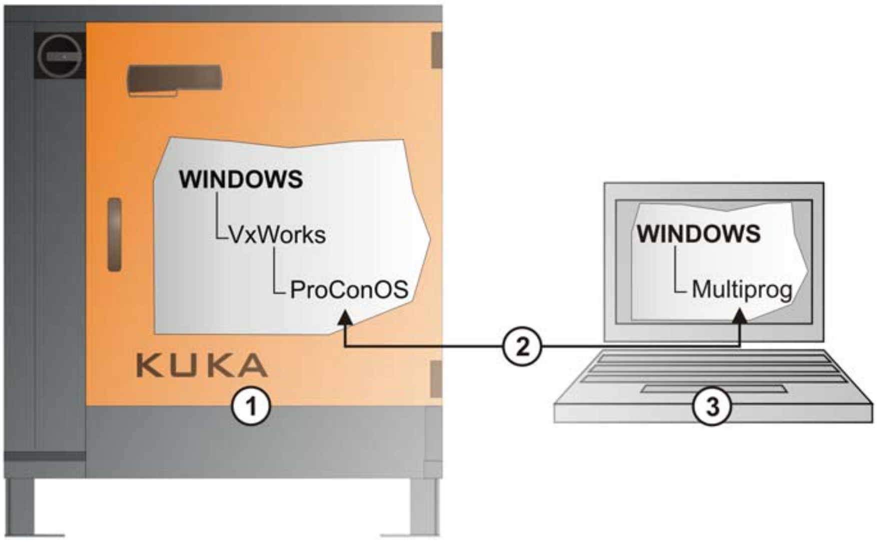


Fig. 6-2: Configuration overview

Item	Description
1	KR C4: ■ Windows operating system ■ VxWorks real-time operating system ■ ProConOS runtime system
2	Connection: KLI
3	External PC/laptop: ■ Windows operating system ■ MULTIPROG PLC development environment ■ WorkVisual offline engineering system

6.3 Configurable ProConOS parameters

Description

The parameters in the initialization file **...Config/User/Common/ProConOS.xml** are described here. The parameter values are used to configure ProConOS during initialization.

6.3.1 PrioSvisor parameter

Description

The value of this parameter specifies the priority of the ProConOS Supervisor task. The ProConOS Supervisor task must have the highest priority of all ProConOS tasks. An example of a Supervisor task is exception handling. The robot controller must be restarted after the value of this parameter is changed.

⚠ The parameter value PrioSvisor must be greater than PRIOUSER.

⚠ The value of the PrioSvisor parameter may be modified only in consultation with KUKA Roboter GmbH.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
PrioSvisor = 0	Highest priority
PrioSvisor = 141	Default priority (default value)
PrioSvisor = 234	Lowest priority

6.3.2 PRIOUSER parameter

Description

The value of this parameter specifies the priority of the ProConOS User tasks. MULTIPROG allows the assignment of 15 priority levels, which can be mapped on ProConOS by means of PRIOUSER. The priority defined via this parameter corresponds to priority 1 in MULTIPROG. The default task has the lowest priority level of the user tasks. The priority of the default task cannot be defined, since it is calculated by ProConOS from the value of the PRIOUSER parameter (DEFAULT TASK = PRIOUSER + 15). The default task receives computing time if no other ProConOS task requires computing time.

⚠ The value of the PRIOUSER parameter must be greater than the value of the PrioSvisor parameter plus 1 (PRIOUSER ≥ PrioSvisor + 1).

⚠ The value of the PRIOUSER parameter may be modified only in consultation with KUKA Roboter GmbH.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
PRIOUSER = 0	Highest priority
PRIOUSER = 151	Default priority (default value)
PRIOUSER = 235	Lowest priority

6.3.3 PRIOSYSTEM parameter

Description

The value of this parameter specifies the priority of the ProConOS System tasks. The functions of the ProConOS System tasks are communication with MULTIPROG, debugging and memory management. System tasks receive the lowest priority level. The robot controller must be restarted after the value of this parameter is changed.

ℹ The value of the PRIOSYSTEM parameter is greater than or equal to the value of the PRIOUSER parameter plus 16 (PRIOSYSTEM ≥ PRIOUSER + 16).

⚠ The value of the PRIOSYSTEM parameter may be modified only in consultation with KUKA Roboter GmbH.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
PRIOSYSTEM = 0	Highest priority
PRIOSYSTEM = 219	Default priority (default value)
PRIOSYSTEM = 250	Lowest priority

6.3.4 PrioIdle parameter

Description

The value of this parameter defines the priority of two tasks which are used for internal management purposes:

- one task that initializes and starts the ProConOS runtime system
- another task that saves the retentive data to the hard drive in the event of a power failure (Powerfail)

The robot controller must be restarted after the value of this parameter is changed.

The priority is freely selectable, but should be adjusted to suit the overall system.

⚠ The value of the PrioIdle parameter may be modified only in consultation with KUKA Roboter GmbH.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
PRIOINTERNAL = 0	Highest priority
PRIOINTERNAL = 190	Default priority (default value)
PRIOINTERNAL = 250	Lowest priority

6.3.5 KukaScheduling parameter

Description

The value of this parameter defines which scheduling method is used.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	KUKA scheduling method is used (NrkTick).
False	KW scheduling method is used (CPUCapMin / CPUCapMax).

6.3.6 KukaSettings/NrkTick parameter

Description

The value of this parameter defines the number of ticks within an interpolation cycle (12 ms) if ProConOS tasks are active.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

6.3.7 KukaSettings/ExempTask parameter

Description The name of the task that must not be adversely affected by the scheduling (= continuous cycle access) is entered in this parameter.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name in the parameter.

6.3.8 FileAccess_RootPath parameter

Description A home directory relative to C:\KRC\ROBOTER is entered in this parameter. The file access modules perform their file operations in this directory. If nothing is entered in the parameter, the path ProConOS_RootPath is used.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the directory in the parameter.

6.3.9 WriteTsmOnStop parameter

Description The value of the parameter defines whether the TSMLog file is saved if the PLC enters the Stop state.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	TsmLog is saved if the PLC enters the STOP state.
False	TsmLog is not saved if the PLC enters the STOP state.

6.3.10 ProgramSize parameter

Description The value of the parameter defines the size (in bytes) of the application program.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

6.3.11 ReloadBootFileAtReconfig parameter

Description The value of the parameter defines whether the boot file is reloaded after a reconfiguration.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	The boot file is reloaded after a reconfiguration.
False	The boot file is not reloaded after a reconfiguration.

6.3.12 IgnoreDeinstalledIoDrv parameter

Description The value of the parameter defines whether an error is signaled if the I/O driver has been uninstalled, but is still active in the ProConOS configuration.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	Error is not signaled.
False	Error is signaled.

6.3.13 WarmStartCheck parameter

Description

The value of the parameter defines whether the retentive data are checked for CRC equality in the case of a HOT start.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	Check is carried out.
False	No check.

6.3.14 DELRETAIN parameter

Description

The value of this parameter defines whether the retentive data are deleted or retained.

The robot controller must be restarted after the value of this parameter is changed.

If retentive variables are inserted in the variable list, all of the subsequent variables are shifted. Thus with DELRETAIN = 1, the consistency of the subsequent retentive variables cannot be ensured. New variables should be inserted only at the end of the variable list.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	Retentive data are set to their initialization values (default value).
False	Retentive data will be retained.

6.3.15 AUTOARCHIVERETAIN parameter

Description

The value of this parameter defines whether the retentive data are saved in the file when the robot controller is shut down.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Range of values	Description
True	The retentive data are saved (default value).
False	The retentive data are not saved.

6.3.16 RETAINSIZE parameter

Description

The value of this parameter defines the size of the area for the retentive data in bytes. This size can be defined if necessary.

If the value of the RETAINSIZE parameter is modified, these settings must be reconfigured in Multiprog.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Value	Description
4096	Default size 4096 bytes

6.3.17 ProConOS_ROOTPATH parameter

Description

The path specified in this parameter defines the directory in which the ProConOS files are located. The directory must be read- and write-enabled.

The robot controller must be restarted after the path specification is changed.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name in the parameter.

Syntax	Description
ProConOS_ROOTPATH = ProConOS	Path relative to C:\KRC\ROBOTER

6.3.18 SIGNALFILES parameter

Description

The SIGNALFILES parameter is used to signal *.dat files to the ProConOS system. These files are used to access I/O data symbolically. The parameter is preset during setup with the following files:

- KRC\R1\System\$CONFIG.DAT
- KRC\STEU\MADA\$MACHINE.DAT

These files can be expanded.

The function is used by the following libraries:

- AutoExLib
- ExtensionLib

Procedure

- Open the file ...\KRC\IRoboter\Config\User\Common\PROCONOS.xml in a text editor and modify the parameter.

6.3.19 SIGNALFILEREAD parameter

Description

The signal files in the parameter **SIGNALFILES** are read in accordance with the entry **SIGNALFILEREAD**.

Procedure

- Open the file ...Config\User\Common\ProConOS.xml in a text editor and enter the name for the parameter.

Syntax	Description
SIGNALFILEREAD = START	The signal files are read once by ProConOS on booting. If the area after “=” is left blank or the parameter is deleted, the signal files are read every time the ProConOS program execution is started.

ℹ If Automatic External interface signals are reconfigured, ProConOS must be stopped and restarted.

7 Diagnosis

7.1 Displaying diagnostic data

Procedure

1. Select **Diagnosis** > **Diagnostic monitor** in the main menu.
2. Select the **ProConOS** module in the **Module** box.

   Diagnostic data are displayed for ProConOS.

Description

The following diagnostic data are displayed:

Name	Description
Run status	ProConOS status: ■ **SPS_ON**: ProConOS is started and no PLC program is loaded. ■ **SPS_RUN**: The PLC program is started. ■ **SPS_STOP**: The PLC program is loaded and not started. ■ **SPS_DEBUGGED**: The PLC program is stopped. ■ **SPS_LOADING**: The PLC program is being loaded.
Ticks per second	Ticks triggered per second by the PLC
ProConOS system tick status	Status of the ProConOS system ticks: ■ **1**: OK ■ **0**: Error
ProConOS system tick	Consecutive counter of the ProConOS system ticks
ProConOS RT tick	ProConOS system ticks in seconds
Project status	Internal status of the project
Default task cycle time	Cycle time of the ProConOS default task (in ms)
ProConOS project source code: ZIP name	■ **Exist**: ProConOS project exists as a ZIP file on the target controller. ■ **None**: ProConOS project does not exist on the target controller.
Name of the boot project	Name of the boot project
Project name	Name of the ProConOS project
Project configuration	Hardware configuration in the project
Resource name	Name of the hardware resource
Number of errors	Number of errors in the ProConOS error memory
Pcos_Error(x)	Error in the ProConOS memory. The first 10 errors are displayed with error code.
Number of user tasks	Number of tasks defined by the user in the ProConOS project
User_Task_Info(x)	Information about a user task: ■ Name ■ Watchdog ■ Type of task: event, cyclical, default A maximum of 16 user tasks can be configured.

Name	Description
User_Task_Statistics(x)	Statistics about a user task: - Name - Cycle time - Delay
Number of system tasks	Number of system tasks (max. 12)
Sys_Task_Info(x)	Information about a system task: - Name - Priority - Mode - Current status
Kernel version	Version of the ProConOS kernel
Firmware version	KUKA firmware version of ProConOS
ProConOS root dir	Directory entered in the ProConOS_ROOTPATH parameter.
ProConOS file access dir	Directory entered in the FileAccess_RootPath parameter.
Configured program size	Value entered in the ProgramSize parameter
Configured retain size	Value entered in the RETAINSIZE parameter
Max. available memory block	Maximum contiguous block of memory available in the system (in bytes)
Program memory	Program memory available for ProConOS (in bytes)
System data memory	Memory available for system data (in bytes)
SPOC: UserSaveActive	Status of the variable bSPOC_UserSafetyActive: - **1**: Not safe - **0**: Safe
SPOC: MotionEnabled	Status of the variable bSPOC_MotionEnabled: - **1**: Motion is enabled. - **0**: Motion is not enabled.
Retain data valid	- **1**: ProConOS data that were saved when the system was shut down are valid. - **0**: ProConOS data that were saved when the system was shut down are not valid.
Exemp task	Name of the task entered in the KukaSettings/ExempTask parameter
Scheduling ticks	Value entered in the KukaSettings/NrkTick parameter
KRC signal read mode	- **each START**: The signal files entered in the SIGNALFILES parameter are read every time ProConOS is started. - **each BOOT**: The signal files entered in the SIGNALFILES parameter are read every time the controller is booted.

8 KUKA Service

8.1 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
- Application used
- Any external axes used
- Description of the problem, duration and frequency of the fault

8.2 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