**Kuka SA Robot Driver Deployment (KRC4)**

**When Customer Has a R**ail

**NOTE: The instructions specific to RAIL implementation are those developed for a KUKA KRC2 controller running under KSS 5.5. The extent to which these instructions are applicable to a KUKA KRC4 controller running under KSS 8.2 has not been verified and caution must be observed with any application of these instructions to a KUKA KRC4 controller.**

**Required Setup**

As of this version (SA Robot Driver Version 2013.09.19 Kuka Rail Support), we support external axis E1. **THE RAIL MUST BE DECOUPLED KINEMATICALLY FROM THE ARM**. This is done by editing the $Machine.dat file so this line reads as follows…

$EX\_KIN={ET1 #NONE,ET2 #NONE,ET3 #NONE,ET4 #NONE,ET5 #NONE,ET6 #NONE} ;

Anything other than #NONE means the external axis is coupled to the robot arm, and you cannot safely run SARobotDriver until this is changed.

In order to operate the brakes on your robot safely, the customer should verify the settings in the line

$BRK\_MODE=

with their Kuka representative. The representative should be consulted to ensure the kinematics decoupling of the rail and robot arm as well.

 **To Test**

To see whether the system has the Required Setup, simply move the rail in external control software to a position a little away from the zero or home position. Make certain the path to the home position is clear, and run the SA Robot Driver. Perform a Cartesian move from SA RobotDriver, being sure to vary the axis along the rail. If the robot is properly decoupled, this will not change the robot’s position on the rail. If this condition is met, then carefully run SA RobotDriver and try moving the robot again in Cartesian space, especially along the rail axis, and then try driving external Axis E1 to a valid position. (Note the ui should show the minimum and maximum limits for the rail’s travel in inches, as well as the current rail position. You should monitor the values in $POS\_ACT and $AXIS\_ACT on the KR C2 controller.) Likewise, when changing the robot’s position on the rail, this should not change the pose of the robot arm.

If there are any doubts regarding any of the above requirements, DO NOT RUN SA RobotDriver until they are resolved.

Installation

1. **Requirements for PC Running SA RobotDriver (Remote Access Computer)**
	1. Windows 7 Professional 64-bit
	2. Localization Settings – Must use ‘.’ as decimal point, not ‘,’
	3. The SA RobotDriver application must be installed (version 2013.11.11 or later).
2. **Requirements for KUKA KRC4 Controller**
	1. The KUKA EthernetKRL 2.1 technology package must be installed as per the documentation provided with this technology package.
	2. As per the documentation for the EthernetKRL 2.1 technology package, an IP address must be established for the EthernetKRL 2.1 communications. For the testing performed using the Office PC simulator, the IP address established was “172.18.1.148” as shown below.



* 1. During network setup for testing, it was necessary to review and modify the Network configuration. This is accessed by activating the “Start-up -> Network configuration” as shown.



* 1. The network configuration established on the first network configuration display was as shown. To address additional advanced configuration options, the “Activate” button was selected.



* 1. The advanced network configuration options established were as shown. Note that that this is simply an example. Be advised that the “Internal subnets” should NOT be used for this interface.



* 1. The following files must be installed on the KRC4 controller in the directories indicated
		1. C:\KRC\ROBOTER\KRC\R1\PROGRAM\SAKRL.src (\*.dat)
		2. C:\KRC\ROBOTER\KRC\ R1\PROGRAM\SAKRL\_Ethernet.src (\*.dat)
		3. C:\KRC\ROBOTER\KRC\ R1\PROGRAM\SAKRL\_EventNotice.src (\*.dat)
		4. C:\KRC\ROBOTER\KRC\ R1\PROGRAM\movetojointset.src (\*.dat)
		5. C:\KRC\ROBOTER\KRC\ R1\PROGRAM\movetoframe.src (\*.dat)
		6. C:\KRC\ROBOTER\KRC\ R1\PROGRAM\MoveToExternalAxisSet.src (\*.dat)
		7. C:\KRC\ROBOTER\Config\User\Common\EthernetKRL\SAKRL.xml

NOTE: For this this installation, the global variables used by the SAKRL module running on the KUKA KRC4 controller have been moved from the “$config.dat” file to the “SAKRL.dat” file.

* 1. The “**SAKRL.xml**” file must be modified to set the IP address of the Remote Access Compter (the PC on which the SA RobotDriver application will run). Refer to **Appendix A** for configuration notes.
1. **Server application and data format files on remote computer (all in SA install)**
	1. SARobotDriver.exe (uses PC’s secondary fixed IP address for KRL/XML connection). On the remote computer, an IP address must be established that will be visible to the KUKA KRC4 controller Ethernet KRL interface. In the example configuration provided, this address is:

IP Address: 172.18.1.149

Mask: 255.255.0.0

Gateway: (NONE)

DNS: (NONE)

* 1. KukaCoreInterface.dll
	2. SA Robot Driver.exe.manifest
	3. RACData.xml

Running SA Robot Driver on the Kuka KRC4

**1) On the Teach Pendant, make sure the program SAKRL (.krl) is selected.**

 **2) On the SmartHMI Pendant, in the upper left of the window, ensure that the “S” (submit interpreter status) indicator is green, the “I” (I/O status for Drives/Motors) is green, and the “R” (program run status) is either yellow or black.**

**Press the “I” motor enable button on the SmartHMI Pendant if the “I” indicator is not green.
Go to the Program menu, and select “Reset Program” if the R indicator is yellow or black.

3) Add your robot model to SA, and run SA Robot Driver.exe. Press “Connect”, and select the Kuka robot from the drop down. You’ll see a progress dialog, informing you that the interface is waiting for a connection to SAKRL.

4) On the SmartHMI Pendant, press and hold the green “Start” button (bottom of teach pendant). Hold until the robot has completed the move to the Kuka home position.

5) The Program Run State Indicator “R” at the top of the SmartHMI Pendant window should be red. Now press the green “Start” button again. The Program Run Indicator “R” should now be green, and the “waiting for connection” on the PC should go away. You are now running the robot driver.

6) You should ALWAYS perform a joint space move OUT of the Kuka home position first, because it is a singularity.**

**NOTE: For the KRC4 SAKRL application as it is currently configured, the initial mandatory PTP move is a move to its current position. This means that there should be no initial motion of the robot – it also means that the robot will not be moving to a predefined “safe” position. If it is desired to always perform a PTP motion to the default HOME position at the start of the application, then the following two lines in the SAKRL.src file should be modified such as to activate the “PTP XHOME” command.**

**;PTP XHOME ; Remove leading “;” to activate this line**

**PTP $AXIS\_ACT ; Insert leading “;” to deactivate this line**

**Appendix A – SAKRL.xml**

-------------------------------------------------------------------------------

<ETHERNETKRL>

 <CONFIGURATION>

 <EXTERNAL>

 **<IP>172.18.1.149</IP>**

 **<PORT>6008</PORT>**

 </EXTERNAL>

 <INTERNAL>

 <ALIVE Set\_Flag="1"/>

 </INTERNAL>

 </CONFIGURATION>

 <RECEIVE>

 <XML>

 <ELEMENT Tag="RACData/MsgType" Type="INT"/>

 <ELEMENT Tag="RACData/ProgNum" Type="INT"/>

 <ELEMENT Tag="RACData/InspectionIndex" Type="INT"/>

 <ELEMENT Tag="RACData/Position/XPos" Type="REAL"/>

 <ELEMENT Tag="RACData/Position/YPos" Type="REAL"/>

 <ELEMENT Tag="RACData/Position/ZPos" Type="REAL"/>

 <ELEMENT Tag="RACData/Position/APos" Type="REAL"/>

 <ELEMENT Tag="RACData/Position/BPos" Type="REAL"/>

 <ELEMENT Tag="RACData/Position/CPos" Type="REAL"/>

 <ELEMENT Tag="RACData/Closure/Status" Type="INT"/>

 <ELEMENT Tag="RACData/Closure/Turn" Type="INT"/>

 <ELEMENT Tag="RACData/Joints/A1" Type="REAL"/>

 <ELEMENT Tag="RACData/Joints/A2" Type="REAL"/>

 <ELEMENT Tag="RACData/Joints/A3" Type="REAL"/>

 <ELEMENT Tag="RACData/Joints/A4" Type="REAL"/>

 <ELEMENT Tag="RACData/Joints/A5" Type="REAL"/>

 <ELEMENT Tag="RACData/Joints/A6" Type="REAL"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex1" Type="REAL"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex2" Type="REAL"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex3" Type="REAL"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex4" Type="REAL"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex5" Type="REAL"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex6" Type="REAL"/>

 <ELEMENT Tag="RACData/Velocity" Type="REAL"/>

 <ELEMENT Tag="RACData/Acceleration" Type="REAL"/>

 <ELEMENT Tag="RACData/JtVelocity" Type="REAL"/>

 <ELEMENT Tag="RACData/JtAcceleration" Type="REAL"/>

 <ELEMENT Tag="RACData/ExAxisVelocity" Type="REAL"/>

 <ELEMENT Tag="RACData/ExAxisAcceleration" Type="REAL"/>

 <ELEMENT Tag="RACData/AtHome" Type="BOOLEAN"/>

 <ELEMENT Tag="RACData/AtStart" Type="BOOLEAN"/>

 <ELEMENT Tag="RACData/Error" Type="STRING"/>

 <ELEMENT Tag="RACData/MsgCount" Type="INT"/>

 <ELEMENT Tag="RACData" Set\_Flag="2"/>

 </XML>

 </RECEIVE>

 <SEND>

 <XML>

 <ELEMENT Tag="RACData/MsgType"/>

 <ELEMENT Tag="RACData/ProgNum"/>

 <ELEMENT Tag="RACData/InspectionIndex"/>

 <ELEMENT Tag="RACData/Position/XPos"/>

 <ELEMENT Tag="RACData/Position/YPos"/>

 <ELEMENT Tag="RACData/Position/ZPos"/>

 <ELEMENT Tag="RACData/Position/APos"/>

 <ELEMENT Tag="RACData/Position/BPos"/>

 <ELEMENT Tag="RACData/Position/CPos"/>

 <ELEMENT Tag="RACData/Closure/Status"/>

 <ELEMENT Tag="RACData/Closure/Turn"/>

 <ELEMENT Tag="RACData/Joints/A1"/>

 <ELEMENT Tag="RACData/Joints/A2"/>

 <ELEMENT Tag="RACData/Joints/A3"/>

 <ELEMENT Tag="RACData/Joints/A4"/>

 <ELEMENT Tag="RACData/Joints/A5"/>

 <ELEMENT Tag="RACData/Joints/A6"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex1"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex2"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex3"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex4"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex5"/>

 <ELEMENT Tag="RACData/ExternalAxes/Ex6"/>

 <ELEMENT Tag="RACData/Velocity"/>

 <ELEMENT Tag="RACData/Acceleration"/>

 <ELEMENT Tag="RACData/JtVelocity"/>

 <ELEMENT Tag="RACData/JtAcceleration"/>

 <ELEMENT Tag="RACData/ExAxisVelocity"/>

 <ELEMENT Tag="RACData/ExAxisAcceleration"/>

 <ELEMENT Tag="RACData/AtHome"/>

 <ELEMENT Tag="RACData/AtStart"/>

 <ELEMENT Tag="RACData/Error"/>

 <ELEMENT Tag="RACData/MsgCount"/>

 </XML>

 </SEND>

</ETHERNETKRL>

-------------------------------------------------------------------------------

NOTES:

1. In this example, the channel assigned for the SA RobotDriver is “SAKRL” (taken from the name of the file).
2. In this example, the IP address of the SA RobotDriver is “172.18.1.150”.
3. In this example, the port address of the SA RobotDriver is “6008”.
4. Note that “<ELEMENT Tag="RACData" Set\_Flag="2"/>” at the end of the “Receive” data. This instantiates the mechanism by which the KUKA KRC4 KRL application detects the arrival of newly received data. When the last element of the received data has been read, the KUKA KRC4 KRL $FLAG[2] will be set. The communication handshake in this application is such that when a message is sent from the KUKA KRC4 controller, $FLAG[2] is cleared -- when a reply to this message is received, $FLAG[2] is set.
5. It is recommended that the **“C:\KRC\ROBOTER\KRC\R1\TP\EthernetKRL\EthernetKRL.dat”** file be modified as follows:

**~~CONST BOOL SHOWMSG=TRUE~~**

**CONST BOOL SHOWMSG=FALSE**

1. It is recommended that the **“C:\KRC\ROBOTER\KRC\R1\TP\EthernetKRL\EthernetKRL.src”** file be modified as follows:

DEF showMsgDepend(Ret:OUT)

DECL EKI\_STATUS Ret

IF (Ret.Msg\_no B\_AND ERRMSG\_FORBIDDEN)==0 THEN

**~~EKI\_CHECK(Ret,#QUIT)~~**

**EKI\_CHECK(Ret,#NOTIFY)**

ENDIF

Ret.Msg\_no = Ret.Msg\_no B\_AND ERRMSG\_ALL

END

1. It is recommended that the **“C:\KRC\ROBOTER\Config\User\Common\Logging\_EthernetKRL.xml”** file be modified as follows:

 **~~<Class Name="EKILogger" LogLevel="warning" />~~**

 **<Class Name="EKILogger" LogLevel="error" />**