Digital Twin KR3
KVP Connection

Ahmed Ibrahim Almohamed

Tuesday 6th August, 2024

1 Introduction

JOpenShowVar is a versatile, open-source communication interface written in Java, designed to be cross-platform. It facilitates the reading and writing of all variables associated with a controlled manipulator. While it is primarily suited for soft real-time applications, JOpenShowVar offers significant flexibility for researchers. It supports the integration of various input devices and sensors, which can be used to explore and develop novel control methodologies. The library is highly compatible, functioning seamlessly with all Kuka industrial robots that utilize either the KR C4 or KR C2 controllers. This makes JOpenShowVar an invaluable tool for experimentation and innovation in robotic control systems, enabling the implementation of alternative control strategies and the advancement of research in automation and robotics. [1]

KUKAVARPROXY is a robust multi-client server capable of serving up to 10 clients simultaneously. It implements the KUKA CrossComm class, which provides a wide range of functionalities essential for controlling and managing KUKA robots. These functionalities include selecting or canceling specific programs, detecting errors and faults, renaming program files, saving programs, resetting I/O drivers, as well as reading and writing variables. [1]

The communication protocol between KUKAVARPROXY and JOpenShowVar utilizes TCP/IP, specifically through the exchange of specially formatted strings on the TCP/IP layer.

KUKAVARPROXY actively listens on TCP port 7000, awaiting connections. Once a connection is established, the server is prepared to handle any read or write requests from the client. This setup allows for efficient and reliable message exchange between KUKAVARPROXY and JOpenShowVar, facilitating seamless communication and control over the robotic systems. [1]

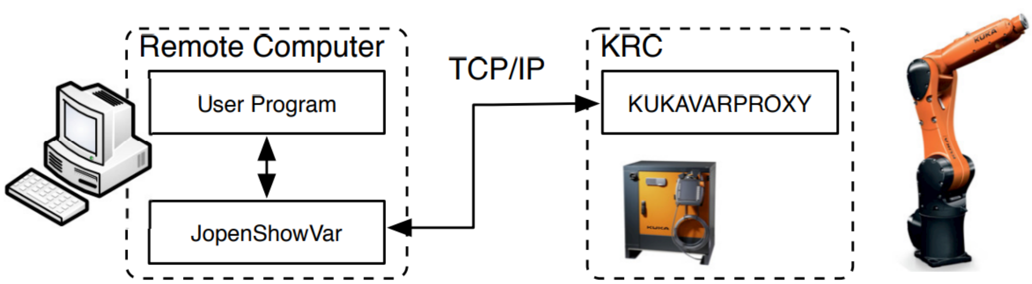

Figure 1: architecture of JOpenShowVar

2 Reading and Writing

2.1 Messages

In the KUKAVARPROXY protocol the main reading and writing is done to the global variables of the KUKA controller (in this case KRC4). This means that we read or write to the global variables of the KUKA controller. According to [2] is a list of all the global variables that we are able to control, together with their descriptions. In the appendix of this file are the tables of all the global variables with the data types and their description.

Appendices

A Tables of Global Variables

The parent variable is listed with its corresponding description and datatype, and any associated child variables are shown directly beneath it with an indentation for more information check [2].

Table 1: Name	Description	Datatype
$A4PAR	Set axis 4 parallel to the last rotational main axis	INT
$ABS_ACCUR	Switch absolutely accurate robot model on/off	BOOL
$ABS_CONVERT	Conversion of point coordinates into absolutely accurate robot model	BOOL
$ABS_RELOAD	Reload absolutely accurate robot model	BOOL
$ACC	Accelerations in the ‘ADVANCE‘ run	STRUC
CP	Path acceleration in the ‘ADVANCE‘ run	REAL
ORI1	Swivel acceleration in the ‘ADVANCE‘ run	REAL
ORI2	Rotational acceleration in the ‘ADVANCE‘ run	REAL
$ACC_ACT_MA	Limit value of axial command acceleration	INT
$ACC_AXIS[]	Acceleration of the axes ‘A1..A6‘ in the ‘ADVANCE‘ run	INT[6]
$ACC_AXIS_C[]	Acceleration of the axes ‘A1..A6‘ in the ‘MAIN‘ run	INT[6]
$ACC_C	Accelerations in the ‘MAIN‘ run	STRUC
CP	Path acceleration in the ‘MAIN‘ run	REAL
ORI1	Swivel acceleration in the ‘MAIN‘ run	REAL
ORI2	Rotational acceleration in the ‘MAIN‘ run	REAL
$ACC_CAR_ACT	The current values of the acceleration components and the total acceleration	STRUC
X	‘X‘ component of the acceleration expressed in **$ACC_CAR_TOOL** frame.	REAL
Y	‘Y‘ component of the acceleration expressed in **$ACC_CAR_TOOL** frame.	REAL

Table 2: Name	Description	Datatype
Z	‘Z‘ component of the acceleration expressed in **$ACC_CAR_TOOL** frame.	REAL
ABS	Acceleration magnitude.	REAL
$ACC_CAR_LIMIT	Maximum permissible value for the acceleration components and the total acceleration	STRUC
X	Maximum permissible value for the ‘X‘ component of the acceleration expressed in **$ACC_CAR_TOOL** frame.	REAL
Y	Maximum permissible value for the ‘Y‘ component of the acceleration expressed in **$ACC_CAR_TOOL** frame.	REAL
Z	Maximum permissible value for the ‘Z‘ component of the acceleration expressed in **$ACC_CAR_TOOL** frame.	REAL
ABS	Maximum permissible value for the acceleration magnitude.	REAL
$ACC_CAR_MAX	Saves the greatest absolute values of **$ACC_CAR_ACT**	STRUC
$ACC_CAR_STOP	Activates/Deactivates stop reaction when the value specified in **$ACC_CAR_LIMIT** is exceeded.	BOOL
$ACC_CAR_TOOL	A point on the tool mounted on the robot at which the current effective acceleration is measured	STRUC
$ACC_EXTAX[]	Acceleration of the external axes ‘E1..E7‘ in the ‘ADVANCE‘ run	INT[6]
$ACC_EXTAX_C[]	Acceleration of the external axes ‘E1..E7‘ in the ‘MAIN‘ run	INT[6]
$ACC_MA	Maximum values for path, swivel and rotational accelerations	STRUC
CP	Maximum path acceleration	REAL
ORI1	Maximum swivel acceleration	REAL
ORI2	Maximum rotational acceleration	REAL
$ACC_OV	Data for acceleration with changes of override	STRUC
CP	Path acceleration with change of override	REAL

Table 3: Name	**Description**	**Datatype**
ORI1	Swivel acceleration with change of override	REAL
ORI2	Rotational acceleration with change of override	REAL
$ACT_BASE	Number of the current BASE system	INT
$ACT_EX_AX	Number of the current external base kinematic system	INT
$ACT_TOOL	Number of the current tool coordinate system	INT
$ACT_VAL_DIF	Maximum permissible difference of encoder actual values when switching on system.	INT
$ADAP_ACC	Activation of acceleration adaptation.	ENUM
$ADVANCE	Specification of the ‘ADVANCE‘ run.	INT
$ALARM_STOP	Specification of the ‘ADVANCE‘ run.	SIGNAL
$ANA_DEL_FLT	Analog output filter	ENUM
$ANIN[]	Analog inputs ‘$ANIN[1]..$ANIN[8]‘	REAL[8]
$ANOUT[]	Analog outputs ‘$ANIN[1]..$ANIN[16]‘	REAL[16]
$APO_DIS_PTP[]	Maximum approximation distance for ‘PTP‘ motions.	REAL[12]
$ASYNC_AX	Motion input for asynchronous external axes ‘E1..E6‘, ‘negative‘ or ‘positive‘ direction.	SIGNAL
$ASYNC_AXi_M	Motion input for asynchronous external axes ‘E1..E6‘, ‘negative‘ direction	SIGNAL
$ASYNC_AXi_P	Motion input for asynchronous external axes ‘E1..E6‘, ‘positive‘ direction	SIGNAL
$ASYNC_AXIS	Bit arrays to switch external axes to asynchronous mode	INT
$ASYNC_FLT	Filter for asynchronous external axes	INT

Table 4: Name	Description	Datatype
$ASYNC_MODE	Mode for asynchronous external axes	INT
$ASYNC_OPT	Option flag for *asynchronous axes are possible*	BOOL
$ASYNC_STATE	Current asynchronous motion execution state	ENUM
$ASYNC_T1_FAST	Control of the velocity reduction factor in ‘Test 1‘ mode	INT
$ASYS	Assignment of the jog keys	ENUM
$AUT	Signal declaration ‘Automatic‘ mode	SIGNAL
$AUX_POWER	Signal declaration for external power supply	SIGNAL
$AXIS_ACT	Current axis-specific robot position	STRUC
$AXIS_ACTMOD	Display of axis angle modulo 180°	STRUC
$AXIS_BACK	Start position of the current motion block, axis-specific	STRUC
$AXIS_CAL	Display whether axis is referenced	STRUC
$AXIS_DIR[]	Direction of rotation for axis ‘A1..A6‘ and external axis ‘E1..E6‘	INT[12]
$AXIS_FOR	Target position of the current motion block, axis-specific	STRUC
$AXIS_HOME[]	Definition of the various home positions	STRUC
$AXIS_INC	Incremental actual values of the axes	STRUC
$AXIS_INT	Robot position at the time of an interrupt	STRUC
$AXIS_JUS	Display whether axis is mastered	STRUC
$AXIS_RESO[]	Resolution of the position sensing system	INT[12]

Table 5: Name	Description	Datatype
$AXIS_RET	Axis positions when leaving the programmed path, axis-specific	STRUC
$AXIS_SEQ[]	Change in sequence of axis ... to axis ...	INT[12]
$AXIS_TYPE[]	Axis type identification	INT[12]
$AXWORKSPACE[]	Definition of axis-specific workspace monitoring	STRUC
$BASE	Base coordinate system in relation to the world coordinate system in the ‘ADVANCE‘ run.	STRUC
$BASE_C	Base coordinate system in relation to the world coordinate system in the ‘MAIN‘ run	STRUC
$BASE_KIN[]	External kinematic / axes in base	CHAR[29]
$BOUNCE_TIME	Bounce time for EMT signals	INT
$BRAKE_SIG	Bit array for axis ‘A1..A6‘ and external axis ‘E1..E6‘ brakes	INT
$BRK_DEL_COM	Time after which the axis brakes are closed on completion of positioning during jogging	INT
$BRK_DEL_EX	Brake delay time for external axes	INT
$BRK_DEL_PRO	Time after which the axis brakes are closed on completion of posi- tioning in the program	INT
$BRK_MAX_TM	Maximum deceleration time for path-maintaining Emergency Stop	INT
$BRK_MODE	Brake control mode	INT
$BRK_OPENTM	Time delay of command value output after axis brakes have been opened	INT
$BUS_PAR	L2 bus interface (KRC32)	STRUC
$CABLE2_MON	Additional motor cable monitoring	BOOL
$CAL_DIFF	Mastering difference for EMT mastering with check run	INT

Table 6: Name	Description	Datatype
$CALP	Reference point offset between mathematical zero point and encoder zero point	STRUC
$CIRC_TYPE	Orientation control with CIRC blocks in the ‘ADVANCE‘ run	ENUM
$CIRC_TYPE_C	Orientation control with CIRC blocks in the ‘MAIN‘ run	ENUM
$CMD	Display assignment number (handle) for command channel	INT
$COM_NAME	Command which is to be processed after next start	CHAR[486]
$COM_VAL_MI[]	Limitation of command speed for axis ‘A1..A6‘ and external axis ‘E1..E6‘	REAL[12]
$CONF_MESS	Signal declaration for ‘reset acknowledgement messages‘.	SIGNAL
$COSYS	Coordinate system for jogging	ENUM
$COUNT_I[]	Freely usable integer variables	INT[32]
$COUP_COMP[,]	Axis coupling factors	STRUC
$CP_VEL_TYPE	Reduction of the CP path velocity	ENUM
$CP_VEL_TYPE	Reduction of the CP path velocity	ENUM
$CPVELREDMELD	Generation of message if path velocity reduced	INT
$CURR_ACT[]	Actual current of axes	REAL[12]
$CURR_CAL[]	Current calibration of axis in the power module	REAL[12]
$CURR_COM_EX[]	Current limitation for external axes in jog mode	REAL[6]
$CURR_LIM[]	Current limitation	INT[12]
$CURR_MAX[]	Maximum effective current on power module output	REAL[12]

Table 7: Name	Description	Datatype
$CURR_MON[]	Permissible rated current	REAL[12]
$CURR_RED[,]	Current limitation for axes in ‘%‘ of the maximum current	REAL[12,2]
$CYC_DEFi[]	Input text for the corresponding cyclical flag	CHAR[470]
$CYCFLAG[]	Cyclical flags	BOOL[32]
$DATA_SERx	Number of serial receive messages read in the channel x buffer	INT
$DATA_INTEGRITY	A variable of type ‘SIGNAL‘ is output either as groups of bits or one bit at a time	BOOL
$DATAPATH[]	Name of the ‘SRC‘ file whose variables in the data list are to be accessed using the variable modification function	CHAR[16]
$DATE	System time and system date	STRUC
$DECEL_MB	Deceleration time during maximum braking	REAL
$DEF_A4FIX	Fixing of axis 4 when palletizing	BOOL
$DEF_FLT_CP	Default filter for CP motion	INT
$DEF_FLT_PTP	Default filter for PTP motion	INT
$DEF_L_CM	Center of mass frame for the default load on the flange in the flange coordinate system	STRUC
X	Offset in the ‘X‘ direction	REAL
Y	Offset in the ‘Y‘ direction	REAL
Z	Offset in the ‘Z‘ direction	REAL
A	Rotation about the ‘Z‘ axis	REAL
B	Rotation about the ‘Y‘ axis	REAL

Table 8: Name	Description	Datatype
C	Rotation about the 'X' axis	REAL
$DEF_L_J	Default moment of inertia of the load on the flange in the default center of mass coordinate system of the load	STRUC
$DEF_L_M	Default mass of the load on the flange	REAL
$DEF_LA3_CM	Center of mass frame for the default mass of the supplementary load on axis 3 in the flange coordinate system	STRUC
X	Offset in the 'X' direction	REAL
Y	Offset in the 'Y' direction	REAL
Z	Offset in the 'Z' direction	REAL
A	Rotation about the 'Z' axis	REAL
B	Rotation about the 'Y' axis	REAL
C	Rotation about the 'X' axis	REAL
$DEF_LA3_J	DDefault moment of inertia of the supplementary load on axis 3	STRUC
$DEF_LA3_M	Default mass of the supplementary load on axis 3	REAL
$DEF_OV_JOG	Default value for override in jog mode	INT
$DEVICE	Operator control device status	ENUM
$DH_4	Denavit-Hartenberg parameters for the wrist (frame between 'A4' and 'A5')	STRUC
DHART_A	N/A	REAL
DHART_D	N/A	REAL
DHART_ALPHA	N/A	REAL

Table 9: Name	Description	Datatype
$DH_5	Denavit-Hartenberg parameters for the wrist (frame between ‘A5‘ and ‘A6‘)	STRUC
DHART_A	N/A	REAL
DHART_D	N/A	REAL
DHART_ALPHA	N/A	REAL
$DIGINi	Assignment of digital inputs 1 to 6	SIGNAL
$DIGINiCODE	Defines whether or not the value for ‘$DIGINi‘ is preceded by a sign	ENUM
$DIR_CAL	Defines the referencing direction for each axis ‘A1..A6‘ and external axis ‘E1..E6‘	INT
$DIS_WRP1	Average distance of wrist point from singularity 1 (Alpha 1 singularity)	REAL
$DIS_WRP2	Average distance of wrist point from singularity 2 (Alpha 5 singularity)	REAL
$DISPLAY_REF	New form output when **$DISPLAY_VAR** is changed	BOOL
$DISPLAY_VAR[]	Observable variables	STRUC
$DIST_NEXT	Distance still to be covered to the next point	REAL
$DISTANCE	Curve length, CP motion	REAL
$DRIVE_CART	Option bit: PTP points with Cartesian coordinates	BOOL
$DRIVE_CP	Option bit: Cartesian robot motion possible (LIN, CIRC)	BOOL
$DRIVES_OFF	Signal declaration ‘Drives OFF‘	SIGNAL
$DRIVES_ON	Signal declaration ‘Drives ON‘	SIGNAL
$DSECHANNEL	Assignment of axes to channels of the digital servoelectronics (DSE)	INT

Table 10: Name	Description	Datatype
$DYN_DAT[]	Model data of the robot for acceleration adaptation, higher motion profile and kinetic energy	REAL[350]
$OV_PRO	Program override	INT
$PRO_MODE	Program run mode dependent on **$INTERPRETER**	ENUM
$PRO_MODE0	Program run mode of the ‘SUBMIT‘ interpreter	ENUM
$PRO_MODE1	Program run mode of the ‘ROBOT‘ interpreter	ENUM
$PRO_STATE	Process state dependent on **$INTERPRETER**	ENUM
$PRO_STATE0	Process state of the ‘SUBMIT‘ interpreter	ENUM
$PRO_STATE1	Process state of the ‘ROBOT‘ interpreter	ENUM
$PRO_NAME[]	Process name dependent on **$INTERPRETER**	CHAR[24]
$PRO_NAME0[]	Process name of the ‘SUBMIT‘ interpreter	CHAR[24]
$PRO_NAME1[]	Process name of the ‘ROBOT‘ interpreter	CHAR[24]

References

[1] F. Sanfilippo, L. I. Hatledal, H. Zhang, M. Fago, and K. Y. Pettersen, "Controlling kuka industrial robots: Flexible communication interface jopenshowvar," *IEEE robotics & automation magazine*, vol. 22, no. 4, pp. 96–109, 2015.

[2] OpenKuka, "System variables in krl (kuka robot language)," https://github.com/OpenKuka/openkuka.github.io/blob/master/krl/reference/yaml/System.Variables n.d., accessed: 2024-08-09.