# Table of contents

1. **Introduction** .................................................. 4

2. **Implementation details** .................................. 5

3. **OPC UA Server setup** ....................................... 6
   3.1 Install OPC UA server ..................................... 6
      3.1.1 Requirements ........................................ 6
      3.1.2 Load the OPCUA.out application file ............... 7
      3.1.3 Server certificate ................................... 8
   3.2 Licensing OPC UA server .................................. 9
      3.2.1 Checking for a valid license ....................... 9
      3.2.2 Request a license file from Yaskawa Europe .... 10
      3.2.3 License activation ................................. 11

4. **Connecting OPC UA Server on port 16448** ............... 13

5. **OPC UA Address space / Information model** ............ 15
   5.1 General overview ....................................... 15
      5.1.1 MotionDevices and controllers .................... 15
      5.1.2 ProzessData ...................................... 16
      5.1.3 Methods .......................................... 16
   5.2 NodeSet ................................................ 19

6. **Configuration file OPCCONFIG.JBI** ....................... 27
   6.1 Configuration default setting .......................... 27
   6.2 Basic structure ....................................... 28
      6.2.1 AppSettings ....................................... 28
      6.2.2 VendorSettings ................................... 33
   6.3 Introduction to the OPCUA Config Editor .............. 36
      6.3.1 Requirements ..................................... 36
      6.3.2 App settings ..................................... 36
      6.3.3 Editing ProcessData ............................... 37
      6.3.4 Editing DeviceInfo ................................ 38
      6.3.5 Editing SystemInfo .............................. 40
   6.4 Information for manual editing .......................... 41

7. **Secured connections** ...................................... 42
   7.1 Encrypted connections .................................. 42
      7.1.1 Preparing certificates ............................ 42
   7.2 Authentication ......................................... 44
      7.2.1 Files to add user settings ....................... 44

8. **Appendix** .................................................. 45
   8.1 Third party licenses ................................... 45
      8.1.1 OPC UA Server ..................................... 45
      8.1.2 OPCUA Config Editor .............................. 45

9. **Imprint** ..................................................... 46
1 Introduction

OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation. It has already become the de facto standard for the automation market and Industry 4.0.

Advantages of OPC UA:

- Platform independence and scalability Ensures the seamless flow of information among devices from multiple vendors.
- Robust security Authentication and encryption capabilities for data protection.
- A sophisticated information model Defines any model from simple to complex.

Yaskawa’s OPC UA server is an optional function which extends the controller by the OPC UA communication protocol to get it ready for industry 4.0. The information model used implements the VDMA’s Robotics companion specification 1.0, which was created by the VDMA Robotics Initiative. This is a working group within VDMA robotics where Yaskawa was also actively involved alongside other robotics vendors.

Yaskawa’s OPC UA server provides the following functionalities and advantages:

- Server runs on the controller using its standard ethernet interface (no additional hardware is required).
- Implements basic OPC UA functionalities (Encryption, Authentication, Subscriptions, Methods,...).
- Structured view of the robot system.
- Read, write access to robot data (Variables, IO’s).
- Access to maintenance data.
- Information model can be expanded by custom data.
- Methods for remote control of the robot (Servo, JobStart, …).

For connecting the server Unified Automations UAExpert is used as the client throughout this document.
## Implementation details

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported controllers</td>
<td>DX200</td>
</tr>
<tr>
<td></td>
<td>YRC1000</td>
</tr>
<tr>
<td></td>
<td>YRC1000m</td>
</tr>
<tr>
<td>Max number of sessions</td>
<td>2</td>
</tr>
<tr>
<td>Max subscriptions per session</td>
<td>5</td>
</tr>
<tr>
<td>Max monitored items per session</td>
<td>600</td>
</tr>
<tr>
<td>Minimum Sampling Interval for process data/</td>
<td>250 / 1000 ms</td>
</tr>
<tr>
<td>other data</td>
<td></td>
</tr>
<tr>
<td>Supported security modes</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>Basic128Rsa15</td>
</tr>
<tr>
<td></td>
<td>Basic256</td>
</tr>
<tr>
<td>Protocol</td>
<td>Binary protocol</td>
</tr>
<tr>
<td></td>
<td>opc.tcp://&lt;IPServer&gt;:16448</td>
</tr>
<tr>
<td>Port</td>
<td>16448</td>
</tr>
<tr>
<td>Number of endpoints</td>
<td>1</td>
</tr>
<tr>
<td>Access to variables</td>
<td>B,I,R,D,S (r,w)</td>
</tr>
<tr>
<td>Access to IOs</td>
<td>All IOs (r)</td>
</tr>
<tr>
<td></td>
<td>General Outputs (w)</td>
</tr>
<tr>
<td></td>
<td>Network Inputs (w)</td>
</tr>
<tr>
<td>Companion spec</td>
<td>VDMA Robotics companion specification</td>
</tr>
<tr>
<td></td>
<td>version V1.0</td>
</tr>
</tbody>
</table>
3 OPC UA Server setup

3.1 Install OPC UA server

3.1.1 Requirements

- DX200 (Software >= DN2.14.01)
- YRC1000 or YRC1000 micro
- Activated MotoPlus Runtime
- Valid license
- FTP function
  - Standard setting (Encrypted FTP is not supported)
  - Additional parameters/settings
    - RS005=1
    - RS007=2
    - RS029=1
    - IO=>Pseudo Input Signal=>Command Remote=ON
  - If using password protection function the standard administrator password must not be changed
  - The standard password for maintenance mode must not be changed in order to read maintenance data correctly

The following system versions support any FTP setting
- YRC1000 with software >= 2.60.10. Set S2C1402 = 3
- YRC1000 micro with software >= 2.10.00. Set S2C1402 = 3
3.1.2  Load the OPCUA.out application file

Start the controller in Maintenance mode

- Holding Main Menu key during startup

Then set the security mode to Management mode.

Copy the OPCUA.out file to the root folder of a CompactFlash/SD card or USB stick and attach it to the Programming Pendant. Then follow installation procedure given below:

1. Select the main menu “MotoPlus APL.”.
2. Select the right device to load the application from

3. Select Load (User Application)
4. Move the cursor and press [SELECT]. Then, the selection indicator appears on the left of the OPCUA.out file.

5. Press [ENTER], then the following confirmation box appears. To load the file, select "YES". If the OPCUA.out file is already available to the controller the "Overwrite confirmation box" appears. Select YES to overwrite the existing file.

3.1.3 Server certificate

By default, the server creates his own certificate which is used for the connection with clients. The certificate is stored inside an inform job on the pendant (OPCCERT.JBI) or on a connected USB storage device (pki/own/). For configuration of the storage path see "CertStore Tag" in the configuration file.

A new certificate is automatically created when no certificate is found in the defined location.

**NOTICE**

Please note that controller time provides the start time of the certificate. In case the controller time is set to a wrong time a connection might not be possible (BadCertificateTimeInvalid). The certificate is valid for ten years.
3.2 Licensing OPC UA server

3.2.1 Checking for a valid license

If you already have installed the OPC UA Server, the license state can be checked using the following procedure:

1. Connect to the OPC UA server as shown in Connecting OPC UA server.
2. In the node tree browse to `Server.VendorServerInfo` and select the `LicenseState` node.
3. Depending on your license state, the value should be either ‘invalid’ or ‘valid’.
4. If your license is valid you can skip this chapter. Otherwise the license must be requested by Yaskawa and activated.
3.2.2 Request a license file from Yaskawa Europe

If a license file is already to hand, these steps can be skipped.

On the controller:

- Upgrade security level to Management.
- Touch [System Info] > [Network Service].
- Write down the MAC address displayed on the pendant for the network card.
  (In case of YRC1000 use the MAC of the first Network card)

- Send the MAC address along with the purchase information to software-licensing@yaskawa.eu.com.

You will receive the “MotoPlusApp.lic” file via email.
3.2.3 License activation

3.2.3.1 DX200/YRC1000

After obtaining a license file follow these steps to activate it on the controller:

- Power off the controller.
- Remove the CF/SD card from the YCP01 board inside the controller's cabinet.
- Place the license file “MotoPlusApp.lic” into the root folder of the CF/SD card.
- Put the CF/SD card back into the YCP01 board on the controller.
- Boot the controller into MAINTENANCE mode by holding down the MAIN MENU button on the pendant while powering on the controller.
- Upgrade the security level to Management.
- Touch [MotoPlus APL] > [License File] > [Load].
• Confirm the dialog.
• Perform CPU reset and boot into normal mode.
The OPC UA license is now activated.

3.2.3.2 YRC1000 micro
• Place the license file “MotoPlusApp.lic” onto the root of a usb or sd card.
• Insert the memory card into the teach pendant.
• Boot the controller into normal mode.
• Elevate the security mode to ‘MANAGEMENT’ (9’s).
• Select ‘EX. MEMORY’.
• Ensure the proper device is selected.
• Select ‘LOAD’, ‘USER FILE’.
• Select/load the license file “MotoPlusApp.lic” from the list.
4 Connecting OPC UA Server on port 16448

- Open UA Expert and right click on Server->Add.
- Double-click to add a new server.
- Enter the URL of the server. The Port is fixed at 16448. The IP address depends on the LAN settings of the controller.
- Navigate the available server endpoints. Depending on the configuration, secured or non-secured endpoints are available. For an easy start select None (no encryption).

- Server is added. If not already connected, right-click on the server to connect.

You are now connected to the Yaskawa OPC UA Server.
5 OPC UA Address space / Information model

The Address Space is based on VDMA’s Robotics Companion Specification. For further information please check this specification.

5.1 General overview

After successfully installing and activating your OPC UA Server, you will be able to browse the OPC UA Address Space by using UA Expert.

The root directory should look like this.

Robot data is located in DeviceSet.MotionDeviceSystem.

A MotionDeviceSystem consists of one controller and depending on configuration 1..n MotionDevices.

5.1.1 MotionDevices and controllers

Depending on the controller configuration, there can be more than one MotionDevice that stands for an independent motion device according to the VDMA Robotics spec. In our model it can be a robot or a single axis (turntable, …). In the case of controller, there is always only one available.
5.1.2 ProzessData

The process data configurable by the user can be found under

DeviceSet.Controllers.Controller_1.ProcessData

5.1.3 Methods

Depending on the configuration, a set of methods to control the robot remotely is available at

Controllers.Controller_1.Methods
5.1.3.1 ExportNodeset
Exports all nodes of the MotionDeviceSystem as a .csv file to a USB device connected to the internal USB connector.

On DX200 use CN106, left port
On YRC1000 use CN102
On YRC1000 micro use the USB port on the front

5.1.3.2 GetAvailableJobs
Provides a list of all available jobs on the controller.

If the hold function is activated by this method, it can’t be released from PP.
To release it, call this method again with the checkbox unchecked.

5.1.3.3 Hold
Turns ON/OFF the hold function.

5.1.3.4 ResetAlarm
Resets a robot alarm.
**5.1.3.5 **ResetError

Resets a robot error.

**5.1.3.6 **SetJob

Sets the job name and line number to the current job of the master task.

JobName has to exist on the Controller and JobLine must be an Integer between 0 and the job length.

**5.1.3.7 **SetServo

Turns ON/OFF of the servo power. Controller must be in remote state.

**5.1.3.8 **StartJob

Starts a job. Controller must be in remote state.

If a JobName is entered the execution will start at the first line of the job. Otherwise the execution of the current job is continued from the current position.
5.2 NodeSet

The following list provides an overview of the current used address space. Maintenance data, number of MotionDevices and Axes might differ depending on controller and robot setup.

There are two namespaces used:
- **NS4** is used for nodes coming from the OPC UA for Robotics Companion Specification
- **NS5** is used for nodes that are specific to Yaskawa robots.

<table>
<thead>
<tr>
<th>Type</th>
<th>Namespace</th>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem</td>
<td>Name of the Motion Device System. Can be overridden by OPCCONFIG_JBI</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.ComponentName</td>
<td>Name of the controller. Can be overridden by OPCCONFIG_JBI</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers</td>
<td>Contains the set of controllers in the motion device system. This is always 1.</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ComponentName</td>
<td>Name of the controller. Can be overridden by OPCCONFIG_JBI</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.CurrentUser</td>
<td>The current logged in user level</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Manufacturer</td>
<td>Controller vendor</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Model</td>
<td>Controller model</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet</td>
<td>The date and time of the last startup of the controller.</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.StartUpTime</td>
<td>The total accumulated time the controller was powered on.</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.TotalPowerOnTime</td>
<td>IP address(es) of the controller</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.IsRunning</td>
<td>True if the robot is running</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.IsPlayback</td>
<td>True if the robot is in playback mode</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.IsEnergySaving</td>
<td>True if the robot is in energy saving mode</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.MovingTime</td>
<td>Accumulated moving time of all devices</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.OperatingTime</td>
<td>Accumulated application specific operating time of all devices</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.PlaybackTime</td>
<td>Accumulated time of playback state of each devices</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ParameterSet.ServoOnTime</td>
<td>Total servo on time of all devices</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ProductCode</td>
<td>The product code of the device</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1.SerialNumber</td>
<td>The serial number of the device</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Software</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Software.Software_1.ComponentName</td>
<td>Software name</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Software.Software_1.Manufacturer</td>
<td>Software manufacturer</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Software.Software_1.ApplicationType</td>
<td>Type of robot application</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Software.Software_1.Language</td>
<td>Configured languages</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls</td>
<td>TaskControls is a container for one or more instances of TaskControlType</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ComponentName</td>
<td>Name of the task. Master</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ParameterSet</td>
<td>Flat list of parameters</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ParameterSet. ExecutionMode</td>
<td>Execution mode of the task control</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ParameterSet. TaskProgramLoaded</td>
<td>The TaskProgramLoaded variable is TRUE if a task program is loaded in the task control, FALSE otherwise</td>
</tr>
<tr>
<td>Variable</td>
<td>4</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ParameterSet. TaskProgramName</td>
<td>Task name (Master, Sub1...Sub15)</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ParameterSet. TaskProgramLine</td>
<td>The current line number of task</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. TaskControls.TaskControl_1...16.ParameterSet. TaskProgramStep</td>
<td>The current step number of task</td>
</tr>
<tr>
<td>Object</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Alarms</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Alarms.Alarm1...4.AxisData</td>
<td>Affected axes if available</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Alarms.Alarm1...4.ControlGroup</td>
<td>Affected control group if available</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Alarms.Alarm1...4.Date</td>
<td>Alarm date</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Alarms.Alarm1...4.JobLine</td>
<td>Job line number</td>
</tr>
<tr>
<td>Variable</td>
<td>5</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Alarms.Alarm1...4.JobName</td>
<td>Name of the job where the alarm occurred if available</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Alarms.Alarm1...4.StepNo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Alarms.Alarm1...4.TaskNo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Alarms.AlarmActive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Alarms.ErrorActive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>MotionDeviceSystem.Controllers.Controller_1.DeviceInfo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>MotionDeviceSystem.Controllers.Controller_1.ProcessData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Maintenance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>MotionDeviceSystem.Controllers.Controller_1.Maintenance.ServoRacks.ServoRack_1...n.Capacitors.Capacitor_1...n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>MotionDeviceSystem.Controllers.Controller_1. Methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Variable 5</td>
<td>Method 5</td>
<td>Sets servo input argument.</td>
<td></td>
</tr>
<tr>
<td>Variable 5</td>
<td>Method 5</td>
<td>Sets job and line number to the current job of the master task.</td>
<td></td>
</tr>
<tr>
<td>Variable 5</td>
<td>Method 5</td>
<td>This function executes the specified job from the beginning. If job-name is empty, the current suspended job is resumed.</td>
<td></td>
</tr>
<tr>
<td>Variable 5</td>
<td>Method 5</td>
<td>Activates/Resets the hold function.</td>
<td></td>
</tr>
<tr>
<td>Variable 5</td>
<td>Method 5</td>
<td>Hold input argument.</td>
<td></td>
</tr>
<tr>
<td>Method 5</td>
<td>Method 5</td>
<td>This function executes the specified job from the beginning. If job-name is empty, the current suspended job is resumed.</td>
<td></td>
</tr>
<tr>
<td>Method 5</td>
<td>Method 5</td>
<td>Sets the job and line number to the current job of the master task.</td>
<td></td>
</tr>
<tr>
<td>Method 5</td>
<td>Method 5</td>
<td>This function executes the specified job from the beginning. If job-name is empty, the current suspended job is resumed.</td>
<td></td>
</tr>
<tr>
<td>Method 5</td>
<td>Method 5</td>
<td>Hold input argument.</td>
<td></td>
</tr>
<tr>
<td>Method 5</td>
<td>Method 5</td>
<td>Returns all available robot jobs.</td>
<td></td>
</tr>
<tr>
<td>Method 5</td>
<td>Method 5</td>
<td>Get available jobs output argument.</td>
<td></td>
</tr>
<tr>
<td>Object 4</td>
<td>Object 4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n</td>
<td></td>
</tr>
<tr>
<td>Object 4</td>
<td>Object 4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.Axes</td>
<td></td>
</tr>
<tr>
<td>Object 4</td>
<td>Object 4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.Axes.Axis_1…n</td>
<td></td>
</tr>
<tr>
<td>Variable 4</td>
<td>Variable 4</td>
<td>Axis name. In case of robot (SLURBT)</td>
<td></td>
</tr>
<tr>
<td>Variable 4</td>
<td>Variable 4</td>
<td>The kind of axis motion, as defined with the AxisMotionProfileEnumeration.</td>
<td></td>
</tr>
<tr>
<td>Variable 4</td>
<td>Variable 4</td>
<td>The axis position</td>
<td></td>
</tr>
<tr>
<td>Variable 4</td>
<td>Variable 4</td>
<td>Degree [deg]</td>
<td></td>
</tr>
<tr>
<td>Variable 4</td>
<td>Variable 4</td>
<td>The axis speed on load side (after gear/spindle) inclusive Unit.</td>
<td></td>
</tr>
<tr>
<td>Variable 4</td>
<td>Variable 4</td>
<td>Degree per second [deg/s]</td>
<td></td>
</tr>
<tr>
<td>Object 5</td>
<td>Object 5</td>
<td>Maintenance parent node.</td>
<td></td>
</tr>
<tr>
<td>Object 5</td>
<td>Object 5</td>
<td>SpeedReducer parent node.</td>
<td></td>
</tr>
<tr>
<td>Variable 5</td>
<td>Variable 5</td>
<td>Design lifetime in hours</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.Axes.Axis_1…n.Maintenance.SpeedReduction.RemainingLifetime</td>
<td>Remaining lifetime in hours</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ComponentName</td>
<td>Custom name of the robot. Can be overridden by OPCCONFIG.JBI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.FlangeLoad.Mass</td>
<td>The weight of the load mounted on the flange</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.FlangeLoad.Mass.EngineeringUnits</td>
<td>Grams [g]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.Manufacturer</td>
<td>Device manufacturer</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.Model</td>
<td>Device model</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.MotionDeviceCategory</td>
<td>The variable MotionDeviceCategory provides the kind of motion device defined by MotionDeviceCategoryEnumeration based on ISO 8373.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ParameterSet.InControl</td>
<td>Servo on state</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ParameterSet.SpeedOverride</td>
<td>SpeedOverride provides the current speed setting in percent of programmed speed (0 - 100%).</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ParameterSet.IsMoving</td>
<td>Signals if the device is moving</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ParameterSet.MovingTime</td>
<td>Total moving time of the robot since initialization of the system</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ParameterSet.PlaybackTime</td>
<td>Total playback time of the robot since initialization of the system</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.ParameterSet.ServoOnTime</td>
<td>Total servo on time of the robot since initialization of the system</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.PowerTrains</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MotionDeviceSystem.MotionDevices.MotionDevice_1…n.PowerTrains.PowerTrain_1…n.Motors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### OPC UA Address space / Information model

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>4 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.PowerTrains.PowerTrain_1…n. Motors.Motor_1.Identifier. SerialNumber</td>
</tr>
<tr>
<td>Object</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.DeviceInfo</td>
</tr>
<tr>
<td>Variable</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.DeviceInfo.ControlGroupName</td>
</tr>
<tr>
<td>Object</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance</td>
</tr>
<tr>
<td>Object</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance. Inspection</td>
</tr>
<tr>
<td>Object</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance. Inspection. GreaseSupply</td>
</tr>
<tr>
<td>Object</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance. Inspection. WireHarnessExchange</td>
</tr>
<tr>
<td>Object</td>
<td>5 MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance. Inspection. BatteryExchange</td>
</tr>
</tbody>
</table>

- **Object 4**: Represents motion devices in a system, including power trains and motors.
- **Variable 4**: Represents parameters of the motor temperature, which can be encoded temperature or degree in [deg].
- **Variable 4**: Represents product codes of the motor and device.
- **Object 5**: Holds device-specific information, including control group names.
- **Object 5**: Includes maintenance and inspection activities for wire harness exchange and battery exchange.
- **Variable 5**: Represents design and remaining lifetimes for various maintenance activities.
<table>
<thead>
<tr>
<th>Object</th>
<th>5</th>
<th>MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance.Inspection. Overhaul</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable 5</td>
<td>MotionDeviceSystem.MotionDevices. MotionDevice_1…n.Maintenance.Inspection. Overhaul.RemainingLifetime</td>
<td>Remaining lifetime in hours</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.SafetyStates</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.SafetyStates. SafetyState_1</td>
</tr>
<tr>
<td>Variable 4</td>
<td>MotionDeviceSystem.SafetyStates. SafetyState_1_ComponentName</td>
<td>Safety state name</td>
</tr>
<tr>
<td>Object</td>
<td>4</td>
<td>MotionDeviceSystem.SafetyStates. SafetyState_1_ParameterSet</td>
</tr>
<tr>
<td>Variable 4</td>
<td>MotionDeviceSystem.SafetyStates. SafetyState_1_ParameterSet.EmergencyStop</td>
<td>The EmergencyStop variable is TRUE if one stop functions in the robot system are active, and FALSE otherwise.</td>
</tr>
<tr>
<td>Variable 4</td>
<td>MotionDeviceSystem.SafetyStates. SafetyState_1_ParameterSet.OperationalMode</td>
<td>The OperationalMode variable provides information about the current operational mode. Allowed values are described in OperationalModeEnumeration, see ISO 10218-1:2011 Ch.5.7 Operational Modes.</td>
</tr>
<tr>
<td>Variable 4</td>
<td>MotionDeviceSystem.SafetyStates. SafetyState_1_ParameterSet.ProtectiveStop</td>
<td>The ProtectiveStop variable is TRUE if one or more of the enabled protective stop functions in the system are active, FALSE otherwise.</td>
</tr>
<tr>
<td>Object</td>
<td>5</td>
<td>MotionDeviceSystem.SystemInfo</td>
</tr>
</tbody>
</table>
6  Configuration file OPCCONFIG.JBI

Configuration of the server is done by editing OPCCONFIG.JBI which contains information in JSON format. The name is fixed to OPCCONFIG.JBI. To edit you can use the OPCUA Config Editor or edit it manually, though this is not recommended.

6.1  Configuration default setting

If no OPCCONFIG.JBI file exists, or the present OPCCONFIG.JBI is corrupted or contains errors, these default settings are used to start the OPC UA Server.

Users
Anonymous:  1 // enable anonymous login
Custom:    0

Security
None:      1 // no encryption
Basic128Rsa15:  0
Basic256:  0
Basic256Sha256:  0

CertStore  0 // store certificate in job (CERT.JBI)

DisableMethods  0 // show available methods

Network
Port1:    1 // OPC UA server listens on port 1
Port2:    0

UTC:      0

ProcessData:    Empty

VendorSettings: Empty

6.2  Basic structure

/JOB
//NAME OPCCONFIG
//POS
///NPOS 0,0,0,0,0,0
///INST
///DATE 2019/01/10 09:31
///ATTR SC,RW
NOP
{"AppSettings": {
...
"},
"VendorSettings": {
...
"}
}
END

All tags discussed here must be separated by a comma, for clarification see Example OPCCONFIG.JBI.
6.2.1 **AppSettings**

6.2.1.1 **CertStore Tag**
Selects where to store the certificates for client and server
- 0: Stores certificate in the CERT.JBI file
  (Only available if security mode is set to none. In this case only the server’s certificate is stored. The certificates from clients are not re-quired).
- 1: Stores certificate on a USB connected to the USB connector
  (This mode is required if encrypted connections are used). Make sure to plug a USB storage device into the USB port:
  – On DX200 CN106, left port (inside cabinet)
  – On YRC1000 CN102 (inside cabinet)
  – On YRC micro use the USB port on the front

```json
"CertStore": "0",
```

6.2.1.2 **UserAccess Tag**
Selects user logon mode, available tags:
- Anonymous: enables anonymous login.
- Custom: enables custom user login (see Authentication).

```json
"UserAccess": {
  "Anonymous": "1",
  "Custom": "0"
}
```

6.2.1.3 **Security Tag**
Selects which encryption to use for connecting. Only one can be active at the same time. In case more than 1 is activated the first one is active. For setup see Encrypted connections.
- None Allows non-encrypted connections.
- Basic128Rsa15 Enables Basic128Rsa15 encryption for connections.
- Basic256 Enables Basic256 encryption for connections.
- Basic256Sha256 Enables Basic265Sha256 encryption for connections.

```json
"Security": {
  "None": "1",
  "Basic128Rsa15": "0",
  "Basic256": "0",
  "Basic256Sha256": "0"
}
```

6.2.1.4 Network Tag
Selects, which network interface to use. If none is selected, or Port 2 is disabled, it defaults to Port 1.
- Port1 Enables network interface 1.
- Port2 Enables network interface 2 (only YRC1000).

On DX200 Port1=CN104
On YRC1000 Port1=CN106 (LAN2)
Port2=CN107 (LAN3)

6.2.1.5 UTC Tag
Sets your current time zone in relation to UTC. It must be an Integer between -24 and +24. The displayed time on the controller side is always the local time. Since OPC UA timestamps are transferred in UTC you must provide your local UTC time shift. Keep in mind that you must manually adjust for daylight savings time.

Example:
You are located in Berlin, therefore your time shift according to UTC is ‘+1’:

..."UTC":"+1"
...

6.2.1.6 DisableMethods Tag
Selects whether to show Methods in
DeviceSet.MotionDeviceSystem.Controllers.Controller_1.Methods
The Value can either be 0 for showing Methods or 1 for hiding them. Note that the folder ...
Controller_1.Methods will exist either way.

..."DisableMethods":"0"
...
6.2.1.7 ProcessData Tag

Configures the robot variables and IOs which should be added to the “ProcessData” node of the controller. Variables can have read or write access.

Note that the available range of variables depends on your controller configuration:

- B000 ..xxx Variable(s) of type Byte
- I000 ..xxx Variable(s) of type Integer
- D000 ..xxx Variable(s) of type Double
- R000 ..xxx Variable(s) of type Real
- M000 ..xxx Register(s)
- S000 ..xxx Variable(s) of type String

For IOs use the IO numbers as on the robot side. This makes it possible to add the complete address space from 00010 to 87017.

Please note that only General Purpose Outputs and Network Inputs allow read/write access. All others are **read-only**.

- 00010 General Purpose Input 1
- 10010 General Purpose Output 1

By using the name tag a custom name can be assigned to the variable. This name as well as the name of the ProcessData Group cannot exceed 31 characters. If no name tag is specified the variable/IO name is used.

To import a group of variables/IOs at once replace the last digit by ‘x’.

- 0001x Imports General Input Group 1 (00010.. 00017)
- B00x Imports B000 to B009

Examples

```json
"Variables": {
  "B001": {
    "Name": "PartCounter",
    "Rights": [{
      "Access": "w"
    }],
  },
  "B002": {
    "Rights": [{
      "Access": "w"
    }],
  },
  "R001": {
    "Name": "PartLocationX",
    "Rights": [{
      "Access": "w"
    }],
  },
  "R002": {
    "Name": "PartLocationY",
    "Rights": [{
      "Access": "w"
    }]
  }
}
```

Import User variables B001, R001 and R002 with a custom name and B002 with its default name.

All variables are writeable.
Import Integer Group I00 to I09 with prefix ‘I_Vars_’ and R001 and R002 with a custom name.
All variables are read-only.

IOs 3 and 5 of group 1, as well as IOs 2 and 7 of group 3
The folder should be called ‘System’ and the I/Os named after the following list:

<50070> RUN (Start Lamp)
<50071> HOLD (Hold Lamp)
<50073> Servo On
<50074> I/O Simulated
<50075> Job Edit Inform
<50076> Jog Opn Inform
<50077> OT Release

...
6.2.2 VendorSettings

6.2.2.1 SystemInfo Tag

Allows adding of custom tags to the MotionDeviceSystem.SystemInfo folder. The tag is limited to 31 characters and the value is limited to a max of 127 characters.

If CSName is specified as tag, the MotionDeviceSystem.ComponentName is overwritten.

E.g.:

```json
...
"SystemInfo": {
  "CSName": "Test Cell",
  "Name": "Yaskawa",
  "Location": "Allershausen"
}
...
```

6.2.2.2 DeviceInfo Tag

Allows adding of custom tags to the MotionDeviceSystem.MotionDevices.MotionDevice_x.DeviceInfo folder.

The tag is limited to 31 characters and the value is limited to a max of 127 characters.

CSSerialNr overwrites MotionDeviceSystem.MotionDevices.MotionDevice_X.SerialNumber

CSModel overwrites MotionDeviceSystem.MotionDevices.MotionDevice_X.Model

CSProductCode overwrites MotionDeviceSystem.MotionDevices.MotionDevice_X.ProductCode

CSName overwrites MotionDeviceSystem.MotionDevices.MotionDevice_X.ComponentName

Example

```json
...
"DeviceInfo": {
  "R1": {
    "InstallDate": "05/10/2018",
    "Description": "YaskawaRobot",
    "ProductionDate": "05/10/2018",
  }
}
...
```

The Control group the device info is applied to is R1,

Install Date 10.05.2018,

Description is YaskawaRobot,

Production Date is 10.05.2018
6.2.2.3 Example OPCCONFIG.JBI

/JOB
//NAME OPCCONFIG
//POS
//NPOS 0,0,0,0,0,0
//INST
//DATE 2019/05/28 09:19
//@ATTR SC,RW
NOP

'{
  "AppSettings": {
    "CertStore": "1",
    "UserAccess": {
      "Anonymous": "1",
      "Custom": "0"},
    "Security": {
      "None": "0",
      "Basic128Rsa15": "1",
      "Basic256": "0",
      "Basic256Sha256": "0"},
    "Network": {
      "Port1": "1",
      "Port2": "0"},
    "UTC": "+1",
    "DisableMethods": "0",
    "ProcessData": {
      "Variables": {
        "B00X": {
          "Name": "B_Vars_",
          "Rights": [{
            "Access": "w"}]},
        "I00X": {
          "Name": "I_Vars_",
          "Rights": [{
            "Access": "r"}]},
        "D00X": {
          "Name": "D_Vars_",
          "Rights": [{
            "Access": "w"}]},
        "M00x": {
          "Name": "Register_",
          "Rights": [{
            "Access": "r"}]}}},
    "VendorSettings": {
      "SystemInfo": {
"Name": "Yaskawa",
"Location": "Allershausen",
"CSName": "Test Cell"},
"DeviceInfo": {
"R1": {
"SerialNr": "34343355",
"OrderNr": "234246634",
"InstallDate": "10.05.2018",
"CSPowerTrain1": {
"CSMotor1": {
"CSSerialNr": "K18365-371-1",
"CSModel": "ERAS-1000-06VXH-3"}
},
"R2": {
"SerialNr": "45454466",
"OrderNr": "234246634",
"InstallDate": "10.05.2018"}
}}
END
6.3 Introduction to the OPCUA Config Editor

The OPCUA Config Editor is a standalone PC application. It simplifies the generation of a configuration file for the OPC UA server. For any changes made in the app to take effect save the config and upload it to the controller.

Please register to our download portal (http://www.yaskawa.eu.com/en/service/robotics-software-download/) to get the latest version.

6.3.1 Requirements

.Net 4.5 Framework.

6.3.2 App settings

![App settings screenshot]

- **ConfigTimestamp**: Timestamp of last edit. Automatically updated when config is saved.
- **Security**: The used encryption method of the OPC UA server
- **Network**: The network interface
  - On DX200 Port1=CN104
  - On YRC1000 Port1=CN106 (LAN2).
  - Port2=CN107 (LAN3).
  - On YRC1000 micro ethernet port on the front
- **UTC**: UTC Offset of the controller time (see UTC Tag)
- **Certificate Storage**: Location where the certificates are stored on the controller (see CertStore Tag)
- **DisableMethods**: Displays or hides the methods in …Controllers.Controller_1.Methods
### 6.3.3 Editing ProcessData

By clicking „New…“ a dialog box is opened, where a group name for a new set of process data can be entered. Once a group is created it can be renamed by clicking “Rename…“ or deleted.

If a ProcessData group is selected from the drop-down menu, a variable or IO can be added by selecting the corresponding checkbox and typing a number into the textbox. By typing ‘x’ for the last digit, a group of variables can be imported.

To add I010 to I019 select the ‘I’ checkbox, type ‘1x’ and click add. By selecting a ProcessData item from the list the values can be modified or deleted. The naming of the IOs is the same as displayed on the teach pendant 10010….87017.

On OPC Server side the groups and variables are displayed in the ProcessData node of the Controller.

*MotionDeviceSystem.Controllers.*

*Controller_1.ProcessData*
6.3.4 Editing DeviceInfo

Add custom data to the DeviceInfo node of each device and controller.

To create a device section click on „New…“ and select the device from the drop-down box.

To add a custom tag enter the tag name and value and click “Add”.
By selecting a tag from the list, the values can be modified or deleted.

On the OPC Server side the added values are displayed in the DeviceInfo node of the selected device or controller.
This section makes it possible to overwrite some existing tags of the address space. These tags start with ‘CS’ and are displayed by clicking the Defaults button.

E.g. CSName overrides the ComponentName of the device.

Changes made to

\texttt{R1.CSPowerTrain1}

are displayed in

\texttt{MotionDeviceSytem.MotionDevice_1.PowerTrains.PowerTrain_1.Motor_1}
6.3.5 Editing SystemInfo

Add custom data to the SystemInfo node of the MotionDeviceSystem.

To add a custom tag enter the tag name and value and click “Add”. By selecting a tag from the list the values can be modified or deleted.

All changes in the tab SystemInfo will be visible in `MotionDeviceSystem.SystemInfo`. If tag CSName is defined, it will over-write `MotionDeviceSystem.ComponentName`.
6.4 Information for manual editing

To edit the OPCCONFIG.JBI file manually, follow these steps:

- If you already have an OPCCONFIG.JBI, export it from the controller to a USB drive
- Edit the OPCCONFIG.JBI according to this manual
- Save the OPCCONFIG.JBI to any USB drive
- Plug the USB drive into the pendant of the controller
- Import OPCCONFIG.JBI as you would with any other job file
- For the changes to take effect reboot the controller

Note that JBI files must comply with the following:

- Each line cannot exceed 33 characters
- A line must contain some characters
- All lines between “NOP” and “END” must begin with an apostrophe
- May not contain any vowels with diacritics (e.g. “ü”).

Please note:

There is no guarantee your config will be loaded correctly by the server or OPCUA Config Editor, if edited manually.
7 Secured connections

If an endpoint with encryption is being used, additional setup is required to establish a connection.

7.1 Encrypted connections

7.1.1 Preparing certificates

When connecting for the first time, certificates are exchanged between the client and the server. In UA Expert to trust a server you must click ‘Trust Server Certificate’, then ‘Continue’.

![Certificate Validation](image)
Secured connections

After the first connection from the client side, the client's certificate is automatically stored in the rejected certificates path. To grant the client access to the server the certificate must be manually moved to the trusted location. Please follow the steps below.

1. Power on the controller.
2. Connect to the server in UA Expert. Your attempt will not succeed. This is normal.
3. Shutdown controller.
4. Remove USB drive and move the client's certificate from `/pki/rejected/` to `/pki/trusted/certs/`
5. Plug the USB drive back into the controller and restart.
6. Try connecting again. Now you should be able to log in to the server.
7.2 Authentication

7.2.1 Files to add user settings

In order to specify custom usernames and passwords, two additional files need to be edited in the root directory of the USB drive: group and passwd. To do so:

- Make sure the controller is turned off.
- Remove the USB drive from controller.
- Plug the USB drive into the computer.
- Right click the file you want to edit>Open.
- Select Editor and edit the file according to this manual.
- When finished click File>Save.
- Plug the USB drive back into controller and start.

7.2.1.1 group

Defines groups and which users are part of this group according to the scheme:

\(<\text{GroupId} \> \quad \text{<GroupName> <all UserNames belonging to the Group separated by commas}>\)

**NOTICE**

Double-check for unnecessary white spaces; UserNames are only separated by a comma, if a UserName contains a white space use quotation marks.

GroupId is numbered consecutively, starting with 0, GroupName and UserNames are arbitrary.

**Example:**

User root is part of group root, users John and Jane are part of operators, Ron Roe is in users:

0 root root
1 operators John,Jane
2 users "Ron Roe"

7.2.1.2 passwd

Defines passwords for the users mentioned in the group file according to the scheme:

\(<\text{UserID} \> \quad <\text{GroupId} \> \quad \text{<UserName> <Password encrypted in SHA1>}>\)

UserID is numbered consecutively, starting with 1, UserName has to match one UserName from group file, if it contains a white space use quotation marks. To obtain the SHA1 encrypt of a password use any online tool, for this example all passwords are "secret", except for Ron Roe, which is “terces”.

**Example:**

1 0 root e5e9fa1ba31ecdlae84f75caaa474f3a663f05f4
2 1 John e5e9fa1ba31ecdlae84f75caaa474f3a663f05f4
3 1 Jane e5e9fa1ba31ecdlae84f75caaa474f3a663f05f4
4 2 "Ron Roe" 4db00e8b8e3445bb6c0bc7b6d4ca5608c20b563b
8 Appendix

8.1 Third party licenses

8.1.1 OPC UA Server

LICENSE STATEMENT
This software includes the following third party software licensed under the MIT license cJSON

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8.1.2 OPCUA Config Editor

LICENSE STATEMENT
This software includes the following third party software licensed under the MIT license Json.NET

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
9 Imprint

Yaskawa Europe GmbH
Yaskawastraße 1
85391 Allershausen
Germany
Phone 00498166900
Fax 0049816690103
YASKAWA Headquarters
YASKAWA Europe GmbH
Robotics Division
Yaskawasraße 1
D-85391 Ailerbueran
Tel. +49 (0) 8165/90-0
Fax +49 (0) 8165/90-103

YASKAWA ACADEMY
and sales office Frankfurt
YASKAWA Europe GmbH
Robotics Division
Hauptstraße 185
D-65760 Eschborn
Tel. +49 (0) 6196/77725-0
Fax +49 (0) 6196/77725-39

YASKAWA GROUP

A  YASKAWA Austria
    Swatch/Inn
    +43(0)1-707-9324-15

CZ  YASKAWA Czech s.r.o.
    Prudnik u Prahy
    +420-257-941-718

E  YASKAWA Ibérica, S.L.
    Ava/Barcelona
    +34-93-5303478

F  YASKAWA France SARL
    Saint-Grignan-da-Grand-Lieu
    +33-2-49151919

FIN YASKAWA Finland Oy
    Turku +358 (0) 403003000

GB YASKAWA UK Ltd.
    Bambury +44-1295-272765

I  YASKAWA Italia s.r.l.
    Torino +39-011-906833

IL YASKAWA Europe Technology Ltd.
    Raan Ha2yin +972-3-9004114

NL YASKAWA Benelux B.V.
    Son +31-40-2868200

PL YASKAWA Polska Sp.z o.o.
    Warsaw +48-22-7528070

RUS YASKAWA Nordic AB
    Moskva +46-460-417-800

SE YASKAWA Nordic AB
    Torsås +46-460-417-800

SI YASKAWA Slovenia
    Dribnik +386-1-8372-410

TR YASKAWA Turkey Bahrik
    Ticaret Ltd. St.
    İstanbul +90-216-5273460

ZA YASKAWA Southern Africa (pty) Ltd
    Johannesburg +27-11-6083182

DISTRIBUTORS

BG AFAMET ROBOTICS Ltd.
    Yambol +359-886-317-294

CH Aervox Bulach Ltd.
    Bulach +41-44-847-17-17

DK Robotcenter Danmark
    Leisning +45-7022-2477

EE RHR Steadmad OÜ
    Tallinn/Estonia +372-62-35-235

GR Gkarios Robotics
    Nea Kifisia +30-210621455

H Flaxman Robotics Kft
    Budapest +36-30-9610065

LT Profibus UAB
    Panevėžys +370-45-518575

N Skala Robotech AS
    Lierstranda +47-32240600

PT ROBOPLAN Ltda
    Aveiro +351-234-943-900

RO Sam Robotics srl
    Timișoara +40-730-279-860

RO MPL Automation S.R.L.
    Satu Mare +40 (0) 261 750 741

robotics@yaskawa.com
www.yaskawa.com