Modbus-OPC UA Wrapper using Node-RED and IoT-2040 with application in the water industry.

Silviu-Iulian Toc, Adrian Korodi
University Politehnica Timisoara/Department of Automation and Applied Informatics, Timisoara, Romania
e-mail: adrian.korodi@upt.ro

Abstract—**In the context of connecting the physical and the digital worlds, common grounds must be established in several areas. Industry 4.0, Industrial Internet of Things (IIoT) principles are focused on interoperability. Regarding the interfacing, the OPC UA (OPC Unified Architecture) protocol is considered to be the most important. Descending from the PC based supervisory levels to the automation level (automation panels, machine and sensorial levels), achieving OPC UA interfacing is harder. Legacy systems, specific protocols, the need of lower cost hardware/software middleware equipment for existing systems that supports the industrial environment, flexibility to augment the existing structure with additional software modules, to develop easy knowledge transfer for integrators with lower software skills, are representing current issues on the automation level. Considering that a large part of the existing automation structures is using Modbus protocol, especially serial Modbus, the paper proposes a Modbus-OPC UA wrapper, considering a middleware IoT 2040 hardware equipment and Node-RED software environment, in order to provide interoperability. The work focuses on the serial Modbus client interface, but it can be extended within the same proposed structure to other protocols. The concept is tested first in the laboratory and then in a real scenario consisting of an existing wastewater pumping station (WWPS).**

I. INTRODUCTION

The industry is more and more concerned with connecting its physical part with the digital infrastructure, respectively to provide interoperability to the entities. Besides the physical communication support provided by different types of equipment, a very important aspect regarding interoperability is related to the interfacing. The main goals of the interoperability are data integration and interoperation between entities. Following Industry 4.0 principles and corresponding studies, the research and industrial community were setting up a common protocol, OPC UA (e.g. [1], [2], [3], [4]). As known, the OPC UA functions on client-server basis, it is platform independent, includes security modes and policies, allows easy addressing (provides an address space on the server side containing browsable nodes), contains all classical OPC features (DA - Data Access, A&E - Alarms and Events, HDA - Historical Data Access), etc.

Part of the research studies related to OPC UA are focusing on implementing OPC UA servers and clients associated for different hardware-software equipment. Some the studies refer to OPC UA severs developed for various process structures (e.g. on the sensorial level in [5], [6], various OPC UA server developments for the process parts in [7], [8]), and some are considering various issues/applications related already functional OPC UA servers (e.g. redundancy in [9], web-based platform for OPC UA in [10], etc.). To provide access to OPC UA servers from mobile devices, an Android based OPC UA client is presented in [11]. The industry is also active in developing embedded OPC UA solutions for PLCs (e.g. Beckhoff, Siemens), HMI panels (e.g. Siemens, Schneider Electric), Gateways (e.g. Softing, Matrikon), SCADA (e.g. Inductive Automation, Schneider Electric, Siemens).

The automation equipment and solutions are usually functioning for long time and therefore many legacy systems exist. The invasive interference in the functioning of local legacy systems is avoided (see [12]). The industry is preoccupied to make these systems interoperable and to integrate them on higher level supervisory applications. But, the industry is also concerned with the interoperation between entities (meaning also horizontal communication and interfacing), respectively in obtaining higher flexibility and adaptability of the owned solutions. This means that locally implemented OPC UA servers (e.g. integrated in the local automation panel) are presenting interest compared to centralizing OPC solutions located on a higher-level PC. Therefore, to combine the requirements of interfacing non-invasively with local automation, to convert legacy systems protocols, and to implement a local OPC UA server suitable for horizontal and vertical interoperability, a middleware solution is needed.

OPC UA based middleware solutions for the industry are researched in the literature (e.g. [13], [14], [15]) to provide interoperability for the local equipment. As generally known, serial Modbus was one of the most widespread protocol used in the past in the industry. Even now, when tendering documents are realized by not so well-prepared designers (e.g. with more electrical background than software and automation), the serial Modbus is implemented on new systems.

The water industry is no exception, many legacy systems, as wastewater pumping stations (WWPS), water wells, pumping stations (PS), chlorine stations, etc., are having their automation governed by a PLC or microcontroller with serial Modbus, or other larger plants, as water or wastewater treatment plants (WTP, WWTP) are using Modbus as internal protocol.

Having the experience in [15], of developing an OPC UA based middleware structure based on Raspberry Pi as hardware and Node-red as software environment, applied as a wrapper structure for Modbus TCP conversion, the current paper proposes a serial Modbus - OPC UA wrapper solution. The structure will consider the IoT-2040 as hardware and Node-red as software environment and the results are applied in the water industry.

The paper presents in the second chapter the functional overview of the solution. The third chapter details the developed application testing, first in the laboratory environment using an Arduino board with implemented Modbus protocol as a center of a small process, and then in a real scenario, represented by a wastewater pumping station controlled through a Wago PLC.

II. Functional Overview of the Solution

The hardware components of wrapper are the IoT-2040 and the RUT240 router. The characteristics of the IoT-2040 (1 GB DDR3, Intel Quark X1020 processor, micro SD card, 2 RJ45 ports – Ethernet, 2 COM ports - RS 232, RS 485, USB, 24VDC supply voltage, Industrial enclosure, etc.) are providing high-processing capabilities, physical interfaces to connect with local automation and to integrate it in local panels, and industry orientation.

The chosen software environment is Node-RED, with its advantages pointed out in [15]. The Node-RED offers the possibility to future extend the interfacing capabilities with other nodes and offers high programming flexibility in the backend, while the flow/node -based frontend offers the possibility to easily transfer the knowledge to other integrators, not highly skilled in software development.

From a functional point of view, the Modbus-OPC UA wrapper has to interface with the local automation processing units (PLCs/microcontrollers), to take-over and process data, to expose the processed data in the OPC UA server providing also distance communication support through the router, respectively to respond to external OPC UA clients' requests and to deliver resulting data back to the Modbus slave (if control actions are allowed).

The OPC UA server is configured with folder and node setup, port, users, etc. Important aspects that could cause different communication issues with the OPC UA client are related to certificate generation and exchange, hostname setup, correct date-time synchronization, security modes and policies.

Having the OPC UA server ready for subscriptions, the functional overview of the application is shown in Fig. 1. The serial Modbus client takes over and sends data to the Data filtering module that analyzes, splits and associates values to local variables. The Data processing module manipulates the already defined variables, implements diagnosis and protection structures, and augments the local logic with new algorithms. The Data structuring module sets up the format of the data as desired to be represented in the OPC UA server (e.g. groups bitwise information into words and sets up arrays in order to control smaller licenses in higher-level SCADA, etc.). The Data sampling control module sets up specific timeframes for OPC UA server injection for each tag. This way, the structure may reduce useless consumption of bandwidth (e.g. the total energy tag or the functioning hours of a motor in a process may not be needed to be noticed by the operator as fast as a motor start/stop status and it usually has a large value). Finally, the Node value injection module, changes the value of the specified node in the OPC UA server.

When the external OPC UA client establishes a control action on a process tag, the Node value change module identifies the action and further, the Modbus address-value structuring module structures the received information and associates the node to a Modbus address. The Modbus value injection module sends data to Modbus client that will action on the external Modbus slave.

Two monitoring and safety control modules are implemented for OPC UA server and for the Modbus client. The modules are monitoring the status of the applications and safety control functions are accomplished, the most impacting ones being restarting the OPC UA server, the Modbus client, or the entire application.

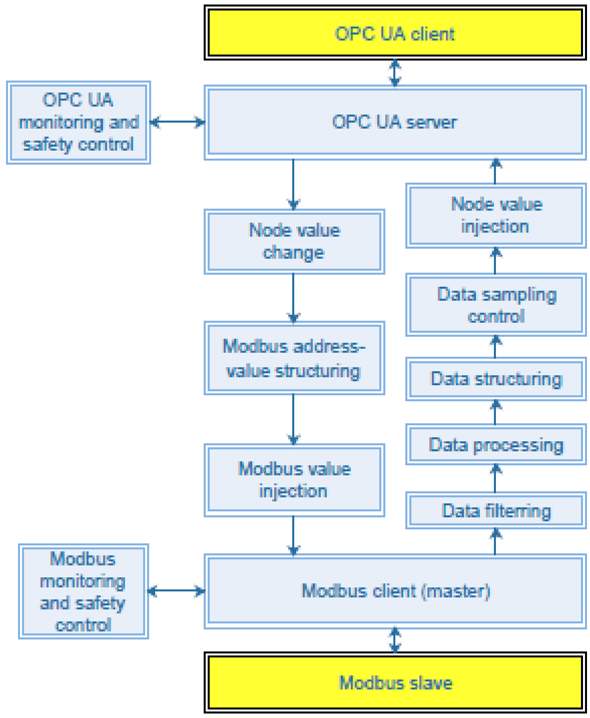

Figure 1. Functional overview of the application

III. Testing the Proposed Solution

The Modbus-OPC UA Wrapper based on IoT 2040 and Node-RED was tested first in a laboratory environment using an Arduino based application as process, and then, in a real scenario inside a functional WWPS.

A. Laboratory Test Application using Arduino

The laboratory environment was built around an Arduino Uno central processing unit. A simple local process was defined consisting mainly of three LEDs controlled by three internal variables, and some internal functions measuring functioning hours and number of starts. The local variables representing the LEDs and the internal switches statuses were included in the local serial Modbus slave structure. The serial Modbus protocol was tested for both RS232 and RS485 physical layers. As depicted in Fig. 2, where Modbus RTU was used, pins 0 and 1 on the Arduino board were user for the serial wiring with the IoT 2040, together with the MAX485 converter. The Ethernet connection was implemented between the IoT 2040 and the RTU 240 router, and the router was configured as in a real scenario, keeping for testing both 4G/LTE and WiFi communications.

The Modbus client was developed in Node-RED and the OPC UA Server was defined, containing the Modbus variables. The additional functions (e.g. different time ranges for populating the OPC UA server, entire structure automatic redeployment in case of failure, protection structures, local Modbus communication status check, OPC UA server status check, router configuration for access control, etc.) were also tested.

For the laboratory application, the OPC Client application was implemented also using Node-RED. First, the application was deployed on a Windows 10 environment, on a PC. The second OPC UA Client was deployed on a Samsung Phone with Android 7. The Node-RED Dashboard node was used to implement an easy graphical user interface (GUI) in order to command and monitor the state of the three LEDs.

The presented status from Fig. 3 depicts the GUI from the OPC UA Client application, where the second and the third switch actioning from the interface determines the activation of LED 2 and 3 (the state of the process is observable also in Fig. 2).

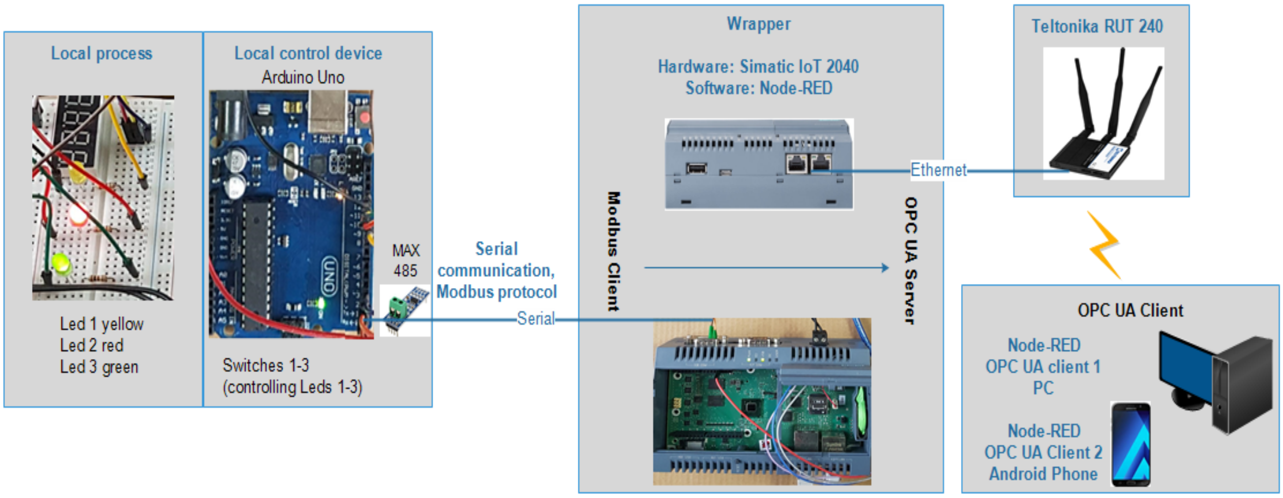

Figure 2. Schematic view of the implemented Modbus-OPC UA wrapper for the Arduino laboratory test application

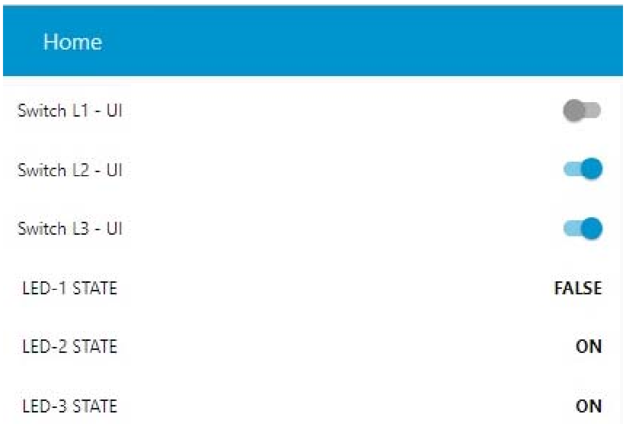

Figure 3. Node-RED Dashboard in OPC UA Client application for Windows (PC) and Android (Phone)

B. Real Scenario Application for a WWPS

The real process is represented by a WWPS with four pumps. The WWPS is cascaded with other WWPSs in order to cover the sewage network wastewater transport over a local area towards the WWTP. The local automation of the WWPS is minimal, the pumps functioning according to a level switch feedback. All four pumps have state values and general fault signals and a general anti-burglary signal is provided. Electrical parameters are measured locally for the whole WWPS: voltages, currents, powers, and total energy. The first control level is represented by a Wilo controller and the PLC of the station is a Wago 750-816. The 750-816 PLC is suitable for Modbus protocol. The local automation panel is illustrated in Fig. 4, and Fig. 5 presents the IoT-2040 and the Teltonika RUT240 router inside the WWPS in connection with the Wago PLC.

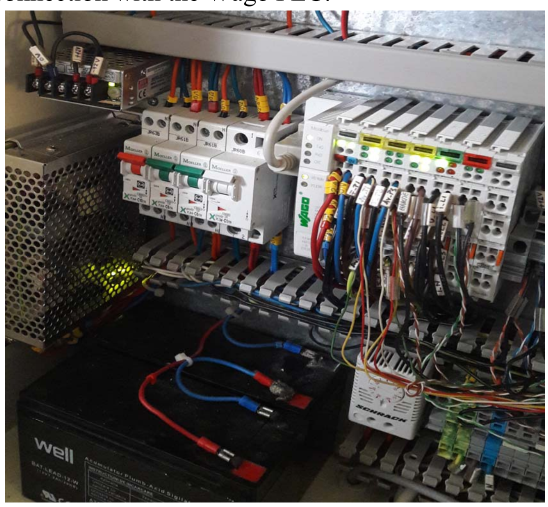

Figure 4. A view inside the local automation panel

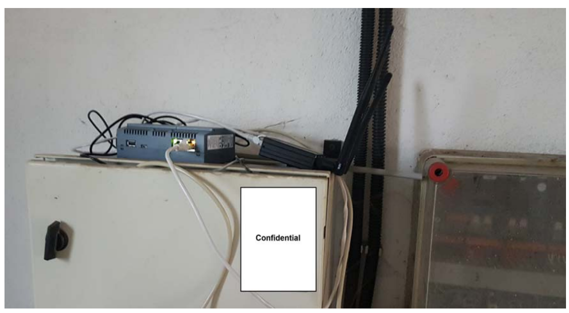

Figure 5. The wrapper inside the WWPS

The main components of the general architecture are extracted in Fig. 6. The developed wrapper is introduced in the functional system in a completely non-invasive manner towards the existing local process. The IoT 2040 is connected to the Wago PLC using a serial connection (RS 232) and the used protocol is Modbus (see Fig. 6). The connection between the IoT 2040 and the RUT 240 router is through Ethernet. The exposed port is defined in the OPC UA Server node.

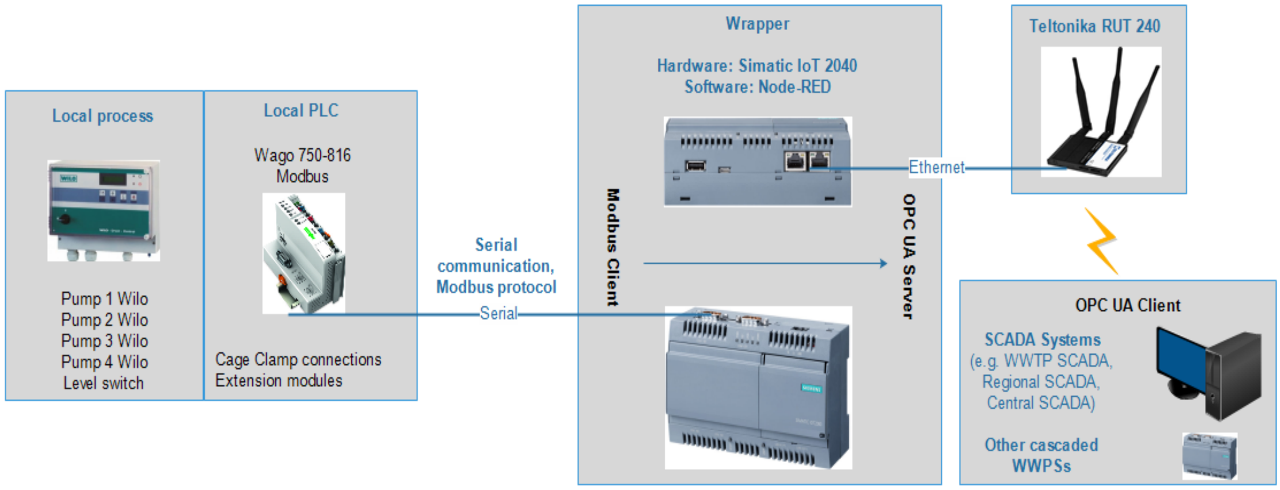

Figure 6. Schematic view of the implemented Modbus-OPC UA wrapper inside the WWPS automation

The results obtained using the initial UA client application from Softing are illustrated in fig. 7, augmented with English explanations, where the main tags (the digital ones) are browsed inside the local OPC UA server address space.

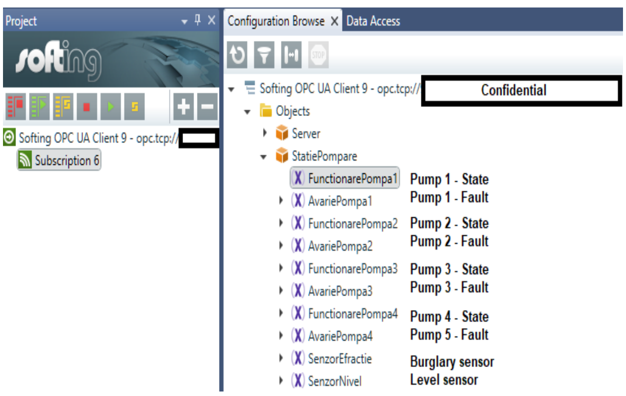

Figure 7. Softing OPC UA client connected to the wrapper

The Softing OPC UA client presents many setting options (e.g. connection response waiting times, session name, etc.) and it is very flexible regarding inconsistencies (e.g. hostname issues) comparing with some other OPC UA clients with lower configuration options, as IGSS. In IGSS, as in many SCADA environments, lots of things are hardcoded for easier usage and increased robustness. Therefore, the IGSS SCADA was needed for testing to prove the complete efficiency of the solution and to provide an environment where the operators can easily check the functioning system using synoptic schemes, alarms and graphics. The main OPC UA client application implemented in the SCADA control room consisted of one main diagram with the WWPS placed on a local map, pointing out the states/fault of the four pumps and rest of the numerical information that was available in the local device. All the tags were tested, including the faults integrated in the IGSS Alarms and Events module. Two diagram embedded graphs were realized and mapped to the digital atoms (see Fig. 8), and one independent Graph object with detailed graphical information. As pointed out in the two diagram embedded graphs from Fig. 8, pumps 3 and 4 started four times between 9:00-13:00 on the 18th of May 2018. The first two pumps are not starting in the rotation algorithm because both are in fault states. During testing, the burglary and the level overflow alarms were induced to analyze the notification times at the problem occurrence/ending.

Figure 8. Augmented screenshot from the IGSS application

Also, during testing, internal software faults were induced and analyzed and the system responded as expected. Inducing the highest magnitude fault that causes to reinitialize and restart the whole application, a complete automatic restart of Node-RED a maximum time of under 5 minutes was noticed until a full recovery of the wrapper.

After testing the entire structure, including the IGSS application used by the operators of the water distribution company, the results are proving that the implemented structure is having a high availability.

IV. CONCLUSIONS

The paper presented a serial Modbus to OPC UA wrapper based on IoT-2040 and RUT240 hardware equipment and Node-RED software environment. The solution was designed as an industry-oriented middleware structure placed on the local level (e.g. automation panel), interfacing with the local PLC/microcontroller, processing the information and exposing the desired data within a local OPC UA server for vertical and/or horizontal integration/interoperation.

The Modbus-OPC UA wrapper was implemented considering fast integration in the industry (industry readiness, easy knowledge-transfer, clear flows, various types of physical interfaces, low-cost solution), with consistent protection/diagnosis modules to confer high availability and robustness.

The developments were tested in the laboratory testbed representing a small process controlled by an Arduino Uno microcontroller, respectively in a real scenario in the water industry. The wrapper was applied in a non-invasive manner inside a functioning wastewater pumping station. The results are showing that the solution is presenting efficiency and increased availability.

The research on the obtained wrapper will be continued with a locally integrated honeypot structure concept that will emulate the functioning of water industry processes, exposing tags as in the real system and monitoring the activities of eventual attackers.

ACKNOWLEDGMENT

This work was supported by UEFISCDI, inside the Bridge Grant programme, project number PN-III-P2-2.1-BG-2016-0208, contract number 40BG/30.09.2016.

REFERENCES

[1] P. Drahos, E. Kucera, O. Haffner, I. Klimo, “Trends in industrial communication and OPC UA”, *Proceedings of the International Conference on Cybernetics & Informatics (K&I)*, pp. 1-5, Lazy pod Makytou, Slovakia, 28 Ian.-03 Feb., 2018

[2] M. Graube, S. Hensel, C. Iatrou, L. Urbas , "Information models in OPC UA and their advantages and disadvantages", *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 12-15 Sept. 2017

[3] H. Haskamp, M. Meyer, R. Mollmann, F. Orth, A. Colombo, “Benchmarking of existing OPC UA implementations for Industrie 4.0-compliant digitalization solutions”, *Proc. of the 15th IEEE Intern. Conf. on Ind. Info. (INDIN)*, Emden, Germany, pp. 589-594, 24-26 July, 2017

[4] S. Cavalieri, A. Regalbuto, “Integration of IEC 61850 SCL and OPC UA to improve interoperability in Smart Grid environment”, *Computer Standards & Interfaces*, vol. 47, pp. 77-99, Aug. 2016

[5] M. Muller, E. Wings, L. Bergmann, “Developing Open Source Cyber-Physical Systems for Service-Oriented Architectures Using OPC UA”, *Proc. of the 15th IEEE Intern. Conf. on Ind. Info. (INDIN)*, Emden, Germany, pp. 83-88, 24-26 July, 2017

[6] A. Veichtlbauer, M. Ortmayer, T. Heistracher, “OPC UA Integration for Field Devices”, *Proc. of the 15th IEEE Intern. Conf. on Ind. Info. (INDIN)*, Emden, Germany, pp. 419-424, 24-26 July, 2017

[7] I. Seilonen, T. Tuovinen, et.al., “Aggregating OPC UA servers for monitoring manufacturing systems and mobile work machines”, *Proc. of IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016

[8] H. Derhamy, J. Ronnholm, J. Delsing, J. Eliasson, J. van Deventer, “Protocol interoperability of OPC UA in Service Oriented Architectures”, *Proc. of the 15th IEEE Intern. Conf. on Ind. Info. (INDIN)*, Emden, Germany, pp. 44-50, 24-26 July, 2017

[9] A. Ismail, W. Kastner, "Throttled service calls in OPC UA", *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pp. 1658 - 1663, Lyon France, Feb. 2018

[10] S. Cavalieri, D. Di Stefano, M.G. Salafia, M.S. Scroppo, "A web-based platform for OPC UA integration in IIoT environment", *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 12-15 Sept. 2017

[11] A.M. Mateoiu, A. Korodi, “OPC-UA based small-scale monitoring and control solution for Android devices. Case study for water treatment plants”, *Proceedings of the 4th International Conference on Control, Automation and Robotics (ICCAR 2018)*, Auckland, New Zealand, 20-23 April, 2018

[12] R. Crisan, A. Korodi, "Noninvasive control solution for energy efficiency in wastewater treatment plants", *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pp. 1604 - 1609, Lyon France, Feb. 2018

[13] M. Zarte, A. Pechmann, J. Wermann, F. Gosewehr, A.W. Colombo, "Building an Industry 4.0-compliant lab environment to demonstrate connectivity between shop floor and IT levels of an enterprise", *Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON 2016)*, Florence, Italy, 23-26 Oct. 2016

[14] A. Balador, N. Ericsson, Z. Bakhshi, "Communication middleware technologies for industrial distributed control systems: A literature review", *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 12-15 Sept. 2017

[15] A. Korodi, I. Silea, “Achieving Interoperability Using Low-Cost Middleware OPC UA Wrapping Structure. Case Study in the Water Industry.”, Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN), pp. 1223-1228, Emden, Germany, 24-26 July, 2017