PLC - VIRTUAL INSTRUMENT INTERACTION USING THE OPC UA STANDARD FOR IIoT APPLICATIONS IN INDUSTRY AND REMOTE EDUCATION

INTERACCIÓN PLC - INSTRUMENTO VIRTUAL USANDO EL ESTÁNDAR OPC UA PARA APLICACIONES DE IIoT EN LA INDUSTRIA Y EN LA EDUCACIÓN REMOTA

Luis Gerardo Carvajal Fernández
Universidad de las Américas Puebla, México
luis.carvajalfz@udlap.mx

José Luis Vázquez González
Universidad de las Américas Puebla, México
josel.vazquez@udlap.mx

Reception: 25/october/2021 **Acceptance**: 19/diciembre/2021

Abstract

Industrial Internet of Things (IIoT) offers connectivity and information access for industrial devices. However, several devices cannot implement it because of being TCP/IP-incapable or having incompatible communication protocols. The OPC UA standard can be a solution by offering a platform-independent client-server communication model. The goal of the current work is to use OPC UA to connect a Programmable Logic Controller (PLC) in S7-PLCSIM and a Virtual Instrument (VI) in LabVIEW running as a Web Services application to enable applications in industry and remote education. The case study is a SCADA system simulated in FluidSIM. The results show a successful connection and demonstrate that the OPC UA standard can be used for implementing IIoT by connecting a simulated PLC to a VI for bidirectional information exchange in monitoring and control tasks. The former enables students to learn interactively and remotely about PLCs, and industries to improve their process control.

Keywords: Industrial Internet of Things (IIoT), OPC UA, Programmable Logic Controller (PLC), Virtual Instrument (VI).

Resumen

El Internet Industrial de las Cosas (IIoT) ofrece conectividad y acceso a información para dispositivos industriales. Sin embargo, muchos dispositivos no pueden implementarlo por falta de capacidades TCP/IP o por protocolos de comunicación incompatibles. El estándar OPC UA puede ser una solución por su modelo de comunicación cliente-servidor independiente de plataforma. El objetivo del trabajo es usar OPC UA para conectar un Controlador Lógico Programable (PLC) en S7-PLCSIM y un Instrumento virtual (VI) en LabVIEW corriendo como una aplicación de Servicios Web para habilitar aplicaciones industriales y en educación remota. Se estudia un sistema SCADA simulado en FluidSIM. Los resultados muestran una conexión exitosa, demostrando que OPC UA puede usarse para implementar IIoT conectando un PLC simulado a un VI para intercambio bidireccional de información en tareas de monitoreo y control. Esto permite a estudiantes aprender interactiva y remotamente sobre PLCs, y a industrias mejorar su control de procesos.

Palabras Claves: *Controlador Lógico Programable (PLC), Instrumento Virtual (VI), Internet Industrial de las Cosas (IIoT), OPC UA.*

1. Introduction

The world is experiencing a transition towards the Fourth Industrial Revolution, also known as Industry 4.0. As technologies evolve, the global panorama shows a clear tendency of a digital transformation for society. The Internet is the most evident demonstration of this, and it has served in the last decades to close the gap between devices. Its impact is undeniable: it provides near-instantaneous connectivity and data exchange worldwide. Now, one of the leading trends of Industry 4.0 promises to use the Internet as media to communicate remotely with industrial machinery. That trend is the Industrial Internet of Things (IIoT), the industrial version of the Internet of Things (IoT). It allows sensors, controllers, and even entire machines in production processes to share data via the Internet.

According to a report by [MarketsandMarkets, 2019], Industrial Internet adoption is recognized as one of the key elements for market growth. The *Industry 4.0* market had an estimated value of USD 71.7 billion in 2019, with expected growth to USD 156.6 billion by 2024. However, there is a challenge that limits the scope of IIoT in industry: vendors have traditionally developed communication protocols specific to their products. Consequently, it becomes a challenge to configure communication between devices from different vendors. Moreover, many of them do not have the capabilities to directly connect to the Internet via the TCP/IP protocol suite.

In the current context, the COVID-19 global pandemic has increased the transition towards digital technology, as it has served as an element to prevent the disconnection of multiple sectors [Wright, 2020]. As [Agrawal, 2021] mention, the technologies associated with Industry 4.0 have been critical to the response of companies towards the crisis. Moreover, the level of implementation of such technologies was related to the ability of companies to respond to the crisis, as shown in figure 1.

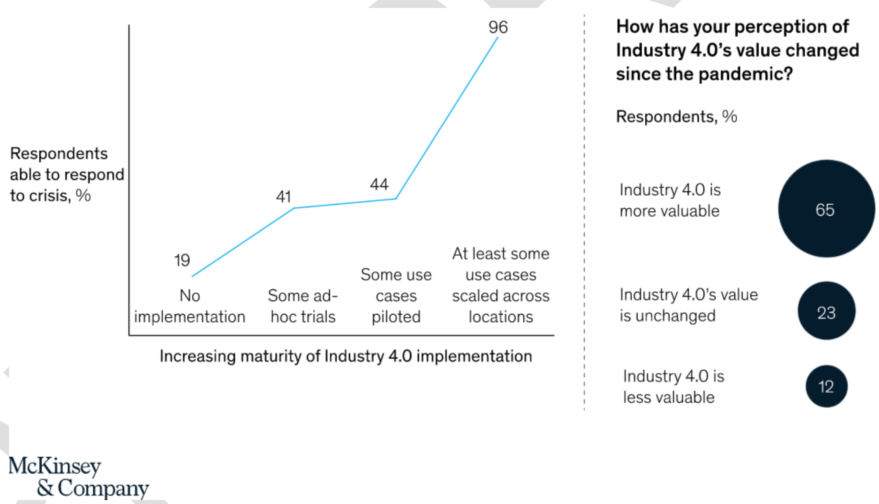

Figure 1 Industry 4.0 implementation vs resilience [Agarwal, 2021].

The pandemic has also impacted the education sector. Students worldwide have been forced to continue their studies at home due to lockdown preventive measures. The adaptation requires them to take advantage of technologies such as the Internet of Things (IoT) to continue with their education; however, this approach is generally limited to online meetings and assignments [Darma, 2020]. In this context, there is an increased challenge in disciplines that require technical skills, such as STEM

(Science, Technology, Engineering, and Mathematics) and medicine. A potential way of learning those skills is by having online access to real environments where students can remotely manipulate devices in a way like an in-person approach. By doing so, they could test their projects in physical devices and monitor real behaviors to validate them against analytical and simulation-based results. Therefore, implementing IIoT for laboratory settings is a potential solution that benefits education.

The transition towards Industry 4.0 technologies demands the connection of devices to the Internet in both previously presented cases (industry and education). To be connected enables access to a wide range of digital technologies (Machine Learning, Cloud Computing, among others) and ensures safe, remote access to machines and devices for monitoring and control.

The objective of the work presented is to apply the OPC UA standard for communication between a Virtual Instrument and a simulated Programmable Logic Controller that interacts with electromechanical processes in a simulator, to implement IIoT via a Web Services application for control and variable monitoring, therefore enabling applications in industry and remote education. In conformity with the proposed objective, the following hypothesis is formulated: the use of the OPC UA standard for connecting a PLC and a Virtual Instrument enables low-cost, platform independent IIoT for industrial and academic applications.

A main element mentioned in the objective is the Programmable Logic Controller (PLC), which is central for automation in industry. It is a digital computer used for the control of electromechanical processes. In contrast with general-purpose computers, a PLC has features that allow it to interact with machines in an industrial environment: multiple inputs and outputs, and increased resistance to thermal, mechanical, and noise perturbations [Lamb, 2013].

As a PLC generates data, the lack of its rapid accessibility and use presents both a challenge and an opportunity. Here is where the current period, the Fourth Industrial Revolution, has its origin. It is based on what are known as cyber-physical production systems, in which the real and digital worlds are connected and interact continuously [Aulbur, 2016].

According to [Lorenz, 2015], nine technology trends are enabling the transformation towards Industry 4.0. These will transform production by integrating isolated manufacturing cells into an automated and optimized production flow. Among these trends, the Industrial Internet of Things (IIoT) is of special interest because it develops as "[...] a natural extension of prior levels of connectivity achieved by the Internet and World Wide Web (WWW) to include physical objects and systems." [Chou, 2019]. IIoT is based upon the Internet of Things (IoT), which emerged as a network that enables machines to communicate among each other [Koc, 2019].

The functionality provided by IoT is adapted to an industrial level with IIoT, and its features evolve from legacy monitoring systems used in industries that used PLCs for control, along with *Supervisory Control And Data Acquisition* (SCADA) systems. The difference between legacy and IIoT systems is the connectivity of the latter over an Internet Protocol (IP) network structure [Zhou, 2017]. Therefore, to fully understand IIoT, it is important to develop on the topics of SCADA systems and IP networks.

A SCADA system consists of software and hardware elements that allow industries to remotely control and monitor industrial processes in real-time, allowing operators to interact with them through devices such as Human-Machine Interfaces (HMIs). Its architecture makes use of PLCs and data acquisition components such as Remote Terminal Units (RTUs). In the last two decades of the century, SCADA evolved to use Local Area Networking (LAN) to connect similar systems; however, a disadvantage of these first interconnected systems is that most protocols were unique to specific vendors, therefore devices from different brands were not able to communicate [Inductive Automation, 2018].

The interconnection of systems, as shown in the case of SCADA, is an important step to take in any context where various devices (computers or other hardware devices) have relevant information that would be valuable to communicate. In such contexts, establishing a network allows them to exchange information and cooperate. Models have been created to describe the different layers in the network; the most common general model is the Open Systems Interconnection (OSI) reference model [Kozierok, 2015]. It was defined by the International Standard

Organization (ISO) with the purpose of "[...] open communication between different systems without requiring changes to logic of the underlying hardware and software" [Jasud, 2017].

This model divides the networking functions into a set of seven conceptual layers (Physical, Data Link, Network, Transport, Session, Presentation, and Application), each representing an abstraction level, i.e., its relationship with the actual hardware; As the layers increase, the interaction with hardware becomes less direct, and the interaction with software extends. The layers can be divided into two groupings: the first one is denoted as the "lower layers" and comprises layers 1-4. It is concerned with data formatting, encoding, and transmission over the network. The second grouping is entitled the "upper layers", comprising layers 5-7. It is focused on the interaction with the user and the implementation of applications that run on the network [Kozierok, 2015].

A different architectural model to represent the functional division of a network is the TCP/IP model, which consists of four layers that approximately match six layers of the OSI model [Kozierok, 2015]. This model eases the understanding of the protocol with the same name, described as "[...] a strong, fast, scalable, and efficient suite of protocols." [Blank, 2004]. It consists of a collection of network protocols, among which the most important ones are the Transmission Control Protocol (TCP) and the Internet Protocol (IP) [Kozierok, 2015]. The relevance of TCP/IP is such that it is referred to as "the language of the Internet" because it is the most widely used protocol for communicating on it. As mentioned in the SCADA systems and observed on the Internet with TCP/IP, protocols are crucial for exchanging information among devices; "a protocol represents communication between logical or physical devices at the same layer of the model [...]" [Kozierok, 2015]. The protocol that a network uses depends on the application. A useful classification distinguishes between protocols used for Information Technology (IT) and Operational Technology (OT); while IT refers to business and office networks, OT is applied to industrial processes and factories [Bloem, 2014].

Nevertheless, is crucial to consider that industrial communications tend to differ according to the application and the brands of components, as mentioned while describing SCADA systems. Industrial protocols are central for automation and control applications in the context of the Third and Fourth Industrial Revolutions. In the early 1980s, vendors and end-users were demanding to establish a global protocol standard. As a result, a wide range of standards was created, including IEC 61158 which covers nine protocol profiles: FOUNDATION Fieldbus, CIP, PROFIBUS/PROFINET, P-NET, WorldFIP, INTERBUS, CC-Link, HART, and SERCOS [Knapp, 2015].

In the context of IIoT, a crucial challenge is to offer a machine-to-machine (M2M) communication that can align and converge different protocols into a unified scheme. It also is to provide an industry-level integration between OT and IT, which have traditionally been managed and controlled independently from one another [Bloem, 2014].

The abovementioned challenges can be resolved with a well-designed solution that ensures data transfer regardless of protocol variety; an important and attractive approach for achieving that goal is called OPC, which stands for Open Platform Communications. The OPC standard (now known as OPC Classic) was released in 1996 with the purpose of abstracting PLC-specific protocols into an interface that would convert generic requests (read/write) into device-specific requests, and vice-versa. A more recent version of the standard, called OPC UA (Unified Architecture), was developed to address the needs of service-oriented architectures in manufacturing systems, such as security and data modeling [OPC Foundation, 2021].

The OPC architecture specifies interfaces for communication among different elements. Its communication structure consists of various OPC clients communicated through one or more OPC servers. Each server vendor supplies a code that determines the data and devices to which a server has access, along with information on how it physically accesses that data [OPC Foundation, 2003]. The client/server relationship is shown in figure 2.

An OPC server is comprised of various objects: the server, groups, and items. The OPC server object holds information about itself and contains the OPC group objects. Each OPC group object keeps the information about the group, and it contains and organizes OPC items locally. Finally, OPC items represent connections to data sources inside of the server. All the access to OPC items is done via an OPC group, within which they are defined and contained [OPC Foundation, 2003]. OPC UA uses a client-server concept like OPC Classic. It defines a server as an application that exposes its information to other applications; inversely, a client is an application that consumes information from others [Mahnke, 2009].

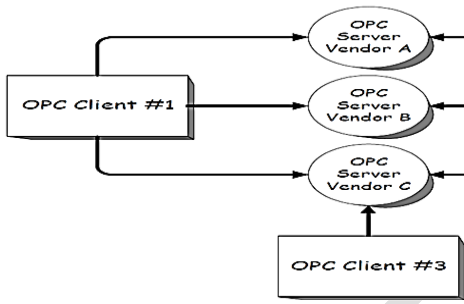

Figure 2 OPC client/server relationship [OPC Foundation, 2003].

The platforms for implementing OPC UA expand in contrast with OPC Classic: the former can be implemented in any platform and Operating System (OS) including embedded processors, Personal Computers (PCs) and servers, while OPC Classic only runs on Windows platforms. The ease of implementation simplifies the integration of multi-vendor systems. Also, OPC UA supports dynamically updated Object-Oriented information modeling and provides predefined data structures [Kominek, 2018].

The purpose of using OPC in IIoT is to offer a robust and efficient data exchange between protocols and systems. It can be implemented using an OPC architecture, where the OPC server acts as a gateway for sharing data on the Internet. It offers an intermediary role for vertical integration, allowing data transfer from the device level to the application level [Ghazivakili, 2018].

There are currently several implementations of OPC for IIoT. For instance, smart devices known as SmartBoxes have been developed by various researchers, such as the effort done by [Torres, 2019] in which each SmartBox (based on a Raspberry Pi hardware platform) can be used for real-time M2M communications using OPC-

UA, along with MQTT (Message Queuing Telemetry Transport). The infrastructure proposed by [Vazquez, 2018] offers various approaches for promoting the learning of industrial automation systems, using OPC servers to exchange digital and analog data that students could use to verify the programming of a PLC. One of the proposed approaches is a remote laboratory where a physical lab could be accessed via Internet, using a web server.

2. Methods

An industrial application was chosen as a case study. It corresponds to a SCADA system with two processes used as a final stage in the manufacturing of plastic bottles. The first process monitors and counts the number of bottles to be packed inside a box, passing by a sensor. It uses an industrial traffic light that changes according to the counted bottles; if the number exceeds a critical value, an alarm is activated and continues active up to two seconds after the value is no longer critical. The second process activates the electric motor of a conveyor band that transports bottles and sends information about the expected position of a bottle being transported by the band, for an operator to visualize as an animation.

The program used for creating the simulated plant was Festo FluidSIM® 4.2p Pneumatics. The system (Figure 3) was divided into input and output sections.

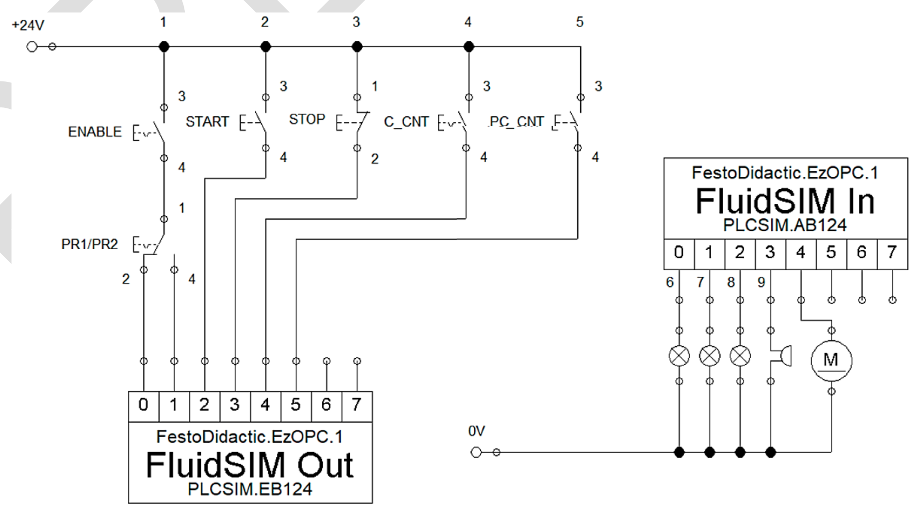

Figure 3 Input (left) and output (right) sections of the simulated plant in FluidSIM.

As it can be observed, both sections are connected to ports called "FluidSIM Out" and "FluidSIM In", which correspond to the output and input of the software towards other programs, respectively. The addresses need to be associated with reachable memory areas of the PLC; therefore, it was important to consider a specific PLC model. The selected device was a Siemens Simatic S7-300/CPU 312C. Its inputs cover the byte address range 124-126, and its outputs, the range 124-125. The ports in FluidSIM allow interaction via the OPC Classic standard. For that purpose, the software Festo Didactic EzOPC V5.6 was selected. It uses the OPC Data Access (Classic) standard and allows connectivity between process simulations and controllers.

The entire procedure of creating, programming, and simulating the PLC was done with the software suite TIA (Totally Integrated Automation) Portal, version V15. The selected PLC has the following characteristics: 10 digital inputs, 6 digital outputs, MPI communication interface and 64 kb of work memory [Siemens, 2021]. It has the required number of digital inputs and outputs to use with the plant. Additionally, it is part of the S7-300 series, which can be simulated using S7-PLCSIM.

The program for the PLC was created after having the tags defined, using the Ladder Logic (LAD) programming language. It was divided into eleven networks, which can be classified into six functional groups. The first group (Networks 1 and 2) is used for starting, interlocking, and stopping Processes 1 and 2, respectively. The second group (Network 3) has the purpose to perform the bottle counting process. The third group (Networks 4-6) corresponds to the three indicator lights of the industrial traffic light used in Process 1. Each network contains conditions that depend on the number of counted bottles. Group 4 (Networks 7 and 8) is responsible for maintaining the alarm active up to two seconds after the value of the counter stops being in a critical condition. The goal of the following group (Network 9) is to activate the electric motor of the conveyor band while Process 2 is active. Finally, the last group (Networks 10 and 11) controls the flow of the animation that gets visualized on the VI to show the position of a bottle moving on the conveyor band.

Next, the OPC UA server was configured. Its purpose is to allow data exchange between the simulated PLC in S7-PLCSIM and the VI, which was created using the software LabVIEW from National Instruments (NI). For that reason, software from NI was used to implement the server, as it offers modules for creating, configuring, and managing OPC UA clients and servers. Following the instructions provided by [National Instruments, 2020a] [National Instruments, 2020b], the LabVIEW Datalogging and Supervisory Control (DSC) 2020 module and the NI OPC Servers 2016 add-on were installed. The DSC module includes tools for real-time monitoring and for networking LabVIEW elements and OPC devices; NI OPC Servers offers functionality for creating and managing channels and devices in an OPC UA server, along with providing a client that connects to LabVIEW applications.

An additional step was needed because the IP address used in S7-PLCSIM is private and cannot be directly used with external applications. The tool NetToPLCsim was used for that purpose; it acts as a network extension of the PLC simulation via TCP/IP [Wiens, 2021]. Consequently, the tags for the variables to share between the PLC and the VI were defined. Figure 4 shows the tag tables for the NI OPC server and the PLC.

Tag Name	Address	Data Type	Scan Rate	Scaling
PR1_EN	I124.0	Boolean	100	None
PR2_EN	I124.1	Boolean	100	None
START	I124.2	Boolean	100	None
STOP	I124.3	Boolean	100	None
C_CNT	I124.4	Boolean	100	None
PC_CNT	I124.5	Boolean	100	None
PR1	M0.0	Boolean	100	None
PR2	M0.1	Boolean	100	None
START_HMI	M0.4	Boolean	100	None
STOP_HMI	M0.5	Boolean	100	None
CNT0_VAL	MW1	Word	100	None
TMR0_VAL	MW3	Word	100	None
CNT1_VAL	MW5	Word	100	None
G_LIGHT	Q124.0	Boolean	100	None
Y_LIGHT	Q124.1	Boolean	100	None
R_LIGHT	Q124.2	Boolean	100	None
ALARM	Q124.3	Boolean	100	None
MOTOR	Q124.4	Boolean	100	None

a) NI OPC server

PLC tags

Name	Data type	Address	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Comment
PR1_EN	Bool	%I124.0		True	True	True	Enable Process 1
PR2_EN	Bool	%I124.1		True	True	True	Enable Process
START	Bool	%I124.2		True	True	True	Start (button)
STOP	Bool	%I124.3		True	True	True	Stop (button, NC)
C_CNT	Bool	%I124.4		True	True	True	Choose to count down/up
PC_CNT	Bool	%I124.5		True	True	True	Piece Count
G_LIGHT	Bool	%Q124.0		True	True	True	Green Light
Y_LIGHT	Bool	%Q124.1		True	True	True	Yellow Light
R_LIGHT	Bool	%Q124.2		True	True	True	Red Light
ALARM	Bool	%Q124.3		True	True	True	Alarm for excess of pieces (10+)
MOTOR	Bool	%Q124.4		True	True	True	Conveyor Band Motor
PR1	Bool	%M0.0		True	True	True	Process 1
PR2	Bool	%M0.1		True	True	True	Process 2
PIECE_CNT	Counter	%C0		True	True	True	Piece counter
CNT0_VAL	Int	%MW1		True	True	True	Value for piece counter
Timer_Alarm	Timer	%T0		True	True	True	Timer for alarm (turns itself off)
TMR0_VAL	Int	%MW3		True	True	True	Value for TMR0
TMR_0	Bool	%M0.2		True	True	True	Timer for alarm
Timer_Bottle	Timer	%T1		True	True	True	Timer for animation transition
TMR_1	Bool	%M0.3		True	True	True	Timer for bottle animation
BOTTLE_CNT	Counter	%C1		True	True	True	Bottle counter
CNT1_VAL	Int	%MW5		True	True	True	Value for bottle movement counter
START_HMI	Bool	%M0.4		True	True	True	Start (VI)
STOP_HMI	Bool	%M0.5		True	True	True	Stop (VI)

b) PLC

Figure 4 Tag tables.

The Virtual Instrument (VI) was configured once the tags were defined. The software used to develop it was NI LabVIEW (2020, 32-bit version). The elements used in the panel were added considering the variables that were defined in the NI OPC server. Animation frames were created with Photoshop. The instructions provided by Travis [Travis, 2007] were used to import the frames to the "Picture Ring" elements in order. The resulting panel is presented in figure 5.

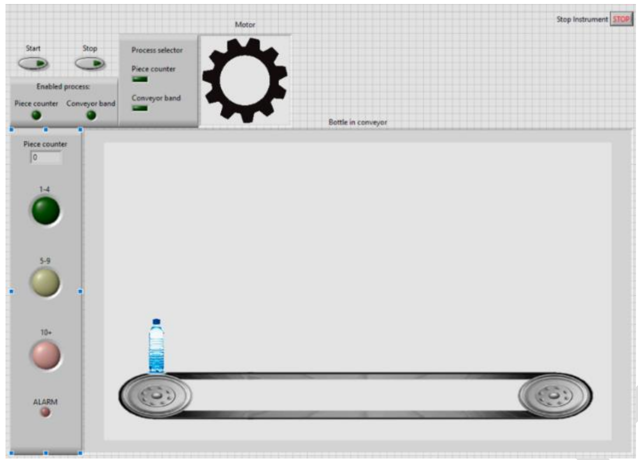

Figure 5 Elements of the front panel of the VI after adding animation frames.

Later, the OPC UA tags were imported to LabVIEW as bound variables. The process consisted of dragging each variable from the project menu to the block diagram and then connecting it to its associated element with a wire. All elements were enclosed in a “while loop” structure that executes every 100 ms (update rate). The block diagram of the VI is shown in figure 6.

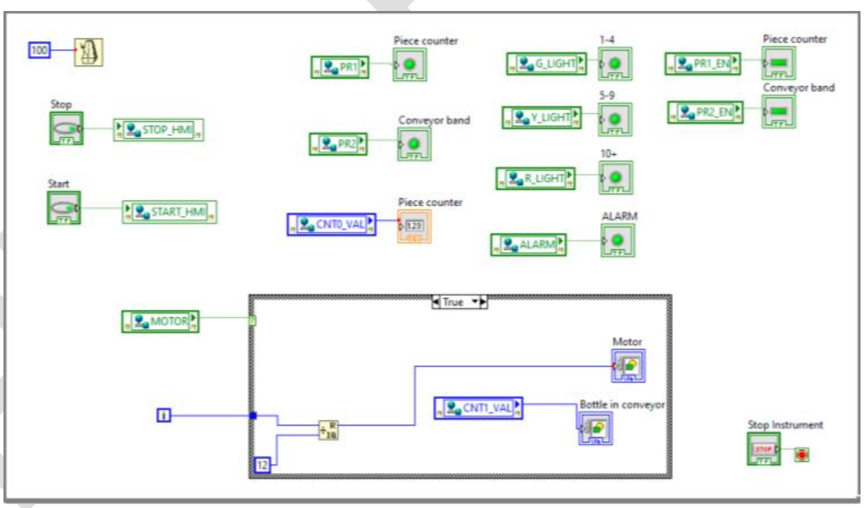

Figure 6 Block diagram of the Virtual Instrument.

With all the elements (simulated plant, PLC, OPC UA server, and VI) configured, various tests were performed to validate their connection; an example is presented in figure 7.

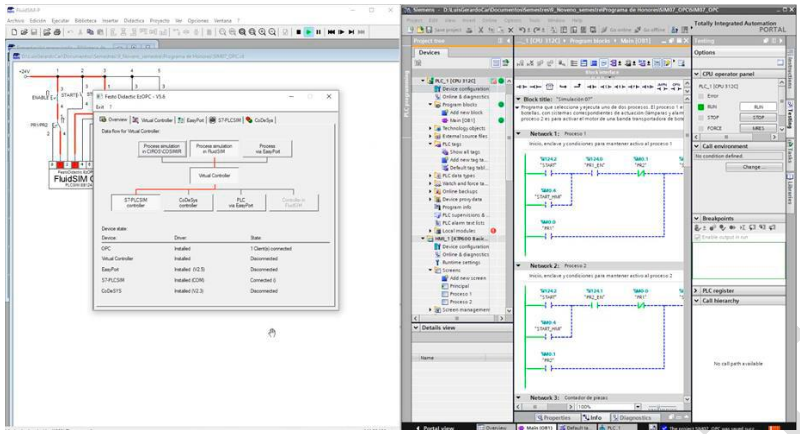

Figure 7 Connection test between FluidSIM and TIA Portal.

Finally, the VI was exported as a Web Services application using a tool provided by LabVIEW, called Web Publishing Tool. It was exported as an "embedded" application, to run an interactive web browser version of the VI. Figure 8 shows its deployment in Internet Explorer.

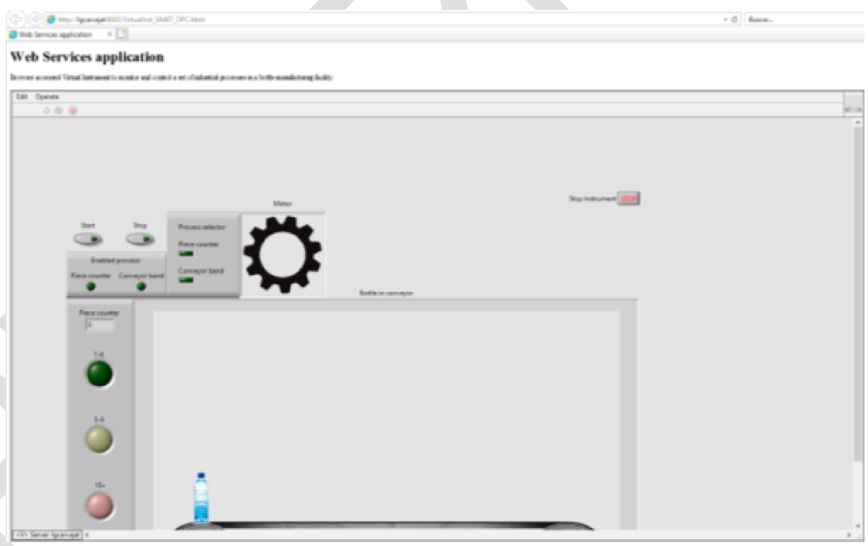

Figure 8 VI deployed as a Web Services application in Internet Explorer.

3. Results

The methodology successfully allowed to connect all the intended elements: the simulated plant in FluidSIM was connected with the simulated PLC in S7-PLCSIM via EzOPC; the latter became an OPC client of the NI OPC server and was able to exchange information with the VI in LabVIEW. Furthermore, the VI was executed as a Web Services application in Internet Explorer.

Images are presented below to demonstrate the above-mentioned; each one shows the simulated plant in FluidSIM on the left and the VI as a Web Services application in Internet Explorer on the right. Figure 9 illustrates Process 1 (counting bottles and activating the industrial traffic light and the alarm according to the value); figure 10 demonstrates the execution of animations in Process 2 (activating the motor of a conveyor band to transport bottles).

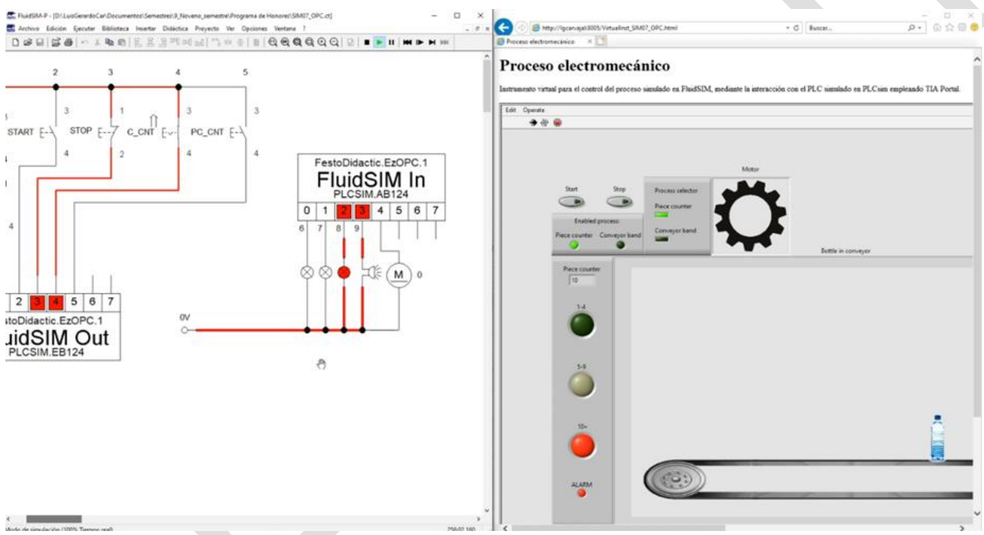

Figure 9 Execution of Process 1 in the plant and Web Services application.

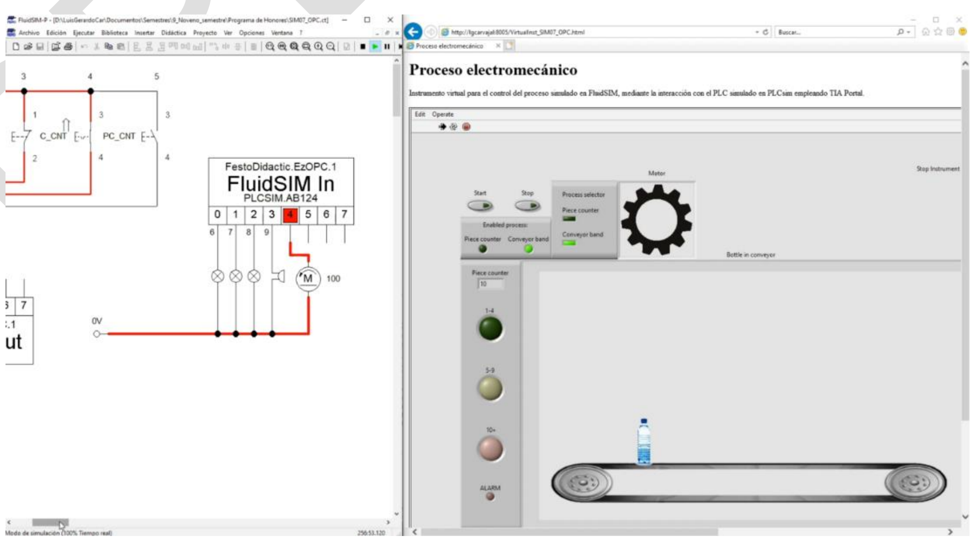

Figure 10 Execution of Process 2 in the plant and Web Services application.

4. Discussion

The results validate the use of OPC UA for control and monitoring purposes with a PLC and a VI, enabling data transfer from/to the plant. Since all the elements are connected and can exchange information for control and monitoring processes, they constitute a SCADA system. Moreover, the system corresponds to an implementation of IIoT because the communication structure allows network access to a plant via a web browser. Even though the implementation procedure may be perceived as extensive, it can be done straightforward and without issues by having the appropriate software installed; the steps that were presented in the methodology can then be replicated to achieve communication using the OPC UA standard. Once configured, the resulting system can be executed continuously while the OPC server remains active.

Minor technical difficulties occurred during the last part of the methodology when attempting to run the Web Services application; it only worked after deactivating antimalware software and using Internet Explorer as the web browser. The fact that only a legacy browser can run the published Web Service and that antimalware software blocks its execution shows a weakness of the publishing tool provided by National Instruments in LabVIEW.

A connection test in a real environment with physical devices was not possible because of the nature of the current situation regarding COVID-19 lockdown policies. However, the results can be extrapolated to in situ applications both in industrial and academic environments. It would only be needed to replace FluidSIM with a physical plant and the simulated PLC with its real equivalent. Besides, the current implementation is already useful for educative purposes in a distance learning approach because students could do course practice of related topics in a simulated environment.

5. Conclusions

The work presented above demonstrates that it is possible to use the OPC UA standard to communicate an industrial process controlled by a PLC with a VI that runs on a Web Services application, achieving an IIoT implementation of a SCADA system. The solution offers a lower cost than alternatives in the IIoT service market; the reason is the relatively low cost of the program licenses. It also enables platform independence since the software can operate on Windows directly or through a Virtual Machine (VM).

The results indicate that it is feasible to replicate the methodology to industrial processes to achieve their remote monitoring and control. The same methodology could also be applied to academic institutions for students to interact remotely with PLCs and industrial plants. By doing so, they would be able to monitor variables and test the execution of programs developed during class. The above would result in a tangible benefit to their learning experience since they would practice in a physical environment without the need of going to a laboratory. This advantage is more substantial in courses that cover subjects such as industrial automation, control systems, production lines, and manufacturing processes.

As a result, students that use a PLC-VI implementation through the OPC UA standard, as described here, can remotely work on projects and validate their functionality from any location with Internet access.

Furthermore, it bridges the learning gap for students who have online study programs, allowing them to get quality results that exceed the capabilities of simulators; the same applies to students who are currently in situations such as the lockdown due to the pandemic.

From the industry perspective, managers and supervisors can take advantage of such implementation by eliminating the need of going to a plant in person to monitor the state of a machine, its produced units, and its real-time operation. The aforementioned helps to avoid risks and extended working hours while it favors flexibility and management of production resources.

An area of improvement is the web browser compatibility of Virtual Instruments. For that reason, a recommendation for further work is to develop a software tool that can export a VI to a Web Services application that runs on current web browsers, either on desktop or on mobile devices. A further area of opportunity is the development of comprehensive software that manages the entire process of connecting industrial devices through the OPC UA standard to offer a ready-to-use IIoT solution.

6. Bibliography and References

[1] Agarwal, M., Dutta, S., Kelly, R., & Millán, I. (2021, January 15). COVID-19: An inflection point for Industry 4.0. Retrieved from McKinsey & Company: https://www.mckinsey.com/business-functions/operations/our-insights/covid-19-an-inflection-point-for-industry-40.

[2] Aulbur, W., CJ, A., & Bigghe, R. (2016). Skill Development for Industry 4.0. Roland Berger GMBH. Retrieved from: http://www.globalskillsummit.com /Whitepaper-Summary.pdf.

[3] Blank, A. G. (2004). TCP/IP Foundations. Alameda: Sybex.

[4] Bloem, J., Van Doorn, M., Duivestein, S., Excoffier, D., Maas, R., & Van Ommeren, E. (2014). The Fourth Industrial Revolution. Things to Tighten the Link Between IT and OT. VINT research report (3), 1-39.

[5] Chou, S. Y. (2019). The Fourth Industrial Revolution: Digital Fusion with Internet of Things. Journal of International Affairs, 107-120.

[6] Darma, D. C., Ilmi, Z., Darma, S., & Syaharuddin, Y. (2020). COVID-19 and its Impact on Education: Challenges from Industry 4.0. Aquademia, 4(2). doi:10.29333/aquademia/8453.

[7] Ghazivakili, M., Demartini, C., & Zunino, C. (2018). Industrial Data-Collector by enabling OPC-UA standard for Industry 4.0. 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS) (pp. 1-8). Imperia: IEEE. doi:10.1109/WFCS.2018.8402364.

[8] Inductive Automation. (2018, September 12). What is SCADA? Retrieved from Inductive Automation: https://www.inductiveautomation.com/resources /article/what-is-scada.

[9] Jasud, P. V. (2017). The OSI Model: Overview on the Seven Layers of Computer Networks. International Journal for Innovative Research in Science & Technology, 4(3), 116-124.

[10] Koc, T. C., & Teker, S. (2019). Industrial Revolutions and its effects on quality of life. PressAcademia Procedia (PAP), 9, 304-311.

[11] Kominek, D., & Brubaker, S. (2018). OPC UA vs. Classic OPC. Why Switch and How to Get Started. Matrikon Inc. & Beeond, Inc.

[12] Knapp, E. D., & Langill, J. T. (2015). Industrial Network Security. Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems (Second ed.). Waltham: Elsevier.

[13] Kozierok, C. M. (2015). The TCP/IP Guide. San Francisco: No Starch Press.

[14] Lamb, F. (2013). Industrial Automation Hands-On. New York: McGraww-Hill Education.

[15] Lorenz, M., Rüßmann, M., Waldner, M., Engel, P., Harnisch, M., & Justus, J. (2015, April 9). ANSFORMATION, INDUSTRY 4.0: https://www.bcg.com/en-mx/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries.

[16] Mahnke, W., Leitner, S. H., & Damm, M. (2009). OPC Unified Architecture. Ladenburg: Springer.

[17] MarketsandMarkets. (2019, December). Industry 4.0 Market by Technology (IoT, Artificial Intelligence, Industrial Metrology, Industrial Robotics, AR & VR, Blockchain, 3D Printing, Digital Twin, and 5G – Offering, Application, and End Users) and Geography- Global Forecast to 2024: https://www.marketsandmarkets.com/Market-Reports/industry-4-market-102536746.html.

[18] National Instruments. (2020a, November 24). Adding a PLC to NI OPC Servers. Retrieved from National Instruments: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YIRNCA4&l=en-US.

[19] National Instruments. (2020b, December 21). Connect LabVIEW to Any PLC Using OPC. Retrieved from National Instruments: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x0MPCAY&l=en-US.

[20] OPC Foundation. (2003). OPC Data Access Custom Interface Specification 3.0. Austin: OPC Foundation.

[21] OPC Foundation. (2021). What is OPC? Retrieved from OPC Foundation: https://opcfoundation.org/about/what-is-opc/.

[22] Siemens. (2021, March 12). 6ES7312-5BF04-0AB0 Data sheet. Retrieved from Siemens Industry Mall: https://mall.industry.siemens.com/mall/en/es/Catalog/Product/6ES7312-5BF04-0AB0.

[23] Torres, P., Dionísio, R., Malhão, S., Neto, L., Ferreira, R., Gouveia, H., & Castro, H. (2019). Cyber-Physical Production Systems supported by Intelligent Devices (SmartBoxes) for Industrial Processes Digitalization. 2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI) (pp. 73-78). Florence: IEEE. doi:10.1109/RTSI.2019.8895553.

[24] Travis, J., & Kring, J. (2007). LabVIEW for Everyone (Third ed.). Upper Saddle River: Prentice Hall.

[25] Vazquez Gonzalez, J. L., Barrios Aviles, J., Rosado Muñoz, A., & Alejos Palomares, R. (2018). An Industrial Automation Course: Common Infrastructure for Physical, Virtual and Remote Laboratories for PLC Programming. International Journal of Online and Biomedical Engineering (iJOE), 14(8), 4-19. doi: https: //doi.org/10.3991/ijoe.v14i08.8758.

[26] Wiens, T. (2021). NetToPLCsim - Network extension for Plcsim. Retrieved from NetToPLCsim: http://nettoplcsim.sourceforge.net.

[27] Wright, C. C. (2020, August). COVID-19 Pandemic and the Digital Revolution. ASA Monitor, 84, 30. doi:10.1097/01.M99.0000695184.52961.46

[28] Zhou, C., Damiano, N., Whisner, B., & Reyes, M. (2017). Industrial Internet of Things: (IIoT) applications in underground coal mines. Min Eng, 50-56. doi:10.19150/me.7919.