

Communication between PLC different vendors using OPC server improved with application device

Ignatius Deradjad Pranowo, Y. B. Theo Bagastama, Thomas A. F. Wibisono

Politeknik Mekatronika Sanata Dharma, Indonesia

Article Info

Article history:

Received Jul 27, 2019

Revised Dec 3, 2019

Accepted Dec 21, 2019

Keywords:

Data communication

Different platform

KEPServerEX6

OPC server

PLC

SCADA

ABSTRACT

Many industries often use different devices and controllers in automation systems. They all face the same difficulty how to exchange data between all those components. This paper proposed the implementation of OPC Server as software interface on communication between two different controllers, PLC Mitsubishi and PLC Omron. The main advantage of the method is the compatibility and solution for the factory difficulty problem because of using several driver controller. The compatibility among the different platforms of both controller, PLC Mitsubishi and PLC Omron, can be reached by use of KEPServerEx6 (OPC server) as a software interface. To test the compatibility amongst two different controllers, there was developed and implemented two field application devices, bottle unscramble and bottle filling station. This implementation shows OPC Server technology resolving data compatibility issues between different platforms and reducing development costs. It is envisaged that the method can be very useful to realize integration.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ignatius Deradjad Pranowo,
Politeknik Mekatronika Sanata Dharma,
Yogyakarta, Indonesia.
Email: dradjad@pmsd.ac.id

1. INTRODUCTION

Remote monitoring and control systems are also referred to as supervisory control and data acquisition (SCADA), the process of collecting data from physical devices to be monitored and applying commands to control devices. SCADA systems typically consist of a human-machine interface (HMI), monitoring computer, remote terminal unit (RTU), programmable logic controller (PLC) and communication infrastructure [1-3]. In a system that uses several different devices and must exchange data such as PLC and HMI; they must collect, analyse, and display data from various devices using different protocols. This task is not easy and error-prone. Every device driver requires software from its company. Changes in protocol specifications result in communication error and adjustments must be made, which are time-consuming and expensive [4].

OPC (OLE for process control) is a standard data exchange in process control, which provides a practical method for exchanging data between software configurations based on Windows platforms and controllers based on field controls [5, 6]. Before OPC became known, to store field control device data, the software configuration must provide communication drivers for each device. In fact, there are various types of the field control devices that are constantly being updated, and the communication protocols used by each control device are also different [4, 7]. OPC technology is a hardware and software interface standard using client/server mode based on COM (component object model)/DCOM (distributed component

object model), which offers a general standard mechanism for client's and server's data communication, exchange, and supports the network distributional application procedure communication as well as the application procedure communication in different platforms [8]. The configuration software can access the data of the field control device in accordance with the interfaces provided by OPC server without needing to understand the detailed information of these hardware devices so as to enhance the flexibility, efficiency, openness and interoperability [9].

Using OPC standard is a good method of communication but still have problems. In the development of OPC servers, there are problems with how to collect, process data from different OPC servers, OPC server connections, and read-write from multi OPC servers. Whereas in the OPC client development work, how to use the application program to create OPC server objects to connect different OPC servers, OPC objects to collect process data from different OPC server devices [7]. Therefore, both of these problems deserve to be studied in depth. This paper provides the design and implementation of OPC server application which connecting between the devices and PLCs. The two prototypes applications of process control are available for real-time communication between PLC Mitsubishi and PLC Omron based on OPC technology. The organization of this paper is as follows: section 2 gives an implemented design of OPC server or method on communication between two different platform devices (PLC Mitsubishi and PLC Omron). The result and discussion about OPC Server implemented on PLC Mitsubishi and PLC Omron is provided in section 3. Section 4 gives the conclusions.

2. METHOD

2.1. Conventional communication system

Conventional communication works by communicating every each end connection. Every end-to-end connection required a custom driver to facilitate communications between specific endpoints. For example, if an HMI needed to communicate with a PLC, it required a custom HMI driver written for the specific protocol used by the PLC. Figure 1 shows the use of custom drivers between every endpoint meant that even a small number of devices and applications quickly involved the use of many drivers. Each driver establishes its own connection to the device or controller that it is designed to communicate with. These things bring the problems of custom drivers in conventional communication.

2.2. OPC communication system

In software network as a central controller, each device are connected to the central controller with the help of standard protocols [10]. OPC is used to answer one of the automation industry's biggest challenges: how to communicate between devices, controllers, and/or applications without getting caught up in the usual custom driver-based connectivity problems. OPC abstracts the data source (e.g., PLC) and data sink (e.g., application device) implementation details from each side so data can be exchanged between them without requiring them to know anything about each other's native communication protocol and internal data organization [9].

Figure 2 shows that OPC acts as an abstraction layer that sits between the data source and the data sink, allowing them to exchange data without knowing anything about each other. The OPC device abstraction is realized by using two, specialized OPC components called an OPC client and OPC server [4]. What is important to note is that just because the Data Source and Data Sink can communicate with each other via OPC does not mean their respective native protocols are no longer necessary or have been replaced by OPC. Instead, these native protocols and/or interfaces are still present, but only communicate with one of the two OPC components.

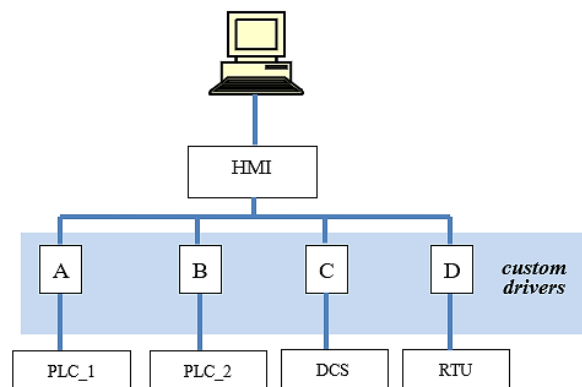


Figure 1. Configuration of conventional communication

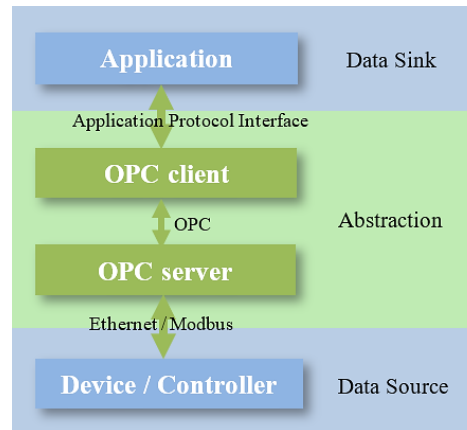


Figure 2. OPC client/server architecture [4]

2.3. PLC device and application device

PLC used is Omron CP1E and Mitsubishi FX3U. PLC Omron is used to control the bottle filling process, and PLC Mitsubishi is used to control the bottle unscramble process. Each PLC uses a software driver from its manufacturer; PLC Omron uses CX Programmer while PLC Mitsubishi uses GX Work3. The application device used in this project is two prototypes which consist of the bottle unscramble station controlled by PLC Mitsubishi, and the bottle filling station controlled by PLC Omron. The two applications were made with operational functions the processes as they represented. Actually, the two applications only needed to show the working of communication system using OPC Server control two different PLC platforms.

2.4. The proposed system

This work was proposed the use of OPC Server to achieve the integration and communication between two PLC different vendors. OPC abstracts the data source (i.e., PLC) and data sink (i.e., application device) implement details from each side so data can be exchanged between them without requiring them to know each other's native communication protocol. The composition of real-time communication developed in the system is shown in Figure 3. Systematic basic hardware items are displayed as follows: PC (personal computer) equipped with ordinary parts. In addition, A PC-Adapter which is used to convert PC's RS232 interface to the PLC interface is furnished. The CX Programmer software can be downloaded to the PLC Omron with the PC-Adapter. Also, the GX Work3 software can be downloaded to the PLC Mitsubishi. The systematic basic software items are shown as follows: Windows 10 Professional, The KEPServerEX6 is a 64-bit OPC Server application program, CX Programmer and GX Work3 it issued to program both PLCs.

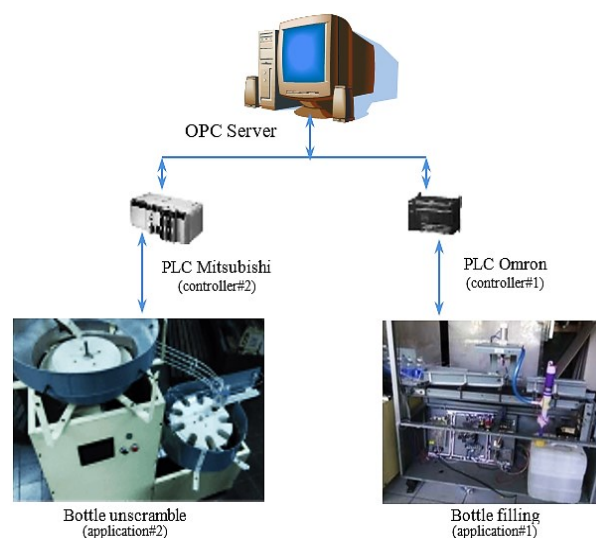


Figure 3. The configuration of communication between different PLC using OPC server

GX Work3 connects the OPC server through OPC interfaces and accesses the data of PLC Omron device register units by accessing the item object in the OPC server. Because group-object provides the interfaces to PLC Omron CP1E, the PLC Omron CP1E exchanges data with group-object but not item-object. The interfaces use the RS232 to realize. The OPC interfaces can connect OPC server after configuring the OPC server object, the group object and the item object. When the OPC interfaces are opened, the OPC customer's application programming will access the OPC server and realize the access to PLC Omron CP1E.

Both application devices, the bottle unscramble and the bottle filling, can communicate with each other through OPC Server software configuration settings. Through this arrangement, the process of filling water into a bottle will only do if it is getting bottles sent from the bottle unscramble process that have been arranged on position (the bottle's hole is above). This information is obtained because the controller of the bottle filling process reads the data released from the sensor as a result of the detection of the presence of the bottles sent from the bottle unscramble process. Meanwhile, the bottle unscramble process will stop, if the queue of bottles in the bottle filling process is still long, and it will reactivate if the queue is empty or depends on the desired PLC program. This information is received by the bottle unscramble process because it reading of data information from the bottle filling process. Since both PLC are not on the same platform, to be able to communicate with requires the OPC Server as software interface.

3. RESULT AND DISCUSSION

3.1. Configuration of variable and connection

The purpose of configuration connection is to set up a connection between the OPC Server and the client, and includes two following respects specifically [11, 12]. The first is hardware configuration. There are two purposes of PLC hardware system configuration for both PLC Omron CP1E and PLC Mitsubishi FX3U. One is to set basic parameter communication and the program addresses between PLC and PC which are used by OPC Server, the other is to open the memory and data section which the PC need to access and download the basic data and program to the PLC. Do run 'KepServerEx6' application, then complete both PLC hardware configuration of CP1E and FX3U, DI/DO modules, OPC Server and setting of the basic parameters of the system. Then, choose the communication port from the setting the PC Interface of the OPC Server, the COM port address and communication speed between PC-Adapter and PC. The second is OPC server configuration. Configuration should be finished in the COM before starting the OPC Server. Firstly, choose the network communication device and set its relational parameters by utilizing 'Setting the PC Interface' application in the KEPServerEX6. Secondly, run the Runtime application in KEPServerEX6 and set COM configuration parameter. In the end, enter the OPC parameter setting of the OPC Server, set communication requiring cycle, and select connection as a communication protocol between PC and PLC. The purpose of configuration variable is to set up the communication data variable between the OPC Server and PLC Omron CP1E / PLC Mitsubishi FX3U. Variable and parameter setting in KEPServerEX6 [13] are briefly presented in Table 1.

3.2. PLC program and KEPServerEX6

We can realize the real-time communication between PLC and process devices through command way. Before we design the real-time communication program between PLC and process devices, the two things should be finished [11]. First, the OPC Foundation offers a set of core components to browse other computers and communicate with them which do not be installed. Secondly, in order to set up a connection between object and OPC server object, we should configure the distributed COM environment between OPC server and OPC customer. The program of both PLC (Mitsubishi and Omron) are as in Table 2.

We can get communication data between bottles unscramble and bottle filling from the process record of the KEPServerEX6 feature. Table 3 shows sample data transfer of one cycle when the PLC Mitsubishi and PLC Omron communicate with each other (TX = to write the data; and RX = to read the data). At 13: 28: 39,416 the PLC Mitsubishi sends data to the PLC Omron; with details 13 data length and data content there are a series of numbers. PLC Omron receives data twice, the first at (time) 13: 28: 39,436 with details of data length 32 and the second at (time) 13: 28: 39,462 with details of data length 11, with each containing a series of data according to the length of the data. Reading of the data from this communication will be better understood through the display in the KEPServerEX6 window of each device (PLC Omron and PLC Mitsubishi), as shown in Figures 4-7.

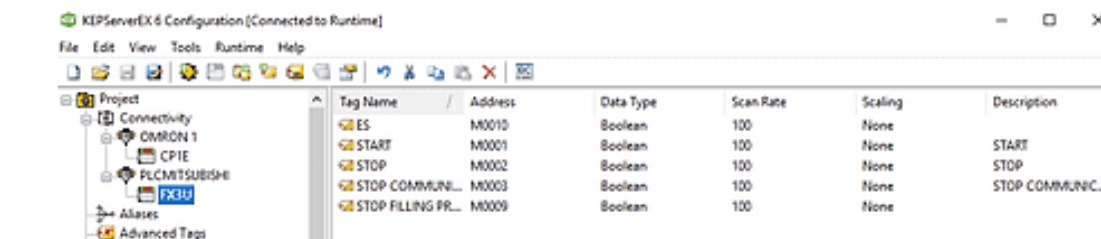
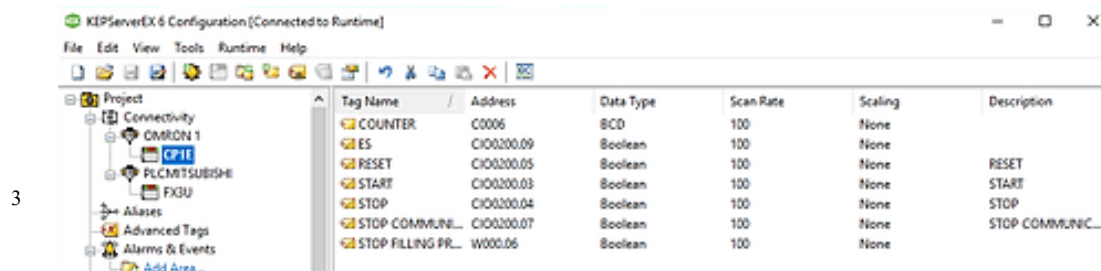
In the bottle filling application (controlled by PLC Omron CP1E), it is programmed to fill 5 times for one process cycle. The counter value will decrease according to the process that has occurred, and the process will stop when the counter value has been fulfilled (the setting value is 5, and will stop if the value is 0). As shown in Figure 4, the bottle filling has a value of 2, which means that 3 filling processes have been fulfilled and 2 remaining filling processes. In Figure 5, the counter will decrease according to the process from 2 to 1

from the setting value 5. Communication between the two PLCs with each different program can provide data to each other with the help of KEPServerEX6 (OPC server).

Figure 6 shows the Counter value is 0 (zero) it shows that the filling process has been fulfilled, and the value of the Stop communication is 1 (one) or active which will cause the process to stop. Figure 7 shows the Reset value is 1 which means that the Reset button has been activated. The counter returns to value 5 and will start for new process for refilling bottles five times. If the Stop button on the bottle unscramble is activated, it will cause the whole process of bottle unscramble and bottle filling to be stopped. This shows that the communication between the two is fulfilled (successfully sending and receiving data). Vice versa, if the emergency button on the bottle filling is pressed (activated), both processes will stop (communication is fulfilled; sending and receiving data is successful).

Table 1. Variable and parameter setting in KEPServerEX6

no	variable	setting	parameter to be set
1	channel	connectivity add channel wizard: select the channel	CJ1 for PLC Omron FX for PLC Mitsubishi
		specify the object	PLC Omron for channel 1 PLC Mitsubishi for channel 2
		select the hardware device type for communication	COM port
		specify port number	port 1 for Omron & port 2 for Mitsubishi
2	device	baud rate	9600
		diagnostic capture	enable
		connectivity add device wizard: specify the object	PLC Omron for device 1 r PLC Mitsubishi for device 2
		model	CJ1 for PLC Omron FX for PLC Mitsubishi
tag address	tag	ID	0.0.0
		data collection	enable
		connectivity property editor : channel 1 device 1 (omron)	tag name: add all input item (start, stop, etc.)
		channel 2 device 2 (mitsubishi)	tag name: add all input item (start, stop, etc.)
		addressed tag name: start, stop, etc. for Omron	start: X002; stop: X003 ... etc
		addressed tag name: start, stop, etc. for Mitsubishi	start: M000; stop: M001 ... etc



communication:
advanced tags

change average tag to be link tag
link the item with the corresponding one

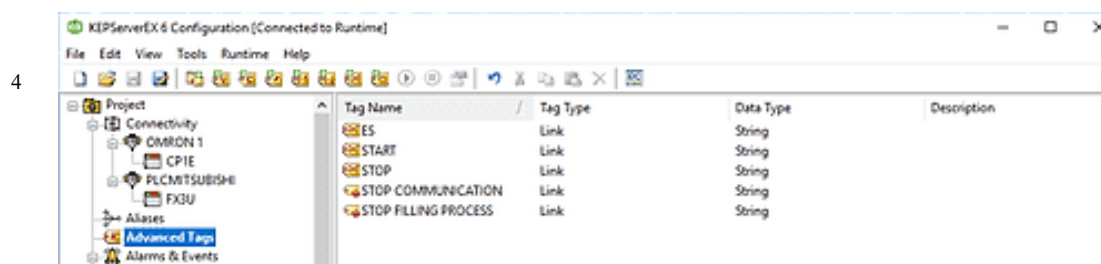


Table 2. Main program of both PLC Mitsubishi and PLC Omron

	PLC OMRON (filling station)		PLC Mitsubishi (unscramble station)
LD 200.00	<i>'continued</i>	<i>''continued</i>	LD X002
OR 200.03	AND W0.05	TIM 005 #8	OUT M1
OUT W0.00	OUT W0.06	LD W0.14	LD M1
LD 200.01	TIM 000 #10	CNT 006 #5	OUT M5
OR 200.04	LD T000	LD C006	LD X000
OUT W0.01	AND W0.06	OR 200.07	OUT M2
LDNOT 0.00	OUT W0.07	ANDNOT W0.14	LD M2
OUT 200.09	LD W0.03	OUT 200.07	OUT M6
LD 0.02	OR W0.08	LD W0.05	LD M3
OUT W0.02	AND W0.07	ANDNOT W0.06	OUT M7
LD LR	OUT W0.08	OR W0.13	LD M5
OUT W0.03	TIM 001 #20	OUT 100.00	OR M8
LD 0.04	LD T001	LD W0.07	ANDNOT M6
OUT W0.04	OR W0.09	ANDNOT W0.12	ANDNOT M7
LD 0.01	AND W0.08	OUT 100.01	ANDNOT M10
OUT 200.05	OUT W0.09	LD W0.12	OUT M8
LD 200.02	TIM 002 #60	ORNOT W0.07	LD M8
OR 200.05	LD T002	OUT 100.02	OUT Y000
OUT W0.14	OR W0.10	LD W0.08	LD M8
OR W0.15	AND W0.09	ANDNOT W0.11	OUT Y001
ANDNOT W0.01	OUT W0.10	OUT 100.03	LD M8
ANDNOT 200.07	TIM 003 #60	LD W0.11	ANDNOT M0
ANDNOT 200.09	LD T003	ORNOT W0.08	ANDNOT M9
OUT W0.15	OR W0.11	OUT 100.04	OUT Y002
LD W0.15	AND W0.10	LD W0.09	@LD X001
AND W0.04	OUT W0.11	ANDNOT W0.10	OR M0
OR W0.05	TIM 004 #20	OUT 100.05	ANDNOT T001
ANDNOT T005	LD T004		OUT M0
ANDNOT W0.01	OR W0.12		TIM 000 #10
ANDNOT 200.07	AND W0.11		LD M0
ANDNOT 200.09	OUT W0.12		OR M9
OUT W0.05	LD W0.04		AND T000
LD W0.04	AND W0.12		OUT M9
@AND W0.02	OUT W0.13		TIM 001 #15
OR W0.06	<i>''continued</i>		LDNOT M0
<i>'continued</i>			OR M9
			OUT Y003

Table 3. Sample of data transfer between PLC Mitsubishi and Omron using KEPServerEX6

PLC Mitsubishi		data													
time	event	length													
13:28:39.416	TX	13	02	45	30	30	38	38	30	30	32	03	44	41	
13:28:39.447	RX	8	02	32	32	30	31	03	43	38					

PLC Omron		data																				
time	event	length																				
13:28:39.436	RX	32	40	30	30	46	41	30	30	43	30	30	30	30	30	31
13:28:39.462	RX	11	30	30	30	30	35	33	33	2A	0D											

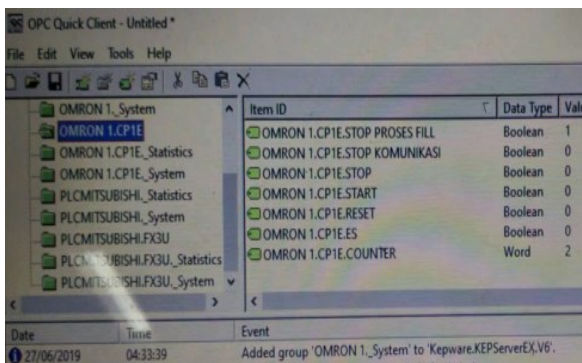


Figure 4. The counter counts the value 2 of 5

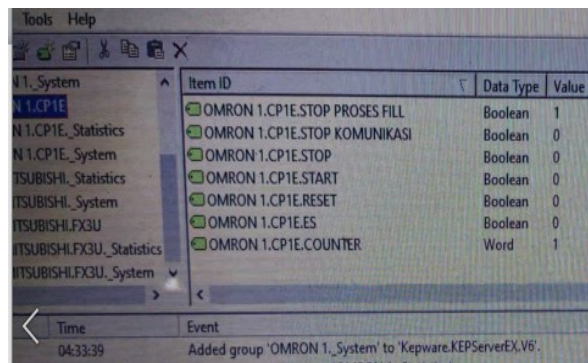
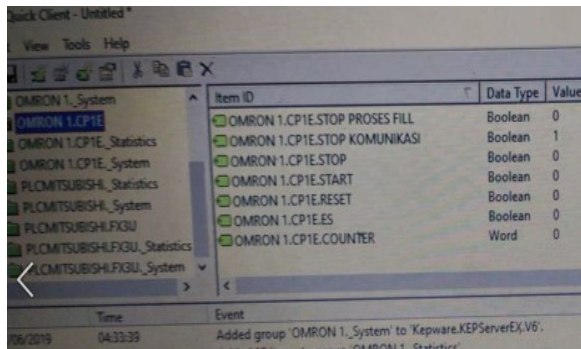


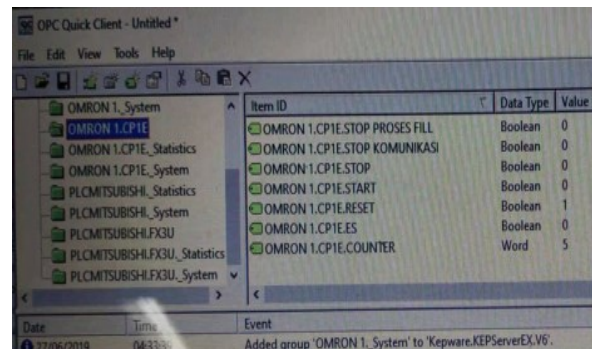
Figure 5. The counter counts the value 1 of 5



The screenshot shows the OPC Quick Client interface with a tree view on the left and a data table on the right. The table lists various PLC items and their current values.

Item ID	Data Type	Value
OMRON 1.CP1E.STOP PROSES FILL	Boolean	0
OMRON 1.CP1E.STOP KOMUNIKASI	Boolean	1
OMRON 1.CP1E.STOP	Boolean	0
OMRON 1.CP1E.START	Boolean	0
OMRON 1.CP1E.RESET	Boolean	0
OMRON 1.CP1E.E5	Boolean	0
OMRON 1.CP1E.COUNTER	Word	0

Figure 6. Stop communication is active, valued 1



The screenshot shows the OPC Quick Client interface with a tree view on the left and a data table on the right. The table lists various PLC items and their current values.

Item ID	Data Type	Value
OMRON 1.CP1E.STOP PROSES FILL	Boolean	0
OMRON 1.CP1E.STOP KOMUNIKASI	Boolean	0
OMRON 1.CP1E.STOP	Boolean	0
OMRON 1.CP1E.START	Boolean	0
OMRON 1.CP1E.RESET	Boolean	1
OMRON 1.CP1E.E5	Boolean	0
OMRON 1.CP1E.COUNTER	Word	5

Figure 7. Reset is active, valued 1

3.3. Discussion

In several previous papers on OPC Server, the usefulness of OPC Server was discussed. Among them are papers written by Zhang [5], Şahin [14], and Li [15] which use OPC Server to monitor PLC, but only one manufacturer is used. Furthermore when there is a demand to integrate two or more different PLC manufacturers, it still needs to be considered. Other paper, Toyland [1] and Wenyan [16] apply the OPC Server for the application of water systems. In the example this application provides instructions on how OPC Server can be used in applications, consequently the usefulness of OPC Server becomes less optimal if only for this purpose. The use of web-based OPC Server in [2] and [8] has provided alternative solutions, but this is quite difficult if applied because it demands a reliable cloud system. Lieping [17] and Zhang [18] offer the real-time communication between MATLAB and PLC, for the use of OPC we must make a program using Visual Basic 6,0 [7] or using MATLAB [19]. Another paper proposed the OPC connectivity-based approach to remote monitoring and control [20] and [21]. Vadi & Bayindir [22] developed an OPC and PLC based remote-access laboratory. Zhu, et.al. [23] used OPC to design the information acquisition for field equipment. Mahmoud, et.al. [24] using OPC to support in advanced process control. Data communication is one of the significant part where the nodes send data towards the destination, some are based on real-time and some are on normal data communication [25]. The use of KEPServerEX6 provides a relatively easy solution for integrator technician who will combine the controls of several other PLCs so that can communicate each other without difficulty with factory driver problems.

In this paper OPC Server has been designed and developed for integration of controls between two PLCs from different manufacturers to communicate. The OPC Server used in this study came from the KEPCWARE, namely KEPServerEX6 [13]. KEPServerEX6 is an application program that uses OPC Server technology. With the help of KEPServerEX6, integration issues two or more different control platforms can be overcome from different manufacturers can be communicated together. So that the ability and capacity of the OPC Server can be used more leverage than just to control one device or one PLC manufacturer.

The obstacle is that in KEPServerEX6 there is no choice for HMI Omron etc. Furthermore the PLC that we use is limited since does not have a serial port. Therefore the mini-USB connection can only be used for one RS232 connection to USB-PC. By the way, this will be easily overcome if the choice of PLC used considers the existence of the ports needed for RS232 and serial connections. The rest of the use of KEPServerEX6 is quite strong and meets the needs of integrating two or more PLCs (controllers) different platform.

4. CONCLUSION

The objective of this study was tried to answer the question how to communicate among different devices and controllers in industrial process. The implementation of OPC Server technology has concluded that communication between PLC Mitsubishi and PLC Omron can be done even though both have its own driver software for controlling the field application devices. The observation from this study finds that using the KEPServerEX6 (OPC server), PLC Mitsubishi and PLC Omron can communicate each other by send and/or receive the data. This work has come up with the finding by using KEPServerEX6 (OPC server) it is possible to realize the integration among the different devices and controllers which is commonly find in the industrial processes. Further work needs to be done to establish the human machine interface in this system, so that the integration and communication among the controller can be monitor and operate easily.

REFERENCES

- [1] Toylan H., Kuscü H., "A research on SCADA application by the help of OPC server for the water tank filling system," *Sci Res Essays*, vol. 5, no. 24, pp. 3932–3938, 2010.
- [2] Şahin C., Bolat E. D., "Development of remote control and monitoring of web-based distributed OPC system," *Comput Stand Interfaces*, vol. 31, no. 5, pp. 984–993, 2009.
- [3] Hong X., Jianhua W., "Using standard components in automation industry: A study on OPC specification," *Comput Stand Interfaces*, vol. 28, no. 4, pp. 386–395, 2006.
- [4] Kominek D., "OPC: The Ins and Outs to What It's About," 2009. [Online]. Available: <http://www.matriconopc.com>. Accessed: 13 Feb 2019.
- [5] Zhang Y., Liu M., Zhao F., "The design and implementation of PLC monitoring system based on OPC," *Applied Mechanics and Materials*, vol. 278–280, pp. 930–935, 2013.
- [6] Guo Z. X., Xie X. Q., Ni Z. G., "The application of OPC DA in factory data acquisition," *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2, pp. 209–212, 2012.
- [7] Guo H., Zhi D., "Design of several OPC servers communication system," *2011 International Conference on Electric Information and Control Engineering*, pp. 317–319, 2011.
- [8] Abbas H. A., Mohamed A. M., "Review on the Design of Web Based SCADA Systems Based on OPC DA Protocol," *International Journal of Computer Networks (IJCN)*, vol. 2, no. 6, pp. 266–277, 2011.
- [9] Schwarz M. H., Borsok J., "A survey on OPC and OPC-UA: About the standard, developments and investigations," *2013 24th Int Conf Information, Commun Autom Technol ICAT 2013*, 2013.
- [10] Assegie T. A., Nair P. S., "A review on software defined network security risks and challenges," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 6, pp. 3168–3174, 2019.
- [11] Foundation O., "Data Access Automation Interface Standard Version 2.02," 4 February 1999. [Online]. Available: <http://www.opcfoundation.org>. Accessed: 13 Feb 2019.
- [12] Foundation O., "Data Access Custom Interface Standard Version 2.05A June 28 2002," Austin, Texas: OPC Foundation, 2002. [Online]. Available: <http://www.opcfoundation.org>. Accessed: 13 Feb 2019.
- [13] P. T. C. Kepware, "Kepserverex," Portland, ME 04101: Kepware, 2018. [Online]. Available: www.kepware.com. Accessed: 13 Feb 2019.
- [14] Şahin C., Bolat E. D., Kraçor M., "Development of OPC Based SCADA System Using SIEMENS CPU 224 for Educational Purposes," *NWSA-Technological Appl Sci*, vol. 7, no. 2, pp. 17–27, 2012.
- [15] Li D. H., Li H., Zhang K. L., "Realization of Real-Time Data Exchange between OPC Client and Multiple Remote Servers," *Appl Mech Mater*, vol. 385–386, no. 1655–1658, 2013.
- [16] Wu W., Gao J., Yuan Y., Zhao H., Chang K., "Water Distribution Network Real-Time Simulation Based on SCADA System using OPC Communication," *2011 Int Conf Networking, Sens Control ICNSC 2011*, no. April, pp. 410–415, 2011.
- [17] Lieping Z., Aiqun Z., Yunsheng Z., "On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology," *Proceeding of the 26th Chinese Control Conference*, Zhangjiajie, Hunan, China, pp. 545–548, 2007.
- [18] Zhang Y. J., "Research and Application of OPC Network Communication in Configuration Software," *Adv Mater Res*, vol. 774–776, pp. 1774–1777, 2013. doi: <http://dx.doi.org/10.3182/20090520-3-KR-3006.00041>. Accessed: 13 Feb 2019.
- [19] Lu Y., Juyang L., "Implementation of the trinity of the control system based on OPC," *Int J Res Eng Sci*, vol. 2, no. 11, pp. 9–13, 2014.
- [20] Tan V. Van, Yoo D. S., Yi M. J., "An approach to remote monitoring and control based on OPC connectivity," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, IFAC, pp. 259–262, 2009.
- [21] Jing C., Wang Y., Song X., "Development of an OPC server for remote monitoring and control based on GPRS networks," *Proceedings-IEEE 2011 10th International Conference on Electronic Measurement and Instruments*, ICEMI 2011, pp. 222–226, 2011.
- [22] Vadi S., "Development of an OPC and PLC Based Remote-Access Laboratory: A Synchronous Motor Control Experiment," *Int J Appl Math Electron Comput*, vol. 3, no. 3, pp. 172–177, 2015.
- [23] Zhu M., Lu S., Du H., Zhu Z., "Design and application of field equipment information system based on OPC," *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, China: IEEE, pp. 7422–7426, 2017.
- [24] Mahmoud M. S., Sabih M., Elshafei M., "Using OPC technology to support the study of advanced process control," *ISA Trans*, vol. 55, pp. 155–167, 2015. doi: <http://dx.doi.org/10.1016/j.isatra.2014.07.013>, Accessed: 13 Feb 2019.
- [25] Mona J., "Data communication in internet of things: Vision, challenges and future direction," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 5, pp. 2057–2062, 2018.