POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

Design, application and analysis of an OPC-based SCADA system

OPC tabanlı SCADA sisteminin tasarımı, uygulaması ve analizi

Yazar(lar) (Author(s)): *Okan DUYMAZLAR[1], Dilşad ENGİN[2]*

ORCID[1]: 0000-0002-1327-7493

ORCID[2]: 0000-0003-0159-275X

To cite to this article: Duymazlar O., Engin D., "Design, application and analysis of an OPC-based SCADA system", ***Journal of Polytechnic***, 26(2): 991-999, (2023).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Duymazlar O., Engin D., "Design, application and analysis of an OPC-based SCADA system", ***Politeknik Dergisi***, 26(2): 991-999, (2023).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: **10.2339/politeknik.1029629**

Design, Application and Analysis of an OPC-based SCADA System

Highlights

- *successfully operated OPC-based SCADA system in runtime*
- *functional and time-based analyzes performed for the physical and SCADA system*
- *different controllers using various industrial communication protocols successfully talked to each other*
- *flexible, extensible, and hardware-independent design advantages provided by OPC-based systems*
- *integration of communication protocol independent OLE controllers with PLCs*

Graphical Abstract

An OPC-based SCADA system was designed for an electro-pneumatic press system with open-source controllers communicating through different protocols; network infrastructure and physical connections were commissioned and tested in runtime and the controllers successfully talked to each other through the system. We commissioned OPC and SCADA software by connecting the hardware with a star topology and performed functional and time-based analyzes.

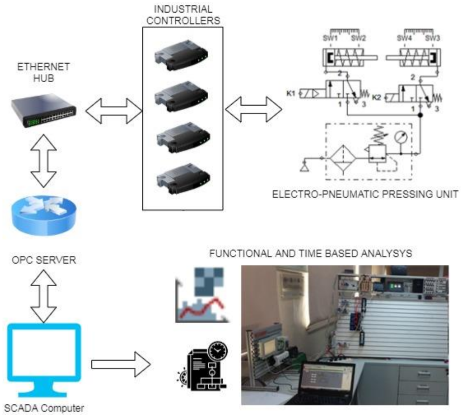

Figure. Graphical abstract

Aim

The aim of our study is to design, implement, and empirically test the physical and software model of an OPC-based SCADA system comprising multiple industrial controllers of various commercial brands with different industrial communication protocols that operates an electro-pneumatic press prototype.

Design & Methodology

Each sub-task of the press was assigned to different industrial controllers, and the network infrastructure of the SCADA system was created with a star topology. The connectivity and interoperability for the networked system comprising multiple devices with different industrial communication protocols was successfully provided.

Originality

Unlike the literature, design, implementation and examination of the SCADA system in terms of physical industrial unit, network infrastructure and software were demonstrated, and advantages and disadvantages of cross-tagging used in OPC server software in terms of runtime and functionality was investigated.

Findings

OPC-based SCADA system was tested in real time and the controllers successfully talked to each other through the system.

Conclusion

The hardware and software structural details, possible malfunctions and data transmission speeds of the system were analyzed and shared.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Design, Application and Analysis of an OPC-based SCADA System

Araştırma Makalesi / Research Article

Okan DUYMAZLAR[1], Dilşad ENGİN[2*]

[1]Mechatronic Engineering MSc., Ege University, Türkiye

[2]Department of Control and Automation Technology, EGE MYO, Ege University, Türkiye

(Geliş/Received : 30.11.2021 ; Kabul/Accepted : 25.02.2022 ; Erken Görünüm/Early View : 15.03.2022)

ABSTRACT

The main goal for our study is the design, implementation, and empirical test of the physical and software model of an OPC-based SCADA system comprising multiple industrial controllers of various commercial brands. In this study, an electro-pneumatic press prototype was prepared with four different PLCs, an open-source controller and an operator panel which have different industrial communication protocols. Each sub-task of the pressing unit was assigned to different industrial controllers, and the network infrastructure of the SCADA system was created with a star topology and taken into operation. The connectivity and interoperability for the designed networked system comprising significant number of devices with different industrial communication protocols was effectively provided. OPC-based SCADA system was tested in runtime mode and the controllers successfully talked to each other through the system. The hardware and software structural details, possible malfunctions and data transmission speeds of the system were analyzed and shared.

Keywords: SCADA system, OPC, multiple controllers, industrial communication, connectivity.

OPC Tabanlı SCADA Sisteminin Tasarımı, Uygulama ve Analizi

ÖZ

Çalışmamızın ana amacı, çeşitli ticari markaların çoklu endüstriyel denetleyicilerini içeren OPC tabanlı bir SCADA sisteminin fiziksel ve yazılım modelinin tasarımı, uygulanması ve deneysel testinin yapılmasıdır. Bu çalışmada, farklı endüstriyel haberleşme protokollerine sahip dört farklı PLC, açık kaynaklı denetleyici ve operatör paneli ile elektro-pnömatik pres prototipi hazırlanmıştır. Pres ünitesinin her bir alt görevi farklı endüstriyel denetleyicilere atanmış ve SCADA sisteminin ağ altyapısı bir yıldız topolojisi ile oluşturularak devreye alınmıştır. Farklı endüstriyel iletişim protokollerine sahip önemli sayıda cihazdan oluşan tasarlanan ağ bağlantılı sistemin bağlanabilirliği ve birlikte çalışabilirliği etkin bir şekilde sağlanmıştır. OPC tabanlı SCADA sistemi çalışma zamanı modunda test edilmiş ve denetleyiciler sistem üzerinden birbirleriyle başarılı bir şekilde haberleşmiştir. Sistemin donanım ve yazılım yapısal detayları, olası arızaları ve veri aktarım hızları analiz edilerek paylaşılmıştır.

Anahtar Kelimeler: SCADA sistemi, OPC, çoklu denetleyiciler, endüstriyel haberleşme, bağlanabilirlik.

1. INTRODUCTION

Today's industrial systems are faced with increasing requirements to respond promptly and effectively to rapid change in market demands. To meet these requirements, new technologies and concepts such as Internet of Things (IoT), Big Data Analytics are being widely used in industrial environments [1]. Industrial systems and processes need to be continuously monitored and supervised to meet the contemporary demands and competition of the market and the customer's requirements.

Operational technology (OT) monitors and controls industrial process and industrial equipment including controllers, sensors and actuators at shop floor level

whereas information technology (IT) comprises software management systems as supervisory control and data acquisition systems (SCADA), enterprise resource planning (ERP), and manufacturing execution systems (MES) [2]. Concordantly, OT in the physical layer and IT in the information part are integrated in order to collaborate. The OT/IT connectivity and interoperability are functional requirements for complex, large-scale, and extensive networked systems comprising a substantial number of devices with different industrial communication protocols [2].

In automation and control systems, data acquisition, exchange, and processing are accomplished in a distributed manner between diverse data sources and users. SCADA systems have become almost mandatory and started to be widely used due to fact that the design and implementation of many industrial applications requires sub-tasks, increasing the need for distributed controllers and smart sensors. SCADA systems meet the need for remote control and visualization of automatic control processes, data acquisition from the field and analysis of the collected data. The main application area

*Sorumlu yazar(Corresponding Author)
e-posta : dilsad.engin@ege.edu.tr

of SCADA systems is supervisory control and monitoring of the dispersed assets [3]. Programmable Logic Controllers (PLC) take place as the control device in industrial applications due to their reliability and durability. PLCs are broadly utilized in the production processes to coordinate the complicated obligations and tasks which include protection monitoring, energy management, gadget control, and automatic manufacturing lines.

Developments in Industry 4.0 brought automation and control to a different extent in many areas such as the field of renewable energy where it is targeted to lower the installation cost ratios with the integration of renewable energy sources utilizing PLC and SCADA based automation systems [4], to accomplish a self-supported and zero-carbon operation in a smart micro grid that combines Renewable Energy Source (RES) that encompasses a photovoltaic generator system, a wind generator system and hydrogen supervised by a system that comprises a PLC and a Human Machine Interface (HMI), and connectivity is supported by OPC server [5].

Interconnection through networks comprising of devices with different industrial protocols require reliable and secure communication. Today, heterogeneous availability of controller technologies with numerous architectures, and different protocols aim to accomplish a flawless industrial communication. OPC is an open platform for reliable and secure communication between the industrial devices as shown in Fig. 1. Classical OPC, so-called Object Linking and Embedding for Process Control (OLE for Process Control) is for abstracting specific protocols associated to PLCs into a standardized communication interface for HMI and SCADA systems that will convert between generic-OPC read/write requests and device-specific requests [6].

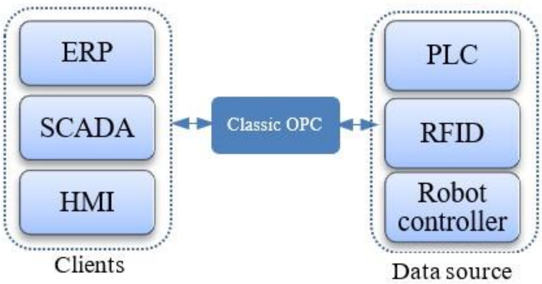

Fig. 1 OPC-based communication scheme [2]

The Open Platform Communication Unified Architecture (OPC UA), released in 2008, is a service-oriented architecture (SOA) platform that accomplishes secure platform-independent communication between devices, and has an extensible ability to add new features to an existing application [7]. OPC UA is the operative interoperability standard for data exchange through devices of diverse manufacturers, providing industrial controller-to-controller connectivity [8].

OPC-based SCADA systems are often preferred for control-oriented applications, security, maintenance, and safety systems, and for monitoring field applications carried out by human operators and intervening when necessary. Aleksandrov et al. [9] used OPC server over Local Area Network with the system that has multiple PLCs which are used to control DC motors and pneumatic actuators. Barsoum and Chin [10] worked on control and monitoring of a system comprising of PLCs of different vendors connected through an OPC server. Sangeetha et al. [11] used OPC-based SCADA system communicating over internet connection to monitoring and tuning PID parameters for controlling the flow rate a basic cascade water level control system. Proposed system included AB Micrologix 1200 PLC as master controller and NI-OPC server for SCADA. They experimentally investigate the delay effect of the internet connection for this application and state that the difference is insignificant. Toylan and Kuşçu [12] designed an OPC-based SCADA system for a water tank filling system with heat exchanger controlled with PWM signals through Siemens S7-200 PLC. They designed and implemented SCADA interface with Siemens WinCC and used Microsoft's OLE/COM based OPC server to send PWM parameters to PLC. Üstünsoy and Sayan [13] aimed to provide monitoring and reporting, and energy management dealing with energy quality and energy efficiency to reduce energy consumption, reduce the costs and increase the network stability with SCADA system supported by PLC. This system was designed and constructed using Schneider Electric TM241CE40T/U PLC where the data was supplied from the energy analyzers through MODBUS RTU protocol, the data transferred to SCADA server via MODBUS TCP protocol was monitored by the designed graphical interface implemented by VijeoCitect software.

Chamorro-Atalaya et al. [14] designed a SCADA system that works with the KEPServerEX 6 OPC Server software to control and monitor a prototype automated fire-fighting system. Experimentally investigated system included a single PLC (Siemens S7-1200, CPU 1212C) programmed with ladder diagrams for only digital feedbacks. İşbilen and Konar [15] designed a SCADA system which is both used for monitoring AIRCRAFT stall warning system and controlling external power unit for ground maintenance. They used Omron CP1L PLC (in simulation mode) and prepared the SCADA interface with Cx-Supervisor Developer software for the examined system. Nicola et al. [16] developed an industrial LabVIEW application for SCADA system and applied OPC client-server communication for monitoring experimental indicators in controllers for a pneumatic system [17]. In [18], PLCs were used for controlling the air compressors' parameters for operation whereas SCADA application panels were employed to monitor these parameters, and the reliability of the OPC UA based SCADA system in multiple networks was proven. Hadi and Sallom [19] used OPC server and SCADA system on the same computer and accessed the parameters of the PLC that controlled the pneumatic equipment.

In this work, we designed an OPC-based SCADA system to monitor and control a prototype electro-pneumatic press system. The system has four standalone industrial controllers (Beckhoff CP2689, Schneider Electric Modicon M241CEC24R, and Siemens S7 1200 CPU 1214C PLCs, an OLE Modbus TCP module) that are assigned to subtasks and an HMI panel (Schneider HMI S5T) to control and monitor the pressing operation. We preferred to use CitectSCADA 2015 software for the supervisory control and data acquisition part of the system with monitoring features and KEPServerEX OPC UA platform as OPC server software. SCADA server and KEPServerEX OPC UA platform were both installed and run on the same computer and physical connection of the controllers to the server were established over star topology LAN connection. We also added a Wi-Fi router to the system for redundancy and mobility purposes of server computer. Proposed system carried out communication and data transfer between PLCs, OPC Server and SCADA computer over variable tags instead of the IEC 61131-3 standard addressing. We implemented the entire system with physical and software parts, accomplished runtime tests and shared structural details, possible malfunctions, and data transmission speeds of the system.

2. ARCHITECTURE OF THE SYSTEM

2.1. Electro-pneumatic Press System and Controllers

The primary elements of the electro-pneumatic press system are two vertically aligned single-acting pneumatic cylinders with 3/2 directional control valves and four independent roller limit switches to supply feedback to controller for the cylinder position. The simplified scheme of the electro-pneumatic press system is shown in Fig. 2(a), and the electro-pneumatic circuit diagram of the system in Fig. 2(b) is designed by FESTO's FluidSIMP 4.0® pneumatic circuit simulation software.

Structural design of the control and communication of the system is shown in Fig. 3 including the industrial controllers. In the field application, the system has two safety switches connected to Beckhoff CP2689 Industrial PC (iPC) and must be initially turned on and then pressing operation was executed with HMI touch panel connected to the system over Schneider Electric M241CEC24R PLC with Modbus serial connection. Schneider PLC's two digital outputs were also assigned to trigger 3/2 directional control valves to operate single acting cylinders synchronously. Four discrete roller type limit switches connected to Siemens S7-1200 PLC for position feedback of the vertically aligned cylinders and Modbus OLE controller was used to count each five consecutive operations and indicate with embedded LED display. Although, a single standalone industrial controller may meet the requirements for the press system, basic tasks were assigned to multiple controllers

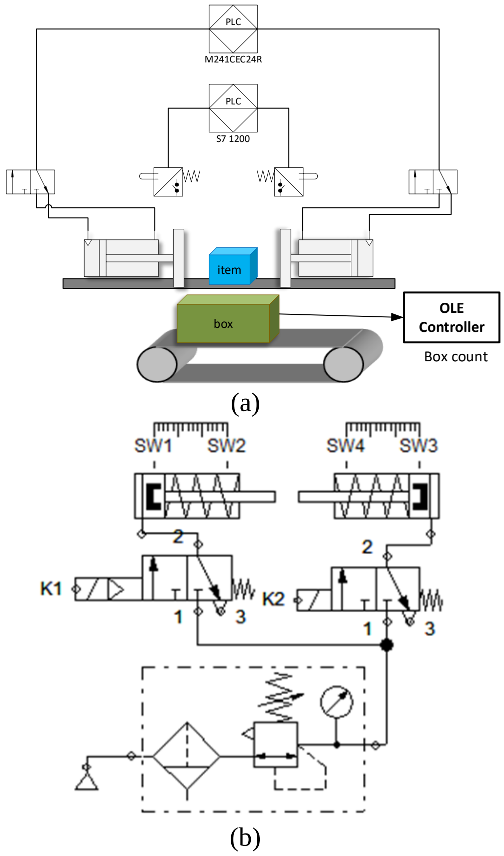

Fig. 2 (a) Simplified scheme of the electro-pneumatic

for demonstrating and analyzing OPC server with multiple controllers from various brands.

Communication between the controllers was not via physical connections, but over the OPC server as seen in Fig. 3. System administrator has full authorization formonitoring of the electro-pneumatic press system and its intervention can be carried out without the need for any permission of the operator who controls the pressing unit via HMI. Industrial controllers can also be controlled independently of each other over SCADA system using OPC Server connection.

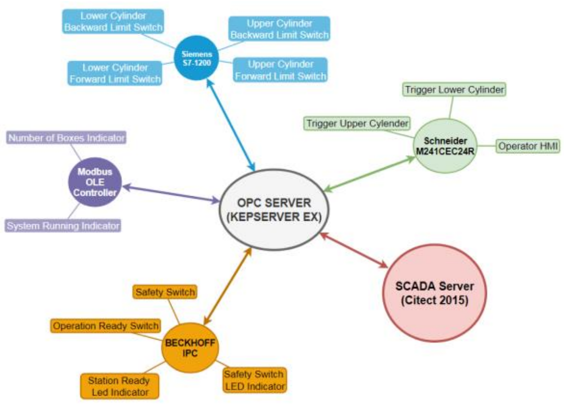

Fig. 3 Structural design of the pressing unit including

2.2. System Network Topology

The designed SCADA system was built using star topology. A simple Ethernet HUB with its 8 Ethernet ports and standard CAT5E crossover cables were used to interconnect the controllers, OPC Server and SCADA computer which form the network backbone as seen in Fig. 4.

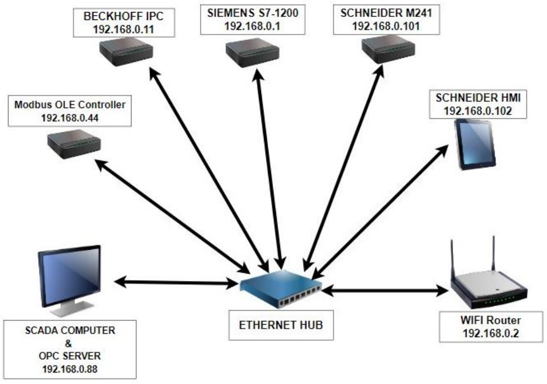

Fig. 4 Network topology of the SCADA system

A Wi-Fi router has also been added to ensure redundancy of the physical connection between the HUB and servers and to increase mobility. C-class fixed IP addresses in the same subnet were used for address configuration of PLCs, OLE controller and Wi-Fi router in the system. Since KEPServerEX, the OPC server software, runs on the SCADA computer, both share the same IP address.

3. OPC SERVER AND INDUSTRIAL COMMUNICATION PROTOCOLS OF THE DEVICES

OPC servers are used for communication between machines and devices from different manufacturers and for bridging between the devices with different protocols. OPC combines industrial communication protocols developed by technology developers such as companies or established organizations under one roof to enable communication between the industrial devices. As we used devices that exchange data in various vendor-specific formats, implementing an OPC server was a necessity.

The OPC server software transmits data and signals it collects from different industrial communication protocols to the main client or main controller, which is mostly a SCADA computer, through a single channel.

This system, in which the OPC server is placed in the center and each network element exchanges data with the OPC server independently of each other, is inspired by the star topology in terms of its structure.,

3.1. OPC Server Software and Configuration

In our study, Kepware's KEPServerEX OPC server software was used to exchange data between the SCADA computer and the industrial controllers. The main reason why OPC server software was preferred was that it had OPC Unified Architecture (OPC UA) support as well as OPC Classic and OPC DA (OPC Data Access) features for legacy devices.

OPC Classical and OPC DA and are both based on the Component Object Model (COM), which is the standard component of Microsoft Windows operating systems. On the contrary, OPC UA-based servers are platform independent and thus allow the implementation SCADA project to be run or integrated on different servers regardless of the operating system.

KEPServerEX software runs on the same computer which host the SCADA software. The computer works on Windows 7 Ultimate 64-bit OS with standard firewall.

A communication channel was built for each different protocol in the OPC server software, and industrial controllers using the same communication standard were grouped with the help of channels. Each individual controller in the SCADA network is identified by a unique ID. Thus, when a device in the SCADA network is to be accessed, a hierarchy was created by first using the channel pointing to the communication protocol and then using the ID of the device. Channels where industrial controllers are grouped according to the communication protocols, Device IDs and protocol-specific access addresses are shown in Table I.

3.2. Data Exhange through Devices

The hierarchy shown in the Fig. 5 is used to access the input/output (I/O) and memory addresses of controllers in the SCADA network. In this figure, an example for the hierarchical operation for BECKHOFF iPC is shown with red arrows. In order to access the desired address of this controller, first the channel is created in the OPC server software, then the unique address of the controller is defined under the channel, and then the tag is assigned to the relevant address.

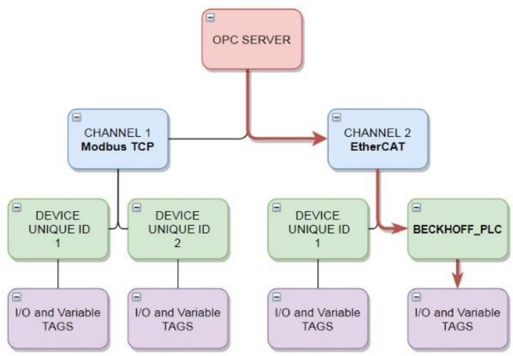

Fig. 5 Tag hierarchy for accessing controllers' I/O and addresses

Table I. Industrial controllers defined in OPC Server

Controller	Industrial Communication	Port	OPC Channel	OPC Device Unique ID	Address
Schneider M241CEC24R	Modbus TCP	502	Channel_1	PLC1_M241	<192.168.0.101>.0
Modbus OLE Controller	Modbus TCP	505		DEVICE1_OLE	<192.168.0.44>.1
Siemens S7-1200	Siemens TCP/IP Ethernet	102	Channel_2	PLC2_S71200	192.168.0.1
Beckhoff CP2689	Beckhoff EtherCAT	801	Channel_3	PLC3_BECKHOFF	192.168.0.11.1.1
Schneider HMIS5T	Modbus Serial	-		DEVICE2_HMI	-

With the purpose of reading and writing to the physical I/Os and memory words of the controllers communicating with Modbus TCP/IP, identification is performed with Modbus addresses. For example, the first physical input bit value (%IX0.0) of Schneider Electric M241CEC24R and Modbus OLE controllers are defined by Modbus Register addresses 100001 and memory bit %MX0.0 with 40001.0 addresses. M241CEC24R PLC program for the assigned sub-task is in Fig. 6. There is no need to make any cross-reference definition for Siemens S7-1200 PLC via KEPServerEX. For this reason, the addresses in the PLC program created with the TIA Portal software were used directly in KEPServerEX without making any changes. The address of the first physical output (I0) in the PLC program is used in the same way on the KEPServerEX side. In the EtherCAT channel created for Beckhoff iPC, addresses can be downloaded directly from the industrial controller. For this reason, there is no need to make any special definitions for Beckhoff iPC [20].

The OPC Server software used in the study also allowed “Advanced Tags” to exchange data or perform conditional triggers between each other without making any changes to the PLC program running on any controller. In this study, we used connection ability of the tags with Modbus OLE Controller and Schneider M241CEC24R PLC to forward the counted number of the pressed items of the system.

The flexibility of SCADA systems designed with the help of this feature, which is unique to OPC software, can be increased. When a task that needs to be added to the system and a controller to control this task is needed, the system can be expanded without any software changes, only with OPC Server Advanced Tags.

4. SCADA SYSTEM

A computer with built-in LAN and wireless adapters and running both OPC Server and SCADA Server software was used as SCADA computer. We used CitectSCADA 2015 software to configure SCADA computer, design SCADA control interfaces, and CitectSCADA runtime for testing and monitoring the system.

4.1. SCADA Computer Configuration

During configuration, we chose the SCADA computer as Standalone since we used one SCADA computer in the designed electro-pneumatic press system. Computer configuration settings were completed through the Express Setup Wizard, which is included in the CitectSCADA 2015 software and guides the system administrator about installation, and server and control client tasks were assigned to the SCADA computer.

4.2. SCADA Interface Screen

The SCADA control interface shown in Fig. 7 was designed using Citect Graphics Builder from CitectSCADA 2015. Through the designed SCADA screen, runtime monitoring of the entire system and intervention when necessary are provided.

4.3. Building SCADA and OPC Server Integration

In SCADA systems, SCADA computer has the most advanced authority. But when it’s come to the OPC-based SCADA system and the network structure perspective, OPC Server also can be seen as another server since it has the authority to access and interfere with all controllers in the system. However, when the authority advantages are compared, the SCADA server has dominance over the OPC server, and for this reason, although the OPC server software is at the center in the network topology, it is configured as a client of the SCADA computer.

```
//automatic mode - control operations via hmi
IF (xscadaMode=FALSE AND xbeckhofenable1=TRUE AND xbeckhofenable2=TRUE)THEN
        IF (xi_hmi_button_fwd=TRUE)THEN
                xq_m241_fwd1:=TRUE;
                xq_m241_fwd2:=TRUE;
                temp_var:=TRUE;
        END_IF
                IF(xi_hmi_button_bwd=TRUE AND temp_var=TRUE)THEN
                        xq_m241_fwd1:=FALSE;
                        xq_m241_fwd2:=FALSE;
                        temp_var:=FALSE;
                        item_count:=item_count+1;
                        IF(item_count>=6)THEN
                                box_count:=box_count+1;
                                //variable query on Modbus_ole device
                                IF(box_count=1 OR box_count=2)THEN
                                        m_ole_box_count:=box_count;
                                END_IF
                                IF(box_count=3)THEN
                                        m_ole_box_count:=box_count+1;
                                END_IF
                        IF(box_count=4)THEN
                                m_ole_box_count:=box_count*2;
                        END_IF
                        // variable query on Modbus_ole device
                                item_count:=1;
                                IF(box_count>=5)THEN
                                        box_count:=1;
                                        m_ole_box_count:=box_count;
                                END_IF
                        END_IF
                END_IF
END_IF
//manual mode – all controls on scada server-> all variables are supervised by scada.
IF(xscadaMode=TRUE)THEN
        alarm:='ADMIN MODE ACTIVE';
        IF(xscada_button_fwd1=TRUE)THEN
                xq_m241_fwd1:=TRUE;
        END_IF
                IF(xscada_button_fwd2=TRUE)THEN
                xq_m241_fwd2:=TRUE;
        END_IF
                IF(xscada_button_bwd1=TRUE)THEN
                xq_m241_fwd1:=FALSE;
        END_IF
                IF(xscada_button_bwd2=TRUE)THEN
                xq_m241_fwd2:=FALSE;
        END_IF
END_IF
```

Fig. 7 Schneider Electric M241CEC24R PLC Codes

As a matter of fact, the OPC Server software does not have the rights to change data in the SCADA software. SCADA software has monitoring authorization and management of the entire system, including the OPC Server. The most authoritative system unit in the planned system is SCADA, and it employs OPC Server as a communication medium. Fig. 8 reveals this approach where the OPC software acts only as a bridge and the

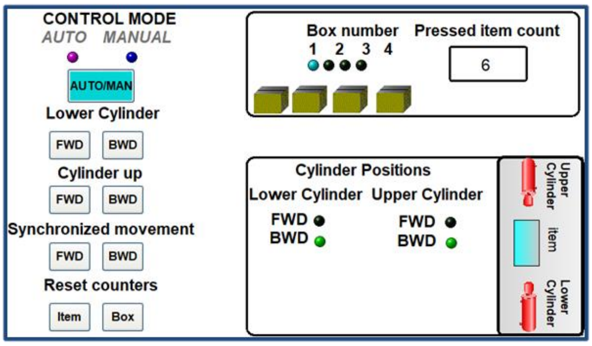

Fig. 6 SCADA interface of the pressing unit

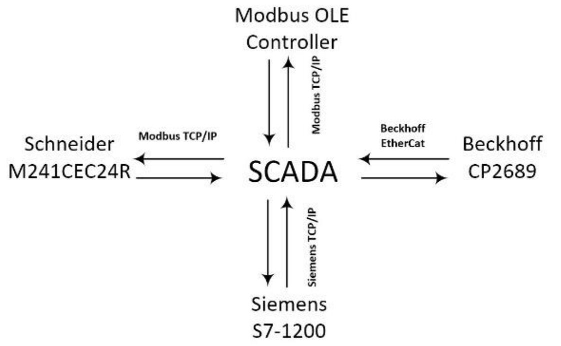

Fig. 8 SCADA system with Industrial Controllers and

SCADA computer accesses the industrial controllers through its own protocols.

For the SCADA system, we need to define the I/O devices and protocols for each controller to communicate with the industrial controllers and the field devices. However, for the OPC-based SCADA systems, there is no need to configure I/O devices for each controller but just a single I/O device uses OPC protocol. In this study, we created an I/O device and selected the OPC driver under the OPC Foundation library of CitectSCADA software.

As in this study, SCADA software and OPC Server software run on the same computer, and if the OPC Server software is installed before the SCADA project is created, the name of the OPC software is used as the I/O address in a case sensitive way. If the OPC Server software runs on a different network device, then the IP address of the running device should be used. Since OPC and SCADA worked on the same computer in this system and KEPServer's OPC software was used, we defined the I/O address as “Kepware.KEPServerEX.V6”. SCADA computers using CitectSCADA need also pre-defined equipment and equipment type, but OPC Server eliminates this requirement too. OPC-based SCADA systems, as we designed in this study, can monitor, or supervise any of the field device using the “ChannelName.UniqueDeviceID.TagName” hierarchy. For instance, in this study Schneider M241CEC24R first digital output was manipulated with the “ModbusChannel.PLC1.CylinderPosition” tag.

After the I/Os and memory words of all controllers in the SCADA system were defined with TAGs, the electro-pneumatic press system and the SCADA computer were tested in runtime mode of CitectSCADA 2015.

5. RESULTS AND DISCUSSION

After the designed electro-pneumatic press system and SCADA system were taken into runtime as shown in Fig. 9, time-based and functional analyzes were performed and OPC-based systems were analyzed on a case-by-case basis. For functional analysis, while the system was operated by the operator using HMI device, it was interrupted at random intervals and tasks such as resetting the counters and memory words, moving the pneumatic cylinders synchronously or individually, changing the permissions in the system were performed over SCADA interface. Electro-pneumatic press system and SCADA computer connection was also tested with both LAN and Wi-Fi for the assigned tasks.

As demonstrated in practice in this study, OPC-based SCADA systems reduce the time spent during system design with the help of OPC software that bridges between protocols, eliminate the complexity of working with more than one protocol, and reduce the investment costs by providing flexibility in hardware choice to the system. In addition to the stated main advantages, connecting the industrial controllers to the SCADA system with the help of smart tags is a great convenience for applications that require rapid expansion and do not allow operational pauses. As a result of the performed time-based analyzes, we found that cross-tagging has advantages allowing flexible and fast revisions of the variables, however it makes the system relatively far from being real-time.

PLCs using different industrial communication protocols were only communicated with the help of TAGs without any software changes and analyzed on a system-specific basis. As a result of cross-tagging, it was observed that transactions took place 750 milliseconds slower on average, moving away from being real-time. When similar operations were performed by making changes to the device tags and SCADA program instead of cross-labeling, it has been observed that the average delay was 250 milliseconds. For this reason, it was concluded that instead of cross-labeling in time-critical applications, it would be more accurate to define control directly on the SCADA system or to change the software in PLCs. The main reasons for the delays in the system were the processing capacity of the PC used as a SCADA computer and the simple Ethernet HUB used in the design of the network infrastructure and sending the same data to all controllers by broadcasting.

6. CONCLUSION

In this study, an OPC-based SCADA system was designed for an electro-pneumatic press system containing PLCs and open-source controllers using different industrial communication protocols of numerous manufacturers; network infrastructure and the physical connections were prepared, commissioned, and analyzed. We commissioned the OPC and SCADA software by running them on the same computer and connecting the hardware with a star topology established with the help of Ethernet HUB. The OPC-based SCADA system was tested in runtime mode and controllers from different commercial brands using various industrial communication protocols successfully talked to each other through the system.

Fig. 9 Runtime tests of the SCADA system

The designed OPC-based SCADA system's functional and time-based analyzes were performed. In addition to the flexibility, extensibility and hardware-independent design advantages provided by the OPC-based systems, which we have examined for all the design stages, the problems that may arise from being away from real-time are also experimentally examined and shared. As a result of the runtime tests carried out, the following observations and findings were obtained.

- ❖ Device to device response time may exceed 500 milliseconds while using star topology created with Ethernet HUB that broadcast all the data to network.
- ❖ Cross tags connecting the input, output and memory words of the hardware also increase the latency times. Although observed delay times are insignificant for the pressing unit used in the study, the construction of the network infrastructure with smart switches, use of computers with more powerful processing capacity and avoiding using cross-tags for critical operations will reduce the delay times in applications.
- ❖ Even though there are increases in delay times, the use of cross tags brings the advantages of flexible and fast system expansion, and additionally, complex systems with multiple controllers can be built with low hardware costs and standard network elements.

7. SUGGESTIONS FOR FUTURE STUDY

For future work, it would be possible to improve overall system performance and reduce latency times from the findings of this work. In cases where the use of star topology is an obligation in industrial networks, using smart switches instead of Ethernet HUB would be more convenient to reduce latency. Although using cross-labeling instead of changing the software of industrial controllers speeds up the commissioning process and allows flexible changes, it will not be suitable for use in critical time operations as it causes delays in device-to-device message transmission. In future works, also an additional fault-detection web interface may be designed for sharing with 3rd party users which are not authorized to access SCADA screen for cyber-security purposes.

DECLARATION OF ETHICAL STANDARDS

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

Okan DUYMAZLAR: Implemented the system, wrote the control programs, and conducted the analysis of the results. Also, wrote the manuscript.

Dilşad ENGİN: Conducted the evaluation of the results; has a role in methodology; wrote part of the manuscript and edited whole paper.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] Givehchi O., Landsdorf K., Simoens P., and Colombo A. W., "Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems", ***IEEE Trans. Ind. Informatics***, 13(6): 3370–3378, (2017).

[2] González I., Calderón A. J., Figueiredo J., and Sousa J. M. C., "A literature survey on open platform communications (OPC) applied to advanced industrial environments", ***Electron***., 8(5): 510, (2019).

[3] Stouffer K., Falco J., and Scarfone K., "Guide to industrial control systems (ICS) security", ***NIST special publication 800-82***, 1-255, (2011).

[4] ŞAHİN C., KARAÇOR M., and ÖZBAY H., "Yenilenebilir Enerji Kaynağı Kurulum Gücü Minimize Katsayısının Belirlenmesi", ***Avrupa Bilim ve Teknol. Derg.***, 15: 404–411, (2019).

[5] González I., Calderón A. J., Barragán A. J., and Andújar J. M., "Integration of sensors, controllers and instruments using a novel OPC architecture", ***Sensors (Switzerland)***, 17(7): 1512 , (2017).

[6] https://opcfoundation.org/about/what-is-opc/ "What is OPC?" , ***OPC Foundation***, 2010, (02 July 2019).

[7] https://opcfoundation.org/about/opc-technologies/opc-ua/ "Unified Architecture", ***OPC Foundation,*** (5 November 2021).

[8] Lutz P., "OPC UA: wider controller communications with TSN, APL: OPC UA uses Ethernet Time-Sensitive Networking (TSN) and Ethernet Advanced Physical Layer (APL) for controller-to-controller networking. Goal: interoperability", ***Control Eng***., 68: 28–30, (2021).

[9] Aleksandrov S., Jovanovi Z., Nikoli S., and Stanimir Č., "MECHATRONIC SYSTEMS CONTROL BASED ON SCADA package", ***FACTA Univ. Ser. Autom. Control Robot***., 10(2): 189–198, (2011).

[10] Barsoum N. N. and Chin P. R., "Ethernet Control AC Motor via PLC Using LabVIEW", ***Intell. Control Autom***., 02(04): 330–339, (2011).

[11] Lakshmi Sangeetha A., Naveenkumar B., Balaji Ganesh A., and Bharathi N., "Experimental validation of PID based cascade control system through SCADA-PLC-OPC and internet architectures", ***Measurement***, 45(4): 643-649, (2012).

[12] Toylan H. and Kuscu H., "A research on SCADA application by the help of OPC server for the water tank filling system", ***Sci. Res. Essays***, 5(24): 3932-3938, (2010).

[13] Üstünsoy F. and Sayan H. H., "PLC Destekli SCADA ile Enerji Yönetimi İçin Örnek Laboratuvar Çalışması", ***J. Polytech***., 21(4): 1007–1014, (2018).

[14] Chamorro-Atalaya O., Arce-Santillan D., Diaz-Leyva T., and Diaz-Choque M., "Supervision and control by SCADA of an automated fire system", ***Indones. J. Electr. Eng. Comput. Sci***., 21(1): 92-100, (2021).

[15] İŞBİLEN F. ve KONAR M., "Uçak Sistemlerinin SCADA İle Modellenmesi", ***Avrupa Bilim ve Teknol. Derg.***, 18: 338-346, (2020).

[16] Nicola M., Nicola C.-I., Duță M., and Sacerdoțianu D., "SCADA Systems Architecture Based on OPC and Web Servers and Integration of Applications for Industrial Process Control", ***Int. J. Control Sci. Eng***., 8(1): 13–21, (2018).

[17] Singh R. and Verma H. K., "Development of PLC-Based Controller for Pneumatic Pressing Machine in Engine-Bearing Manufacturing Plant", ***Procedia Computer Science***, 125: 449–458, (2018).

[18] Niculescu F., Savescu A., and Mitru A., "Transmitting data over the network using an OPC server", ***22nd International Conference on Circuits, Systems, Communications and Computers (CSCC 2018)***, Majorka, Spain, in ***MATEC Web of Conferences***, 210, 03002: 373-378, (2018).

[19] Hadi H. H. and Sallom M. Y., "Pneumatic Control System of Automatic Production Line Using SCADA Implement PLC", in ***4th Scientific International Conference Najaf, SICN 2019***, Al-Najaf, Iraq, 37–42, (2019).

[20] Duymazlar O. and Engin D., "Design and Application of OPC-based SCADA System with Multiple Controllers: An Electro-pneumatic Case Study," in Proceedings - ***2019 Innovations in Intelligent Systems and Applications Conference***, ***ASYU 2019***, İzmir, Türkiye, 208-213, (2019)