

# Using of OPC SERVER Technology to Solve Communication Problems and Incompatibility between Different Components and Devices in a Control System

Hazem Zidan <sup>[1]</sup>, Their Ibrahim <sup>[2]</sup>

Department of Industrial Automation Engineering  
University of Tartous  
Syria

## ABSTRACT

In this essay we will study the using of OPC standard in adapting and removing the incompatibility between different components and devices in the control system. The OPC works as a third party that connect between devices (data sources) and programs that require this data, in case those programs may can't connect directly with that devices as a result of incompatibility between them. This incompatibility happens because these devices and programs are from various vendors or because the devices are very old and can't connect with newer programs, this results from the using of two different communication protocols in the components of control system. OPC will be a suitable solution for communications problems between these devices and programs in the control system, the OPC will connect to each of them and transfer data as good as if the connection between them is direct without any third party.

**Keywords:**— SCADA, PLC, OPC, Control, Communication Protocols.

## I. INTRODUCTION

The industrial networks used in the control systems have evolved considerably over the past decades. Many types of networks have emerged and now we have networks that reach a huge number of points spread around the world. This variety and diversity of network types is due to the many protocols used and developed by different companies in order to communicate with various devices. Over time, this has resulted in gaps between the old devices and the modern communication protocols, which appear in turn because of their incompatibility, which necessitated the existence of a medium that provides access to old devices without having to change them.

For this purpose, a standard protocol agreed upon by several large international specialized industrial companies, called "OPC" (OLE for Process Control), where "OLE" is an abbreviation for Object Linking and Embedding, a working environment in Windows that is used to link objects and embed them in windows and applications [1].

OPC is the interoperability standard for the secure and reliable exchange of data in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors [1].

We will use the OPC in solving one of the communications problems between two different control systems in the Syrian Company for Oil Transport (SCOT).

The Syrian company for oil transport (Scot) is the main operator of transportation activities of petroleum from and to Syria. Scot operates many pipeline networks that transport crude oil petroleum fields Banyas and Tartous terminals.

SCOT also stores the oil products in a field of tanks contain about 42 tanks then the products are transported from Banyas to other arias in the country. [2]

The Main pumping control system in SCOT has been developed recently by using new SCADA & PLC systems to control the pumping and storing processes. The company also has an old sub-pumping system which controlled by its own PLC.

A problem appears when SCOT wanted the sub-pumping system to be monitored by the new SCADA. The sub-system PLC uses Modbus protocol for communication, and the SCADA system doesn't have a full support for Modbus (this will be explained later) and there will be some faults in reading data from the old PLC, so there was a strong need to use a third party program to connect them together.

An OPC Server program will be a good solution for this problem, we chose a program which is supported by both of SCADA and old PLC, made the needed configurations as we will see later, then a connection was created and data was transmitted between them successfully.

## II. A QUICK OVERVIEW ABOUT MODBUS

This protocol regulates and manages the communication process for the transmission of data in the industrial network between controllers and related devices. It defines a message structure that controllers will recognize and use, regardless of the type of networks over which they communicate. It describes the process a controller uses to request access to another device, how it will respond to requests from the other devices, and how errors will be detected and reported. It establishes a common format for the layout and contents of message fields [3].

What concerns us here is the transmission modes, the serial port communication parameters (baud rate, parity mode, etc.), and how to address the variables used in the control process. The transmission modes that will be used here is Modbus RTU, and for addressing, the protocol uses a different addressing method depending on the data type, in this project we will use following types of data [3]:

- Coils Status: ON/OFF status of discrete outputs, the addressing of it is in the form of (0xxxxx).
- Input Status: ON/OFF status of discrete inputs, the addressing of it is in the form of (1xxxxx).
- Input Registers: the binary contents of input registers, the addressing of it is in the form of (3xxxxx).

## III. GATHERING INFORMATION

First we need to know what are the tags and variables in the PLC that need to be monitored, so we'll collect information about them, (Tag Name, Tag Description, and Tag Address)

We found two types of Tags to be monitored:

- Discrete Inputs with addresses (10xxxx) which includes alarms about temperatures and statuses of pumps ... etc.
- Input Registers with addresses (3xxxxx0) which include readings from transmitters.

As we said previously that there is a problem in SCADA that couldn't read some data in the right way, the problem is with Input Registers, SCADA program read these data as Integers but we need to read them as Real (float) numbers, so there will be wrong readings of these data, that is the reason why we suggest to use OPC program.

In return to variable tags collecting, we found 80 variables that we need to monitor the sub-pumps. In the next table we can find an example of tags that we will use:

TABLE I  
SOME VARIABLE TAGS OF THE SUB-PUMPS

Variable Name	Variable Description	Address
P6_C_OUT_P_LO	Pump 6 Coupler Output Pressure Lo	000257
P6_C_OIL_T_TRIP	Pump 6 Coupler oil Temperature Trip	000259
P6_C_INLET_T_HI	Pump 6 Coupler Inlet Temp Hi	000260
P6_P_NDE_SEAL_LK	Pump 6 Pump Output Seal Leakage	000266
P6_F_MOT_RUN	Pump 6 Field Motor Running	000270
P6_F_FLOW_SW	PUMP 6 FLOW SWITCH	000272
P6_M_TEMP1	Pump 6 Motor Temperature 1	320480
P6_M_TEMP2	Pump 6 Motor Temperature 2	320481
P6_C_INP_BRNG_T	Pump 6 Coupler Bearing Temperature	320492
P6_P_DE_VIBR	Pump 6 Pump Input Vibration	320500

## IV. CONFIGURATION OF OPC PROGRAM

After successful installation of an OPC server program a configuration have to be created and registered. To create the configuration means to define the list of devices connected to the OPC server, to define parameters of these connections and to define the list of tags (OPC items) that should be read from the device. Configuration have to be saved and registered, so it can be loaded immediately after OPC server starts [4].

After gathering information about the PLC system, we want to enter data into the OPC program, this includes the name and address of each tag. Also we need to define the serial port communication parameters (baud rate, Data Bits, Stop Bit, parity mode), so the OPC configuration is completed and we can now connect OPC to the PLC and ensure that our work is so far so good [5].

## V. CONFIGURATION OF SCADA SYSTEM

The SCADA system has the ability to communicate as an OPC client, so we chose this kind of communication in it, and then we connected the SCADA to the OPC program that we previously installed, after that we draw data from OPC program then we have all tags with their addresses in SCADA program.

Now we have all wanted tags, we can begin to design the interface windows that we will use to monitor the sub-pumping system.

## **VI. CONCLUSION**

After we finish the configurations of both OPC and SCADA programs we can now test them. We connected the PLC to the monitoring computer through the proper port, Run the OPC and SCADA programs and ensure that all parameters can be read correctly and the work of the sub-pumps can be monitored without any problems.

Result proved that OPC technology is an excellent solution to the problems that existed, despite the time and effort taken by the process of redesigning and writing variables and linking, but at last we have reached the desired result

## **VII. ACKNOWLEDGEMENT**

This article is part of a research submitted for a master's degree in industrial automation engineering at the University of Tartous (Resolution No. 1690/2018).

The authors would like to thank Morris Alsaad and Ali Youssef for their help and technical support for this work.

## **REFERENCES**

- [1] (2018) OPC Foundation website. [Online]. Available: <https://opcfoundation.org/about/what-is-opc>
- [2] (2018) SCOT website. [Online]. Available: <http://www.scot.gov.sy>
- [3] *Modicon Modbus Protocol Reference Guide*, Modicon, 1996.
- [4] *Reliance 4 – OPC Tutorial*, Geovap, 2018.
- [5] *Keeware to TBox Serial Communications Setup*, PCD, 2015.