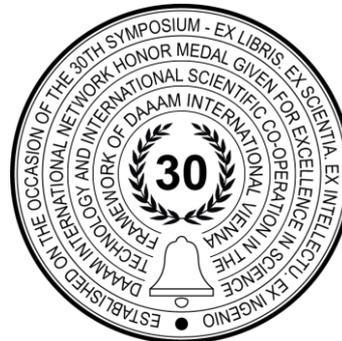


DEVELOPMENT OF MACHINE MONITORING SYSTEM

Roman Polak, Vaclava Lasova, Petr Bernardin, Petr Janda & Marcel Svagr



This Publication has to be referred as: Polak, R[oman]; Lasova, V[aclava]; Bernardin, P[etr]; Janda, P[etr] & Svagr, M[arcel] (2020). Development of Machine Monitoring System, Proceedings of the 31st DAAAM International Symposium, pp.0310-0314, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-29-7, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/31st.daaam.proceedings.043

Abstract

This paper deals with a complete design and implementation of a monitoring system for mobile devices, which was prepared for the company Škoda Machine Tool a.s.. It is a monitoring of machines that have an OPC UA Server for network communication. This communication protocol is used to transfer data from the machine to the server. The data transfer run in the set interval in seconds. The data is stored in a database for further processing. Access to the database is possible via a mobile application (Android or iOS device) or a web application. The web application also has an administration interface for setting up the whole system. These end-user applications allow monitoring of the current main machine state, secondary states and current axle loads. It is also possible to display statistics in a selected or preset time interval.

Keywords: monitoring; machine tool; mobile application; OPC UA Server; OPC .NET API

1. Introduction

The main goal of this work is to connect a mobile device to a machine tool. This work design and implement a solution for monitoring machine tools via the OPCUA interface. This OPCUA interface is a general and open solution that uses a large number of industrial machines. From this point of view, the solution can be described as general and usable in many cases. The main objective of this monitoring system is to display data from the machine on a mobile device. Due to the technologies used and data retention requirements, it is not possible to connect mobile devices directly to the machine. For this reason, it is necessary to process the data from the machine and prepare it for the end mobile devices. A specific solution is described below. The paper describes the development of the monitoring system from design to implementation. The first part describes the design of the system including individual parts and communication between them. At the same time the implementation of server and client part is described. Principles and technologies used are given in these parts. The last chapter describes the mobile application, parts, options, installation method and settings.

2. System design and implementation

The system was initially designed on the basis of a third-party system already in operation, but without a mobile application. Only part of the required data could be obtained from this solution. It was not possible to automate the data retrieval to guarantee the timeliness of the data. For this reason, the system is designed using OPC UA Server.

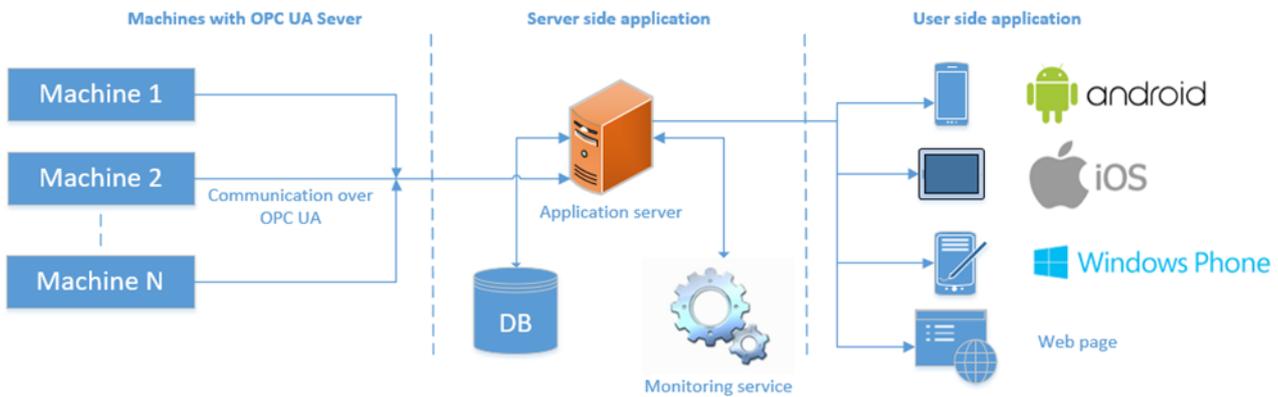


Fig. 1. Diagram of the proposed system

The system diagram (Figure 1) shows the individual parts of the system and the communication between them. Machine tools from Škoda Machine Tool a.s. have Siemens Sinumerik control system including OPC UA Server. The basic requirement is that the machine-side OPC UA Server, including configured addresses, is turned on. The whole core of the system is an application on the server that accesses these addresses and retrieves the required information from them. The loaded information is stored in a database for later processing. Communication between the server and the machines is only one-way, i.e. data is only read from the machine and stored in a database. Client application databases (Android, iOS, website) access data via the application interface. The client side is described below.

2.1 Server side

The system core or server part of the application is created using the .NET C# programming language. [1] It also uses the OPC .NET API¹ library to connect and retrieve data from the machine. MS SQL Server database is used to store information. The database can be used with the existing version or the “Express” version is sufficient, which is free to download. The resulting compiled program is executable on the server as a regular system service. It is possible to set when the server part should be started and whether this action should take place automatically or manually.

The OPC Unified Architecture (UA), released in 2008, is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework. [7] For Client-Server communication the full range of information model access is available via services and in doing so follows the design paradigm of service-oriented architecture (SOA), with which a service provider receives requests, processes them and sends the results back with the response. [3] OPC UA with client/server communication is used extensively in automation. With this proven, one-to-one communication mechanism, each OPC UA client obtains access to the data on the OPC UA server via point-to-point communication. The OPC UA client sends a request to the OPC UA server, from which it receives a response. [6][9] This form of communication enables a reliable, secure, encrypted data exchange with reduced bandwidth and no loss of data, even if the environment and network quality are sub-optimal and are experiencing disturbances (Figure 2). [11][13] Client/server communication is always based on TCP/IP². [4][12]

There is also a web part of the application on the server, which serves as a configuration interface of the whole system in case the system administrator is logged in. The same environment is available to ordinary users as the mobile app described below. This part of the application is also created using .NET C# programming language using HTML³ 5 and CSS⁴ 3 web technologies.

2.2 Client side

Client applications are used to display data from machines. In practice, however, they display data from the server (see system diagram), even if it is up to date information about the machine. Client applications are not demanding because all data is processed by the server. Mobile apps retrieve the minimum amount of information needed to display. Data collection, storage and calculation of statistical data takes place on the server. The mobile application then only displays this data. Communication between the mobile device and the server takes place using the TCP/IP protocol. [10] The mobile device must have access to the corporate server with the machine monitoring system application installed. The method of connection (Wifi or mobile data) depends on the method of implementation and location of the mobile device. [8]

¹ API – Application Programming Interface

² TCP/IP – Transmission Control Protocol / Internet Protocol

³ HTML – Hypertext Markup Language

⁴ CSS – Cascading Stylesheet

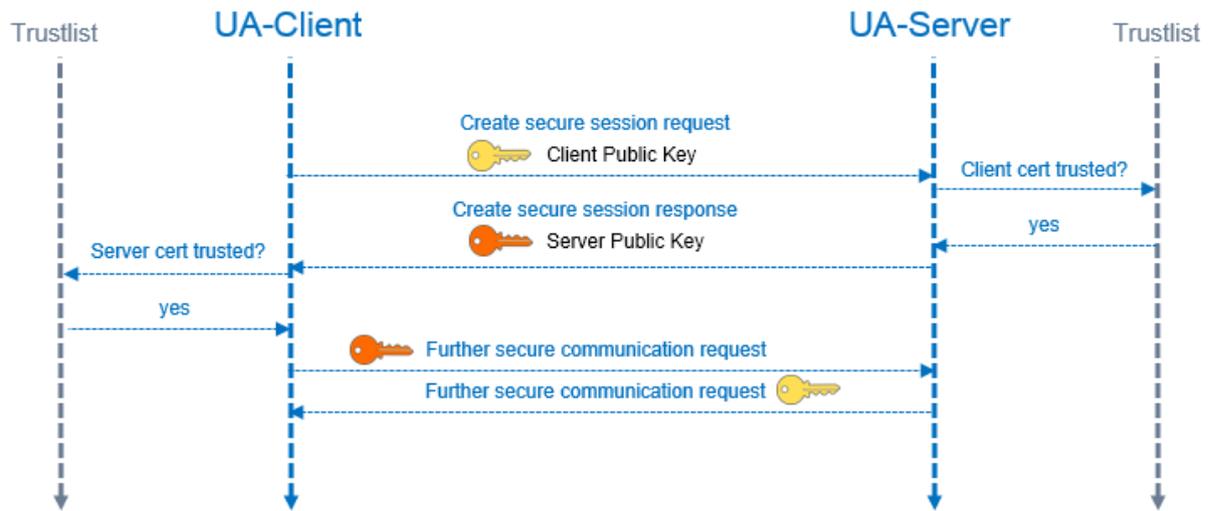


Fig. 2. OPC UA communication

The mobile client app is available for Android and iOS devices. Both are native applications written in Java (Android) and Swift (iOS). [2][5]. The mobile application currently has these features and information below. Live and Statistics screens are on Figure 3.

Live information

- Machine selection within the company
- Main status display (Cycle, Adjustment, Fault, Off)
- Display of minor states (Spindle rotation, Working feed)
- Axis load (By machine type - Spindle, X; Y; Z; W axis)

Statistics

- Display of predefined interval (Week, Month, Year, All)
- Display your own time interval (According to the specified time interval)

Settings

- Setting up the connection to the application server (IP, port)
- Other settings take place within the server application



Fig. 3. Mobile application (Live and Statistics)

3. Deployment in production environment

Deployment of the monitoring system into the production environment can be adapted to specific conditions. Depending on the requirements and possibilities of the company, the application can be deployed within the company network. All devices connected to this network can connect to the application server and receive data from machines. In case of access outside the company network, there are more possibilities. The end user device (mobile application) can connect to the VPN. Then the device will behave in the same way as if it were connected to the local network. Disclosure outside the corporate network is specific to the company and can be directed directly without using VPN. The application will then work with any Internet connection, such as a data connection from the operator. This deployment principle is displayed on Figure 4.

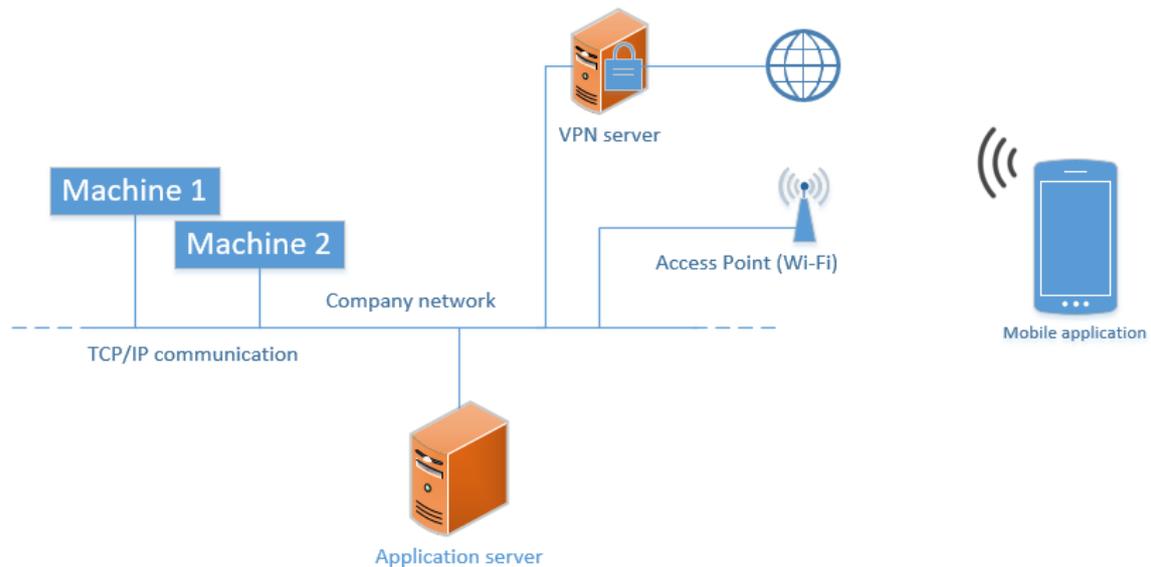


Fig. 4. Network deployment diagram

Server-side installation is performed using an installation package that contains the application itself, MS SQL Server installation, including database deployment and the web part of the application. Installation on mobile devices is only possible through official stores Android - Google Play and the App Store - iOS. The application is currently only available in the testing mode within the internal user group.

4. Conclusion

The solution presented in this article allows monitoring of current values directly from the machine. The machine must be connected to a data network and support the OPCUA protocol. Machine monitoring from mobile devices is currently in test mode for Android devices. The original plan for using existing machine data had to be revised using the OPC UA Server. Thanks to this step it is possible to monitor the current data from the machine. The whole core of the application is on the server side and does not burden the end devices. For example, displaying statistical data from a longer time interval is time consuming to calculate. The statistics are calculated on the server and only the result information is transferred to the mobile device. The future expansion plan of the application is in adding information screens including diagnostic data.

5. Acknowledgments

This paper is based upon the work sponsored under the project "Comprehensive support for designing of technical equipment" no. SGS-2019-001 and in cooperation with Škoda Machine Tool a.s. using the laboratory for research and testing purposes.

6. References

- [1] Microsoft, C# Programming Guide (2020). Available from: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>.
- [2] Google, Documentation-Android Developers (2020). Available from: <https://developer.android.com/docs>.

- [3] O. Foundation, Unified Architecture (2020). Available from: <https://opcfoundation.org/about/opc-technologies/opc-ua/>.
- [4] Beckhoff, Information system - OPC UA Certificate exchange (2020). Available from: <https://developer.apple.com/documentation/>.
- [5] A. Inc., Apple Developer Documentation.
- [6] Stopper, Markus, and Branko Katalinic. "Design aspects of the OPC unified architecture." *Annals of DAAAM & Proceedings*, 2008, p. 1321+.
- [7] Zezulka F., Marcon P., Bradac Z., Arm J., Benesl T., Vesely I. *Communication Systems for Industry 4.0 and the IoT. IFAC/PapersOnLine*, Volume 51, Issue 6, 2018, Pages 150-155.
- [8] Matýsek M., Adámek M., Karafiát T. The mobile monitoring and control of real systems. *Annals of DAAAM and Proceedings of the International DAAAM Symposium*. 2015, vol. 2015-January, p. 104-108
- [9] Kumar, R.; Bose, A.K. *Internet of Things and OPC UA. ICNS 2015*, 2015, 52.
- [10] Leitner, S. H., & Mahnke, W. (2006). *OPC UA–service-oriented architecture for industrial applications. ABB Corporate Research Center*, 48, 61-66.
- [11] Huang, R., Liu, F., & Dongbo, P. (2010). Research on OPC UA security. In *2010 5th IEEE Conference on Industrial Electronics and Applications*, pp. 1439-1444.
- [12] Drahoš, P., Kučera, E., Haffner, O., & Klimo, I. (2018). Trends in industrial communication and OPC UA. In *2018 Cybernetics & Informatics (K&I)*, pp. 1-5.
- [13] Beckhoff Automation GmbH, Beckhoff information system – Secured OPC UA Communication, Available from: <https://infosys.beckhoff.com/>