Streaming Information Transmission Based on OPC UA

Liu Wei, Zu Yunxiao*, Li Weihai
School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
zuyx@bupt.edu.cn

Abstract. OPC UA is a new specification developed on the basis of classic OPC. It overcomes the limitations of classic OPC, has the advantages of platform independence and scalability, and is widely used in industrial fields. Node-Red is a visual programming tool based on node.js that allows you to combine modules to write applications and make programming simpler. This paper designs a Node-Red to create a stream as an OPC UA client by connecting OPC UA related node modules, connect to the OPC UA server, and call services such as reading and writing, subscription, etc., to realize real-time monitoring and display of the workshop temperature, and start the fan when the temperature is too high. In addition, Node-Red has a display module that provides a more diverse selection of information.

1. Introduction
With the concept of Germany's "Industry 4.0", smart manufacturing has become a research hotspot in the global manufacturing industry. For this reason, China has formulated the "Made in China 2025" project with information technology and industrialization as the main line. OPC UA (Unified Architecture) technology is a new technology proposed by the OPC Foundation and is widely used in industrial control. It provides a secure, reliable and vendor-independent communication interface that enables the transfer of raw data and preprocessed information from the manufacturing level to the production planning or ERP level [1].

The OPC UA client connects to the server, invokes the server's services, completes the task of reading and writing content on the address space, and subscribes to variables or events through monitoring items. The real-time monitoring of temperature has an important significance in the industry to ensure smooth production and safety.

1.1. OPC UA
OPC unified architecture (OPC UA) is a new specification developed by the OPC foundation on the classic OPC technology. It not only eliminates the limitations of the classic OPC, but also enriches the functions of the classic OPC, solves the existing problems, with the advantages of platform independence, scalability, safety and reliability, OPC UA is widely used in data exchange between industrial automation systems in manufacturing and process industries. Based on an advanced platform-independent transport protocol, OPC UA enables OPC UA applications to run on smart devices and controllers, DCS and scada systems, and even on MES and ERP systems.

The most common architectural mode of OPC UA is the client/server (C/S) model, which is also the basic OPC UA communication model. The server encapsulates the source of the process information so that the information can be accessed through the interface, after the client connects to the server, it can
access and use the data the server provides. The server provides the service, the client sends a request message to the server, and the server responds to the request.

![Client/Server model](image)

Figure 1. Client/Server model.

OPC UA is based on Service Oriented Architecture (SOA) [2], making OPC UA easier to use than traditional OPC. A service is an interface between a server and a client. It is independent of the transport protocol and development environment. It is defined in an abstract way, using a transport mechanism to exchange information between the client and the server.

The basis of OPC UA is data transmission and information modeling. Traditional OPC can only provide pure data, such as temperature values measured by temperature sensors, or pressure values measured by pressure sensors. OPC UA provides a more efficient way of presenting data semantics, in addition to the data provided by traditional OPC, it also allows the display of measurement results to be provided by a certain type of sensing device and allows for the display of the type hierarchy supported by that device.

![Temperature sensor data provided in the OPC UA Server](image)

Figure 2. Temperature sensor data provided in the OPC UA Server.

1.2. Node-Red

Node-Red is a visual programming tool developed by IBM that allows programmers to write applications by combining components. These components can be hardware devices Web APIs, functions, or online services.

Node-Red provides a web-based programming environment. Programming is done by dragging and dropping the defined node to the workspace and creating a data stream with a wire connection node. The programmer saves and executes it with one click by clicking the deploy button. The program is saved in the format of JSON string, which is convenient for users to share and modify.

Node-Red is based on Node.js, and its execution model is the same as Node.js, which is also event-driven non-blocking. In theory, all modules of Node.js can be encapsulated into one or several nodes of Node-Red. As a visual programming tool, Node-Red simplifies the tedium of programming and makes the interface more concise.
Node-Red is a graphical programming method based on node.js. This article designs the OPC UA client based on the existing nodes of the Node-Red library.

2. OPC UA client design

OPC UA usually uses a client/server communication mode, and the client gets the service by sending a request to the server. The OPC UA client application uses the application program interface (API) to invoke the services provided by the server. The OPC UA communication stack converts the client's application program interface (API) calls into messages, which can send the underlying communication entity to the server [3].

![OPC UA Client Architecture](image)

Figure 3. OPC UA Client Architecture.

2.1. Nodes

There are sets of OPC UA nodes in Node-Red, and the OPC UA nodes can also be customized, which encapsulates various types of section objects and necessary functions. To build the OPC UA client with Node-Red, connect the required modules by dragging and dropping, and set the end node in the corresponding module to connect the client to the server. Just set the Id of the object node to implement the read and write function, avoiding large-scale programming and simplifying the workflow.

There are also database module interfaces in Node-Red, which support databases such as mysql and mongodb. The read data can be stored in the database, which is convenient for querying at any time, and can also directly operate the database in Node-Red, which is convenient and quick. The Dashboard module is used as a display module in Node-Red to assume display functionality.

2.2. The function to be realized

This paper designs a factory-based workshop client to achieve real-time monitoring and display of the workshop temperature, and can start the fan program when the temperature is too high. The temperature sensor contains a status variable to indicate if it is running. There are also configuration variables that the client can read, and the configuration can be changed by writing variables [4]. The client can call methods to start or stop the fan. In addition, the client can subscribe to real-time temperature data measured by the temperature sensor and display it.

(1) Connecting to the server
(2) reading data
(3) Subscription data
(4) calling method
(5) Chart display

Figure 4. OPC UA Client Functions.

3. Implementation
The server is uniquely identified by the endpoint and connected to the server [5]. The read and write service can not only read and write the value of the variable, but also read and write the attributes of the node. By directly defining the NodeId of the target node in the input node, the content in the node can be read or written.

Subscriptions are groups of information sources that are managed by a monitoring item. There are two common types of monitoring items. The first is monitoring data and the second is monitoring events. Monitoring items usually select some common settings, such as subscription interval, monitoring mode, etc [6].

By inputting the nodeId and the subscription interval of the monitoring object in the input node, real-time monitoring of the data stored in the node at the time interval can be realized.

The chart node can output the data as a chart. By using the function node, the format of the node data is adjusted so that the abscissa of the graph is time and the ordinate is temperature [7].

Figure 5. OPC UA Client in Node-Red.

The Switch node determines the temperature of the subscription. When the temperature is too high, the method is called to start the fan. When the temperature is normal, the fan is stopped.

4. Conclusion
In the context of "Smart Manufacturing 2025", this paper designs an OPC UA client based on OPC UA. In addition to basic read and write services, real-time monitoring and display of the shop floor temperature can be realized, and the temperature can be too high. When the fan is started, centralized management of the production workshop is realized, and the workload of the on-site maintenance personnel is reduced [8]. Temperature monitoring is widely used in modern industrial production and is of great significance to the safety of industrial production.
Acknowledgments
This paper is supported by Beijing Key Laboratory of Work Safety Intelligent Monitoring (Beijing University of Posts and Telecommunications).

References
[1] Thomas Hadlich 2006 Providing device integration with OPC UA Industrial Informatics IEEE International Conference
[2] Stefan-Helmut Leitner and Wolfgang Mahnke 2007 OPC UA-service-oriented architecture for industrial applications
[3] Wang Hua and Liu Feng 2008 OPC UA technology and application Industrial Control Computer 21 (12)
[4] Lu Huiming and Yan Zhifeng 2010 Research on key technology of the address space for OPC UA server Advanced Computer Control (ICACC) 2010 2nd International Conference
[5] Zhang Lizhan, Jin Qibing and Zhao Dali 2008 Integration system of management and control based on OPC UA specification [J] Industrial Control Computer 21(9):26-27
[6] IEC 62541 -1:OPC Unified Architecture Specification -Part 1:Overview and Concepts[S].IEC, 2008
[7] Nguyen Thi Thanh Tu and Huynh Quyet Thang 2013 design and development of the air conditioning system by using OPC UA specifications and modbus protocol IEEE 8th Conference on Industrial Electronics and Applications (ICIEA) p105
[8] Daoqu Geng, Yi Wen and Qi Gao 2018 research on semantic extension method of real-time data for OPC UA 3rd International Conference on Materials Science, Machinery and Energy Engineering (MSMEE 2018) p104