

Integrated Controller “MICREX-SX” System and FA Application Software

Chitose Nakamura
Yoshitaka Takeda

1. Introduction

Today, the environment surrounding manufacturing industries is undergoing a significant shift. It has been several years since the trends toward openness, multi-vendor systems, networked systems and multimedia as symbolized by PC (personal computer) systems have advanced from information processing systems to FA (factory automation) systems. In the meantime, personal computers equipped with a standard OS (operating system) such as Windows*¹ have been increasingly introduced in factory lines for the purpose of production management, production control, monitoring and operation. The realization of custom manufacturing to meet a variety of consumer needs and achieve labor savings, however, requires seamless data transfer between an information management system with PCs and an FA system, allowing quick and satisfactory real-time response to changes at each phase based on ever-changing data. Nowadays, the environment surrounding manufacturing industries is changing remarkably due to the latest trend toward IT (information technology).

This paper introduces the MICREX-SX (hereinafter referred to as SX) system's response to these changes by focusing on the data transfer between FA application software and a PLC (programmable controller) system.

2. Technical Trends in Manufacturing Industries

The trend toward IT based on open and multi-vendor systems, which has become widespread in information systems, is eventually infiltrating FA systems.

Ethernet*² is widely used as a LAN (local area network) in information management systems and also as a LAN between controllers as represented by FL-

net. Standard operating systems such as Windows NT*³ and CE*⁴ are extending their application range from monitoring systems to control systems. The same is surely expected of Java*⁵. Distributed object-oriented technology such as DCOM (distributed component object mode) and CORBA (common object request broker architecture) have great potential to alter control systems in the future. In addition, controllers with a built-in function to log onto the internet have been developed and introduced to the market.

With this being the situation, a variety of technologies born from IT have become essential for FA systems and the problem of how to combine these technologies to construct a system has become important.

3. Current Status of FA Application

When constructing FA applications in the past, there has been a tendency to customize individual functions by adopting a custom HMI (human machine interface) and SCADA (supervisory control and data acquisition). In most cases, data transfer between an FA system and PLCs was implemented by newly developed communication functionality using a PLC-specific protocol or by using a PLC access function provided by the PLC manufacturer. In addition, it was necessary for the SCADA manufacturer to develop a specific driver for each PLC or a DDE- (dynamic data exchange) interface.

4. Requirements for FA Application Development

When constructing an FA system, it is always important to realize the system in a short period and at low cost, and to facilitate its possible modification or enhancement in the future. Moreover, to survive in a fiercely competitive market it is necessary to utilize the data in a system to realize strategic management.

The technology level in manufacturing industries

*1 Windows: A registered trademark of Microsoft Corp., USA

*2 Ethernet: A registered trademark of Xerox Corp., USA

*3 Windows NT: A registered trademark of Microsoft Corp., USA

*4 Windows CE: A registered trademark of Microsoft Corp., USA

*5 Java: A trademark of Sun Microsystems, Inc., USA

is now sufficiently high to meet those requirements. Utilization of the technologies described above and components based on these technologies allows access via a network to the shop floor PLCs and PLC data seamlessly in a short time and at low cost. In other words, it must be possible to display PLC data on the screen by merely specifying definitions and creating a simple program. Now is the time manufacturers should offer solutions to the challenges described above.

5. Countermeasures in an SX System

Under the above circumstances, the following interfaces have been developed so that data in an SX system can be accessed by variable name to reduce the man-hours required and to facilitate system development.

- (1) OPC [OLE (object linking and embedding) for process control] interface
- (2) C interface

These interfaces permit the sharing and exchanging of PLC data with machines other than PLCs and the transferring of data to the application. That is to say, a seamless connection between the shop floor and office can be realized via a network.

Specifically, a conventional FA application can be realized with an office application such as Excel^{*6} equipped with a VBA (Visual Basic^{*7} for application). In Excel, a GUI (graphical user interface) screen can easily be created, allowing those without expert knowledge to easily develop a simple monitoring HMI.

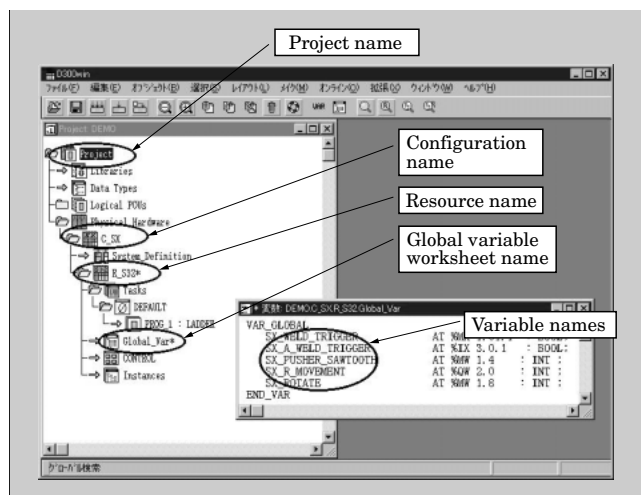
5.1 Variable names

I/O and memory in an SX system can be accessed by variable names. Variable names are defined with

*6 Excel: A registered trademark of Microsoft Corp., USA

*7 Visual Basic: A registered trademark of Microsoft Corp., USA

Fig.1 Definition of variable names with D300win



an integrated programming tool (hereafter referred to as the D300win) in compliance with the international standard language (IEC61131-3). As shown in Fig. 1, variable names are controlled at the layers of project name, configuration name, resource name and global variable worksheet name. Users can access PLC memory and I/O with variable names defined according to the D300win. Each interface shares the properties of variable names defined with the D300win, facilitating control of the variable names.

5.2 Data access through an OPC interface

OPC is a data exchange specification for FA/PA (process automation), based on OLE/COM (component object model), and provides a client-server type object interface. Several specifications for OPC have been formulated. Among these, the OPC data access (OPC DA) specification was made public in October 1998 by the OPC foundation, a nonprofit organization, and products made to this specification have been put on the market by several manufacturers. A report issued by the Japan OPC Council shows that as of October 30, 1999, there are approximately 500 such products worldwide. As shown in Fig. 2, in conventional systems, vendors had to support a different data exchange interface for each PLC manufacturer and client. As shown in Fig. 3, the introduction of an OPC DA interface, a software interface common among PLC manufacturers, however, has facilitated the seamless data exchange of PLC data. Figure 4 shows the mounting of the OPC DA interface. The OPC DA interface features access to PLC data with the variable name of "ItemID."

Products made to the OPC DA specification are expected to bring the merits shown in Fig. 5 to users, system integrators and manufacturers.

With the proliferation of products in compliance with the OPC specification, system expansion, component selection and data access have been facilitated. This has resulted in a high degree of data integration such as the direct transfer of process control application data to data base software and to spreadsheet

Fig.2 Conventional system

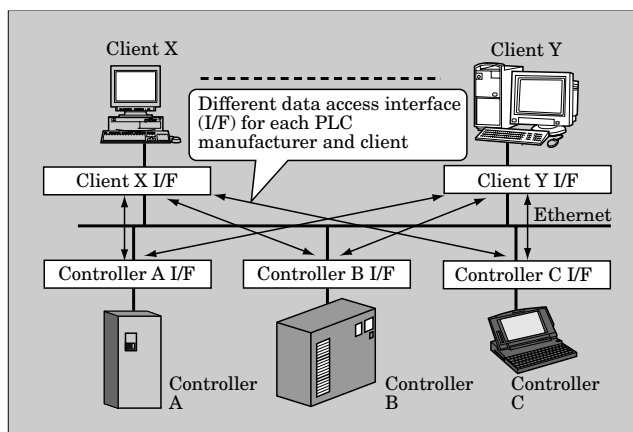


Fig.3 OPC-mounted system (conceptual diagram)

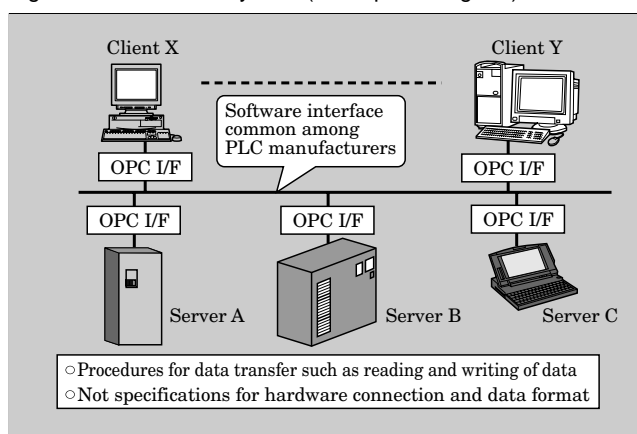


Fig.4 Mounting of the OPC interface

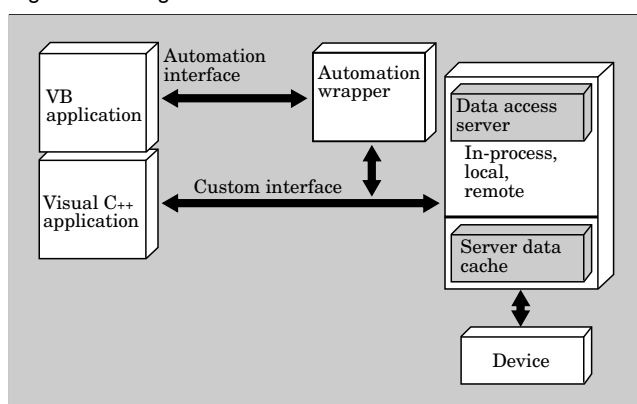
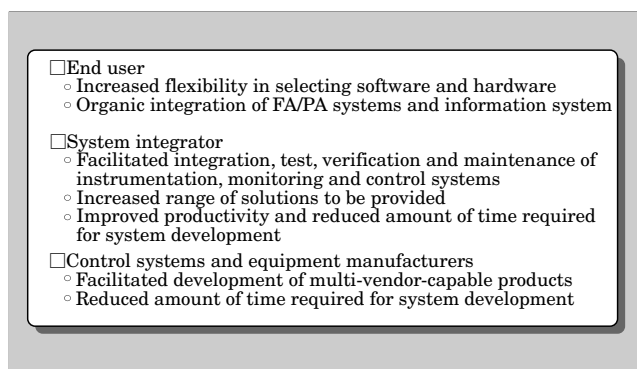


Fig.5 Merits of OPC



software, and automatic editing of the data.

Based on the trend of standardization by OPC, Fuji Electric has developed OPC DA servers intended for an SX system in compliance with specifications of OPC DA version 1.0A and 2.0.

Servers for SX support data access as shown in Table 1. The servers are connected to PLCs in the manner shown in Fig. 6.

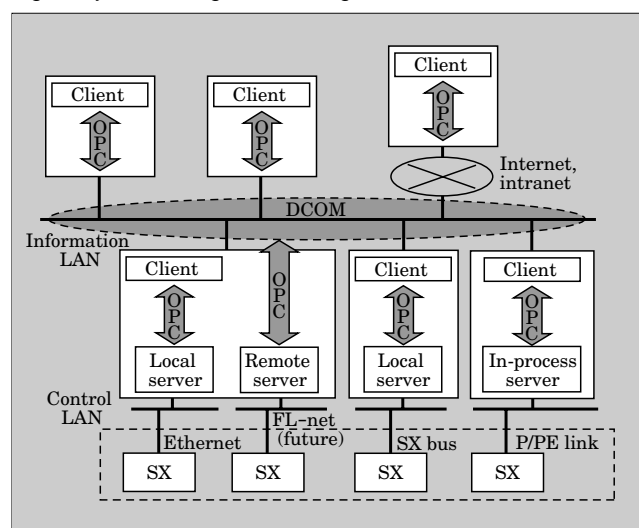
5.3 Data access through a C interface

A C interface permits data access without requiring awareness of the PLC-specific protocol and allows a developer to access an SX system by programming in

Table 1 Support functions for OPC DA in compliance with SX

Function	Method	Description
Process data read	Synchronous read	Read process data corresponding to an item. Wait until the read is completed.
	Asynchronous read	Read process data corresponding to an item. Return to original state immediately when required. When the read is completed, call a method on the data access client side.
	Refresh	Read process data from all active items.
	Notification of data change (subscription)	Check data in a certain cycle. If there is a data change beyond certain limits, notify the data access client.
Process data write	Synchronous write	Output data to an item. Wait until the write is completed.
	Asynchronous write	Output data to an item. Return to original state immediately when required. When the write is completed, call a method on the data access client side.

Fig.6 System configuration using SXs



the familiar C language. The C interface is particularly useful for adding SX to a system that uses conventional machines such as MICREX-F and FLEX-PC, and is also useful when replacing existing machines with new ones.

To be specific, the C interface supports an interface for reading and writing multi-point data into PLCs. As in the case of OPC, the C interface is connected to PLCs in various networks.

6. Example Applications

Various combinations of the above mentioned interfaces allow flexible and seamless data transfer between PLCs and an information system.

At this time, FA applications can be constructed with the following methods.



* All brand names and product names in this journal might be trademarks or registered trademarks of their respective companies.