CONNECTION OF SIEMENS PLC TO LABVIEW USING OPC

MAHFUZAH MUSTAFA

A thesis submitted
as partial in fulfilment of the requirements for the award
of the degree of Master of Electrical Engineering

Fakulti Kejuruteraan Elektrik Dan Elektronik
Kolej Universiti Teknologi Tun Hussein Onn

OCTOBER, 2004

DEDICATION

For those who pray my success and
For those waiting for my success

ACKNOWLEDGEMENT

Firstly, I would like to praise and thank the almighty Allah S.W.T for giving me the determination and the ability to complete this project.

Then, I wish to extend my sincere appreciation to Mr. Van Raaij my dedicated supervisor at Germany, for his continuous supervision, guidance, advice and constructive ideas throughout the course of this study and making this project successful. For supervisor at Malaysia, Dr. Mohamad Nor and Professor Mortaza, thank you for supervision at Malaysia.

I extend my sincere gratitude to my friend at Georg-Simon-Ohm-Fachhochshule Nürnberg Christian, Kristian and Martin for guidance, support and encouragement during the project.

Finally, I would like to express my indebtedness to my family for their moral support, encouragement and invaluable assistance in all my undertaking.

ABSTRACT

The project is to enable the Siemens PLC (Programmable Logic Control) S7-300 to communicate with the LabVIEW. The communication between S7-300 and LabVIEW is via OPC (OLE for Process Control).Development of OPC using SIMATIC NET OPC Server. OPC is industry standard provides real plug-and-play software technology for process control and factory automation. OPC integrate software and hardware across spectrum of vendors easily. Visualization of Festo Didactic process developed using LabVIEW. Festo Didactic is small system which consist actuators and sensors .The process of Festo is to make hole for three different material red plastic, black plastic and metal. PROFIBUS is a medium to transfer data from PLC to LabVIEW or LabVIEW to PLC.

ABSTRAK

Projek ini membolehkan komunikasi di antara Siemens PLC (Programmable Logic Control) S7-300 dengan LabVIEW. Komunikasi di antara PLC dengan LabVIEW adalah melalui OPC (OLE for Process Control). Pembangunan OPC menggunakan SIMATIC NET OPC Server.OPC adalah industri standard yang memberikan keadaan sebenar plug-and-play teknologi perisian untuk kawalan process dan automasi industri.. OPC juga mengintegrasikan perisian dan perkakasan bagi semua pembekal dengan mudah. Visualisasi proses Festo Didactic dibangunkan dengan menggunakan LabVIEW. Festo Didactic adalah system kecil yang mengandungi banyak pengesan dan pengerak. Proses yang dijalankan oleh Festo adalah melubangkan tiga jenis bahan iaitu plastik berwarna merah, plastic berwarna hitam dan juga logam.PROFIBUS adalah media yang digunakan untuk menghantar data dari PLC ke LabVIEW atau LabVIEW ke PLC.

TABLE OF CONTENT

CONTENTS **PAGE**

Appendix D: LabVIEW Programming and Data from PLC
READ and WRITE by OPC Scout

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DDE	-	Dynamic Data Exchange
DP	-	Decentralized Periphery
FBD	-	Function Block Diagram
FMS	-	Fieldbus Message Specification
I/O	-	Input/Output
LabVIEW	-	Laboratory Virtual Instrument Engineering Workbench.
LAD	-	Ladder Logic
LED	-	Light Emitting Diode
MPI	-	Multipoint Interface
OLE	-	Object Linking and Embedding
PC	-	Personal Computer
PCI	-	Peripheral Component Interconnect
PLC	-	Programmable Logic Control
PROFIBUS	-	Process Field Bus
STL	-	Statement List
TCP/IP	-	Transmission Control Protocol / Internet Protocol
VI	-	Virtual Instrument
WinCC	-	Window Control Centre

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

Traditionally, the industrial automation market has seen a proliferation of proprietary interface standards. Hundreds of different and incompatible proprietary interface standards developed by suppliers are required to communicate with automation systems and devices. There is a barrier because of incompatibilities and proprietary communication interfaces between different suppliers' automation hardware and software [19].Using proprietary interface method locked users into particular vendor. If users required changes in system, users were forced to return to the same vendor and subsequently forced to pay for solutions that were not always optimal. Creating a multivendor system, in automation company which can choose and apply the best products and systems for any given application often requires to invest a significant amount of money on system integration. The money invested is not just to get the desired automation and control functionality from the system. Much effort is required to ensure that the systems, can share information and interoperate with other automation and business systems in the plant or factory.

OPC *(OLE for Process Control)* removes barriers between traditionally proprietary factory floor devices, systems and other manufacturing software. OPC is open connectivity in industrial automation and the enterprise systems that support industry [19]. The standard that enables integrators to connect disparate systems together, creating robust solutions and providing true interoperability; while at the same time reducing implementation time and costs. Interoperability is assured through the creation and maintenance of open standard specifications [19]. The OPC enables a fully scalable solution for future changes and expansion. The users no longer tied or locked in to a single vendor. In addition, long-term maintenance and upgrading can be done by removing and replacing individual components in a system without any work needed to "wire up" the new pieces [19]. The advantage of OPC to supplier is reducing their time in driver development so they focus on at communication to end device, rather than worrying about different client communication schemes [19]. At the end, vendor become more competitive and offer superior product and solution to maintain their products and customers.

1.1 Problem Statement

Now day, application software should readily communicate with digital plant-floor devices as well as other applications. Most manufacturers pressing need of making hardware and software work together. The main problem of manufacturers is interfaces not standard. The proprietary system not integrates among each other.

In the absence of any standard, vendors have developed proprietary hardware and software solutions. Integrating different systems increase cost to manufacturer as well as long term maintenance and support. The custom drivers and interfaces can be written, but day by day thousands of different types of control devices and software packages increases rapidly. The other problem faces by manufactures are inconsistencies among different vendors' drivers, hardware features that are not universally supported, hardware upgrades that can wreck an existing driver, and access conflicts. When supplier comes out with new controller, software developers have to write a new driver. Each time manufacturer purchase new controller then they also has to buy new driver for the controller. This problem adding manufacturer costs.

All the problem should be solve of getting a variety of systems to work together. The solution is having a standard that provides real plug-and-play software technology for process control and factory automation for every system, every device and every driver can freely communicate, connect and integrated. The standard is OPC *(OLE for Process Control)* where OLE is Object Linking and Embedding.

Formerly student at Georg-Simon Ohm, Fachhochshule Nürnberg, Germany develop a proprietary which Siemens PLC (S7-300) with Siemens software, WinCC *(Window Control Centre)*. The OPC verified through this project which Siemens PLC (S7-300) integrated with LabVIEW.

1.2 Project Overview

Figure 1.1 shows the overview of the project. This project developed using a small factory automation called Festo Didactic. Important of the project is used OPC as the connection between Siemens PLC (S7-300) and LabVIEW. The Computer1 consist OPC which OPC Server configured by SIMATIC NET. The SIMATIC NET is software from Siemens to develop OPC Server. LabVIEW as OPC Client and also as visualization interface for Festo Didactic. Festo Didactic is small system consist sensors, actuators and stepper motors. The connection Computer1 to S7-300 is via PROFIBUS. To enable this communication, a module card CP5611 installed at Computer1. The CP5611 installed at any free PCI *(Peripheral Component Interconnect)* slot on the Computer1. The PLC connects directly to sensors, actuators and stepper motors at Festo Didactic. The Computer2 configured hardware S7-300 and connection via MPI *(Multipoint Interface)* and PROFIBUS. The Computer2 created PLC programming via STEP7. The STEP7 is software from Siemens to write a Siemens PLC programming. The description of Festo will discuss on Chapter 5 on Hardware specifications.

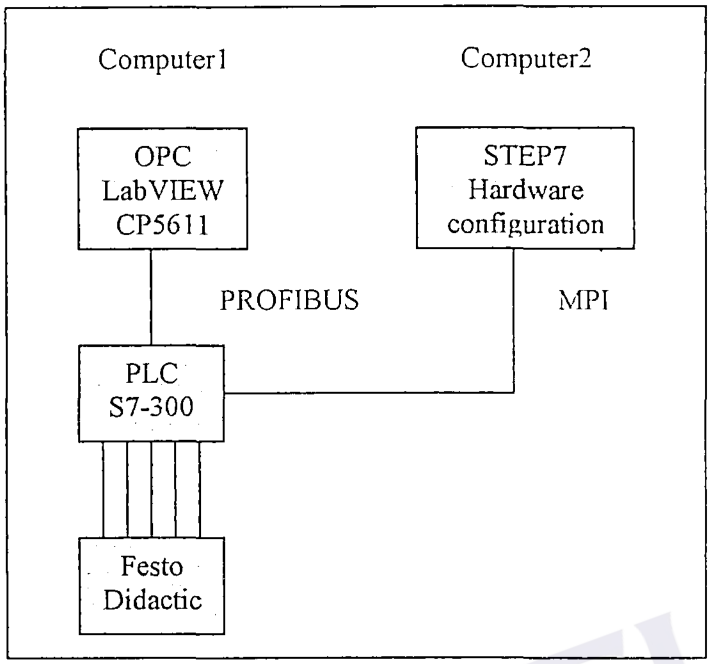

Figure 1.1: The connection between S7-300 and LabVIEW via OPC

1.3 Project Aim

The aim of this project is to develop an OPC for Festo Didactic

1.4 Objectives

The objectives of the project are:-

1. Develop OPC Server and OPC Client.
2. Enable the connection from Computer to Siemens PLC (S7-300) through PROFIBUS (RS485).
3. Develop the visualization for the Festo Didactic using LabVIEW.
4. Enhancing PROFIBUS (RS485) from 1.5 Mbps to 12 Mbps.
5. Performance Test and find out the limitations

1.5 Project Scope

The project concentrated development on OPC, LabVIEW and CP5611 module card. The project also concentrated on transmission technology via PROFIBUS (RS485). The formerly student, at Georg-Simon-Ohm, Fachhochshule Nürnberg, Germany has been done development on Siemens PLC (S7-300) to Festo Didactic, WinCC *(Window Control Centre)* and MPI connection.

CHAPTER II

THEORETICAL BACKGROUND

2.1 OPC *(OLE for Process Control)*

As Computer becomes more and more important even in industrial automation. Most Computer interface based on Microsoft Windows technologies. The first solution provided with Microsoft 3.1 was the Dynamic Data Exchange (DDE). Although an open and finally widely accepted standard its communication performance was not satisfactory. The development of the OPC (OLE for Process Control) architecture aims at solving the question of performance while providing an open interface standard between application.

2.1.1OPC Background

A standard mechanism for communicating to numerous data sources, either devices on the factory floor, or a database in a control room is the motivation for OPC. The architecture for the Process Industry shown in Figure 2.1 involves the following levels:

1. Field Management is to provide data on the health of a device, its configuration parameters, materials of construction, etc. The data must be presented to the user, and any applications using it, in a consistent manner.

2. Process Management is where installation of Distributed Control Systems (DCS) and SCADA systems to monitor and control manufacturing processes makes data available electronically which had been gathered manually.
3. Business Management is to give benefits can be gained by installing the control systems. This can be done by integrating the information collected from the process into the business systems managing the financial aspects of the manufacturing process. Providing this information in a consistent manner to client applications minimizes the effort required to provide this integration.

To do these things effectively, manufacturers need to access data from the plant floor and integrate it into their existing business systems. Manufacturers must be able to utilize off the shelf tools (SCADA Packages, Databases, spreadsheets, etc.) to assemble a system to meet their needs. The key is an open and effective communication architecture concentrating on data access, and not the types of data.

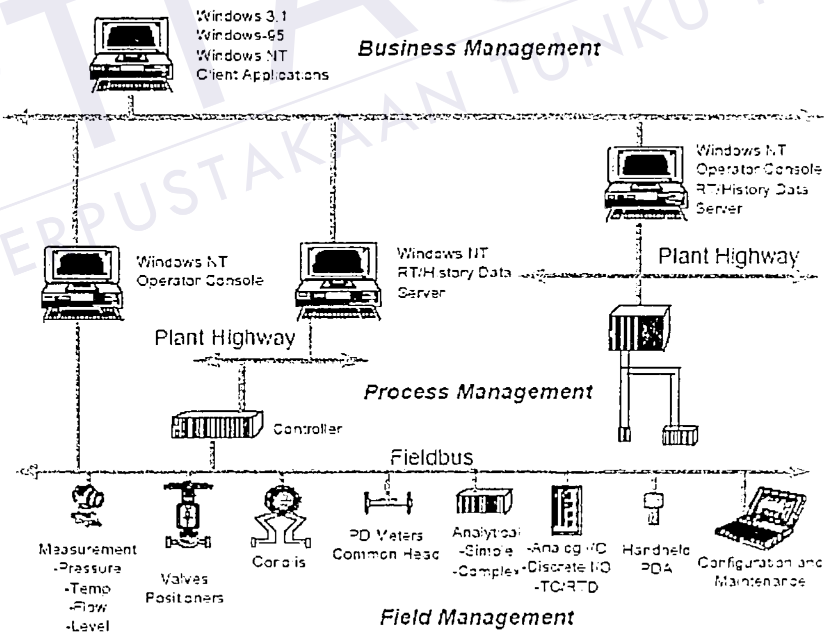

Figure 2.1: Process Control Information Architecture

2.1.2 OPC Object and Interface

The OPC Client can connect to OPC Servers provided by one or more vendor. OPC Server may be provided by different vendors. Vendor supplied code determines the devices and data to which each server has access, the data names, and the details about how the server physically access that data.

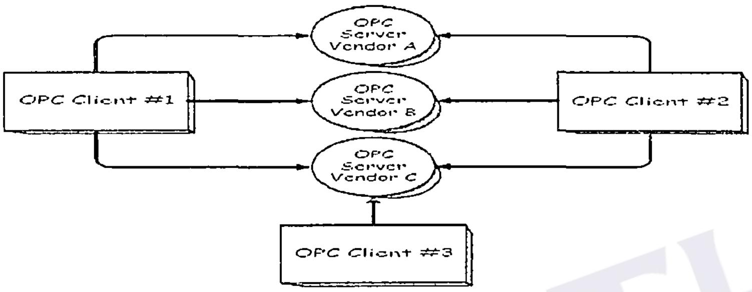

Figure 2.2: OPC Client/Server Relationship

2.1.3 General OPC Architecture and Components

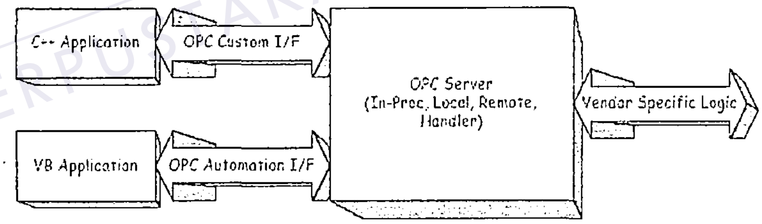

Figure 2.3: OPC Interface

OPC specifications contain two set interface; Custom Interface and automation Interface. This is shown in Figure 2.3.The OPC Specification specifies COM interfaces. It specifies the behavior that the interfaces are expected to provide to the client applications that use them. Included are descriptions of architectures and interfaces that seemed most appropriate for those architectures. Like all COM implementations, the architecture of

OPC is a client-server model where the OPC Server component provides an interface to the OPC objects and manages them. There are several unique considerations in implementing an OPC Server. The main issue is the frequency of data transfer over non-sharable communications paths to physical devices or other data bases. Thus, we expect that OPC Servers will either be a local or remote EXE which includes code that is responsible for efficient data collection from a physical device or a data base. An OPC client application communicates to an OPC server through the specified custom and automation interfaces. OPC servers must implement the custom interface, and optionally may implement the automation interface. In some cases the OPC Foundation provides a standard automation interface wrapper. This "wrapperDLL" can be used for any vendor-specific custom-server.

2.1.4 Benefit of OPC

OPC will mean the dawning of a new day for users of industrial software and hardware. En-user will receive benefit not only from the improvement in salability and integration, but also from continual improvement in technology. Vendors of industrial application, who until now spent time and money on developing software incompatible with other vendors' products, will now make an effort to develop high-quality products totally compatible with all applications. Improvement in function, quality and service will be possible throughout the industry.

For end users of industrial application, software will be one more plug-and-play technology. Just as there is a hardware backplane in PCs to which components such as memory and hard disks can be added, Microsoft's OLE/DCOM technology will make possible the creation of a software backplane allowing easy installation of many different software packages. Even though a particular manufacturer may wish to produce a new product by new process, their principal vendor may not be able to supply all necessary software and hardware components. If software and hardware developed in accordance with OPC is used, the manufacturer only has to find and install necessary software and hardware. Manufacturer will no longer be bound to one solution from a single vendor.

2.2 PROFIBUS

PROFIBUS *(Process Field Bus)* was developed in the beginning of 1991. PROFIBUS communication is anchored in the international standards IEC 61158 and IEC 61784 [17]

2.2.1 PROFIBUS Background

Before PROFIBUS plant automation was based on a centralized system with individual connections to each field device (e.g. sensors and actuators). Assembly mistakes were easily made but not so easily solved. The aim of beginning PROFIBUS was to create a single-cable solution that could provide substantial cost reduction. Primarily, this was aimed at design, commissioning, engineering and system start-up. Goals included fewer drawings, easier cable assembly and faster start-up. PROFIBUS is defined as a collapsed architecture supporting layers 1,2 and 7 of the OSI reference model as shown in Figure 2.4 [17]. For the optimized support of dedicated application areas, three version FMS (Fieldbus Message Specification), DP (Decentral Periphery) and PA (Process Automation). Since this project used PROFIBUS DP, the other types of PROFIBUS is not in the scope of this thesis. DP is the simple, fast, cyclic and deterministic process data exchange between a bus master and the assigned slave devices [17].

Sender	Receiver	Designation and function of the layers	
7	7	Application layer	Interface to application program with application-oriented commands (read, write)
6	6	Presentation layer	Representation (coding) of data for analysis and interpretation in the next layer
5	5	Session layer	Establishing and clearing temporary station connections; synchronization of communicating processes
4	4	Transport layer	Controlling data transmission for layer 5 (transport errors, break down into packets)
3	3	Network layer	Establishing and clearing connections, avoiding network congestion
2	2	Data-link layer	Description of bus access protocol (Medium Access Control, MAC) including data security
1	1	Physical layer	Definition of the medium (hardware), coding and speed of the data transmission
Transmission medium			

Figure 2.4: The OSI reference model

2.2.2Technical Data of PROFIBUS DP

The following parameters are specified of PROFIBUS DP:-

1. The bus allocation occurs by the PROFIBUS-DP after the processing of ''Token passing with supported master-slave'.
2. Typical cycle time is given with 5 -10 ms.
3. A maximum of 127 nodes with a frame length of 0-246 bytes user data can be connected.
4. Standard-transmission rates are defined as 9.6 Kbps / 19.2 Kbps / 93.75 Kbps / 187.5 Kbps / 500 Kbps / 1.5 Mbps / 3 Mbps / 6 Mbps / 12 Mbps.
5. The bus configuration is modular expandable where as the peripherals and field devices are connected and unconnected during the operation.

BIBLIOGRAPHY

1. Bishop, R.H.(2001). "Student Edition: LabVIEW^{TM} 6i" United States of America:Prentice Hall.5-6.

2. Hanim Salleh, Talal F. Yusaf, M. Khafif Zol Azlan. (2000). "Level Control Experiment via Internet" *IEEE.* II-546 – II-549.

3. Heldauer, C. (2004). "Umbau einer Modelfabrik und Anpassung der Software zur Steuerung und zur Visualisierung". Georg-Simon-Ohm-Fachhochshule, Nürnberg: Diplomarbeit

4. Janke, M. (2000). "OPC-Plug and Play Integration to Legacy Systems"*IEEE.* 68-72.

5. Johnson, G.W., Jennings, R.(2001)."LabVIEW Graphical Programming: Practical Applications in Instrumentation and Control" third edition. United States of America:McGraw-Hill. 485-486.

6. Mylvaganam, Saba. (2003). "From Sensor to Web using PLC with Embedded Web Server for Remote Monitoring of Processes" *IEEE.* 966-969.

7. Naghedolfeizi, M., Arora, S. (2002). "Operating, Monitoring, and Controlling Plant Components Over Cyberspace" *IEEE.* 887-894.

8. Pattle, R., Rämisch, J. (1997). "OPC the Defacto Standard for Real Time Communication" *IEEE.* 289-294.

9. Sandusky, R.D. (1989). "PLC and PC System Documentation Concepts" *IEEE.* 39-39

10. Shimanuki, Yoh. (1999)."OLE for Process Control (OPC) for New Industrial Automation Systems"*IEEE*. VI-1048 – VI-1050.

11. Webb, J.W., Reis, R.A.(1999)."Programmable Logic Cntrollers: Principles and Applications" forth edition. United States of America:Prentice-Hall. 4-5.

12. Zheng, Li., Nakagawa, Hiroyuki. (2002). "OPC (OLE for Process Control) Specification and its Developments" *IEEE*. 917-920.

13. Machado, C., Fonseca, J., Mendes, J. (2003)."Automatic Velocity Control in Cutting-off Machine" *IEEE*.1046-1051.

14. Kleines, H., Wüstner, P., Settke, K., Zwoll, K. (2002)."Access to Industrial Process Periphery via Java for Process Control" *IEEE*. 465-468.

15. Dahlhoff, H. (2002)."Enhancing Productivity with Automation of Material and Information Flow by Chaining Distributed Controller Units" *IEEE*. 297-302.

16. National Instrument
http://www.ni.com

17. PROFIBUS
http://www.profibus.com

18. Siemens
http://www4.ad.siemens.de

19. The OPC Foundation
http://www.opcfoundation.org