PROCEEDINGS

53. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

Faculty of
Mechanical Engineering

PROSPECTS IN MECHANICAL ENGINEERING

8 - 12 September 2008

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=17534

Published by
Impressum

Publisher Herausgeber	Der Rektor der Technischen Universität Ilmenau Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor Redaktion	Referat Marketing und Studentische Angelegenheiten Andrea Schneider Fakultät für Maschinenbau Univ.-Prof. Dr.-Ing. habil. Peter Kurz, Univ.-Prof. Dr.-Ing. habil. Rainer Grünwald, Univ.-Prof. Dr.-Ing. habil. Prof. h. c. Dr. h. c. mult. Gerd Jäger, Dr.-Ing Beate Schlütter, Dipl.-Ing. Silke Stauche
Editorial Deadline Redaktionsschluss	17. August 2008
Publishing House Verlag	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16, 98693 Ilmenau

CD-ROM-Version:

Implementation Realisierung	Technische Universität Ilmenau Christian Weigel, Helge Drumm
Production Herstellung	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts

ISBN: 978-3-938843-40-6 (CD-ROM-Version)

Online-Version:

Implementation Realisierung	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=17534

53rd Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
08 – 12 September 2008

M. Friedrich / S. Kain / M. Merz / T. Fiala

Open-Source communication solution for exchanging data between PC and ethernet enabled S7-PLC

Abstract

Factory automation systems can no more be seen as self-contained units. In fact they are part of the company's computer network [1]. Also a fixed way of communication between Programmable Logic Controllers (PLC) and Engineering Computers is not given any more. Several scenarios are known, where a PLC is remotely monitored or controlled by another PC in variable ways. In this case the PC is not integrated into the automation system and may not have installed the software required for engineering the PLC.

The goal of the presented concept is to show a simple way to communicate with Programmable Logic Controllers. With regard to communication hardware it is reasonable to rely on standard Ethernet technology as desktop PCs offer a suitable interface. State of the art PLCs with an appropriate communications processor can also be integrated into standard computer networks for non real-time applications.

Motivation

Today's automation technology faces manifold challenges - above all the variant variety of deployed devices and systems as well as particularly the complexity of proprietary solutions of information technology in automation technology. The standardization of information technology in automation still starts out and is well-known as a condition for the continuous integration of automation technology into enterprise processes [3]. Thus standardization and resorting to available standards for technology and information represent a basic condition for a comprehensive IT-integration and co-operation in and to enterprise processes. Standardization enables a clear structuring for automation technology, in order to deal with rising complexity [1].

Open communication standards moreover enable a continuous communication from the

shop floor to the management. Despite this vertical integration of automation components and systems, the presented change, away from isolated solutions, offers likewise ways for the horizontal integration within a hierarchy level. In vertical direction special attention is paid on observation of existing plants and systems, data acquisition and generally consolidated information passing.

Usecases

In Automation Technology, the complexity of systems keeps on rising fulfilling more and more requirements concerning functionality and quality. For dealing with this complexity the development is supported by a structured and tool-supported development process. At the end of the development process, the transfer of complex automation systems from development to operation phase, the commissioning, can cause high efforts.
As an example, the correction of errors, implemented during the engineering phase, can cause incalculable delays during the commissioning phase. Failures can often be deduced to mistakes in the engineering of complex automation components [4]. Software-interfaces for accessing the automation components during the commissioning phase enable the identification of occurred failures, and thus the identification and removal of their causes.
Therefore one approach for commissioning complex automation plants is to activate the plant component by component, e.g. according the Hybrid Commissioning approach [5]. If the automation function depends on different distributed components in the automation system, the signals of components which are not yet operational might be necessary for enabling the functionality. Emulating these signals concerning their (software) communication interfaces could be realized by simulation-based methods accessing on the components communication interface.
The efforts for identifying and correcting failures detected during the commissioning can be reduced extremely by the use of software tools. If software tools adapted to the type of the production plant are available, failure identification and debugging can be reduced extremely during the commissioning. The connection of simulation systems to the real production plant for testing, ranging from single components up to compounds of components, as well as monitoring via software interfaces available in the automation systems, enables an efficient support during the commissioning. Therefore the control system of the plant, e.g. the programmable logic controller, doesn’t have to be adapted.

In the operation phase of plants, the communication interface of the components enables monitoring of the automation system. The currently measured states of the system can be identified and proposed to upper-level mechanisms, e.g. for realizing diagnosis programs. Components of the automation system can even be forced via the communication interfaces, as necessary for. connecting manufacturing execution systems. Therefore the efforts for realizing the tool support mainly depend on the proposed software interfaces.

Benchmark of communication concepts

The presented use cases pose different functional requirements, but base on a shared communication interface. For realizing this interface standard solutions are available. OPC[1] bases on the COM/DCOM[2] technology and enables the communication with components manufacturer spanning. But it has to be engineered costly and requires additional software components in the control system. Proprietary communication protocols, e.g. like Step 7 DDX[3], propose a big functionality to the user, but this is often restricted to components and products of the respective brand.

If no additional software should be integrated in the automation system, an open technology for proposing a communication interface is necessary. This interface can be the basis for application specific PC-based software tools, which are connected to the plant via interfaces available in the plant.

The reusability and portability of the software tools mainly depend on the reuse of standard software elements. Thus, a communication interface has to fulfil the following requirements:

- platform independence
- renunciation on special hardware solutions
- renunciation of proprietary programming libraries
- use of a flexible programming language
- enabling communication in heterogeneous environments
- online monitoring and forcing
- online data exchange
- open interface for integration of several PC-applications

[1] OPC: OLE for Process Control
[2] COM/DCOM: Component Object Model/Distributed Component Object Model

Solution

The presented solution enables monitoring, reading and changing of system states of an Ethernet-enabled Siemens PLC for use in a PC-based application.

PLCs of the S7 series support a set of communication technologies, amongst others MPI[4], PROFIBUS, PROFINET as well as generic industrial Ethernet (including W-LAN) [6,7]. Ethernet-enabled CPUs[5] resp. communication processors therefore can be integrated in Ethernet-(office) networks. To meet the requirements, standard Ethernet provides an established and readily available communication technology. Suitable hardware interfaces are available on common PCs, supported by appropriate libraries and APIs[6] on usual mainstream platforms - thus, additional special hardware, e.g. for connecting field-buses, is not necessary. Assuming appropriate communication protocols, standard Ethernet allows crossing the borders of individual subnets, firewalls or gateways.

Besides choosing an adequate transmitting medium, selection of a communication protocol is decisive. On the basis of standard Ethernet, basically five alternatives arise [6,7,8,9,11]:

- connection over HTTP[7]

  By using a specific communication processor, variable values can be accessed via a website (Java-applets) for both input and output. Though suitable for simple monitoring/controlling needs, an online data exchange tends to be rather cumbersome.

- OPC-server/client

  Standard OPC is established especially in heterogeneous controller environments. The mapped data model of a logic controller gets accessible via standard libraries. Client side applications are normally bound to the Microsoft Windows platform, as dependencies towards the communication standard (D)COM exist. Platform boundaries as well as the additional projection costs have to be taken into consideration.

---

[3] DDX: Dynamic Data Exchange
[4] MPI: Multi Point Interface
[5] CPU: Central Processing Unit
[6] API: Application Programming Interface

- communication blocks and socket connections

  A possibility to achieve higher data rates for an online communication arises in shifting communication to lower layers of the OSI-7-Layer-Model. Siemens S7 PLCs propose so called active low-level socket connections via dedicated communication components [7,11]. Given the fact that communication thereafter normally runs asynchronous and hence across several PLC cycles, data arrays storing data for sending resp. receiving would have to be held constant [11] during cycles. Besides higher costs for engineering, implementation of the client side PC application proves to be time-consuming.
- passive low-level connection

  Besides an active low-level connection on a lower layer, connections can also be passive: Siemens equipped S7-PLCs with a socket server which enables access to arbitrary data sections. Implementation of a PC communication interface proves to be extremely time-consuming, too [12].
- connection over S7-protocol

  Complexity of implementations can be reduced by encapsulating data exchange as well as management routines. The S7 protocol supplies an ergonomic and comfortable communication interface. Unfortunately, the S7 protocol is proprietary and the specification is not published. Siemens provides the commercial programming library Prodave [10], which enables communication to a S7 PLC via the S7 protocol. The company Deltalogic [8] offers a programming library, called ACCON-AGLink, too. Both solutions do not meet the requirements of openness. An open re-implementation of the protocol by the free developer Thomas Hergenhand is available as an Open-Source project named “NoDave” [9].

The presented constraints lead to the selection of the S7 protocol provided by the library “NoDave”. By using this protocol, data exchange between PC and PLC without changing an existing PLC program or insertion of additional components becomes possible. In addition this library supports more physical interfaces than solely Ethernet ports. The necessary data rate for online communication is provided as well.

Due to a modularization approach according to the object oriented paradigm, the presented adaptable solution gets usable on many common platforms, via adequate and flexible interfaces and resorting to the established language standard ANSI-C++.

---

[7] HTTP: Hypertext Transfer Protocol

The low-level routines of the Open-Source library “NoDave” were consistently encapsulated, therefore enabling and providing ergonomic and simple access methods. Only the IP[8]-address of the communication processor is needed for parameterization; changes on existing programs are not necessary. The system state of the PLC is represented by global variables which are defined in a so called symbol table for Siemens S7. This table, exported to a file, forms the basis for all data exchange. The implemented additional function libraries of this solution enable an optional generation, reading and writing of such a file detached from an IDE[9] resp. the use of an existing symbol list as a reference for the depicted communication solution. Thereby, for reading or writing variables the knowledge of a variables name is totally sufficient. The knowledge of memory address, number of bytes and order, source resp. destination data types or the communication direction usually necessary gets completely encapsulated. The result and/or argument of the inquiry is implemented as character string for universal further application. The presented solution furthermore offers the possibility to generate PLC programs in structured text (ST, Siemens: SCL[10]) [13]. Due to its detailed interfaces, the modularity of the developed solution enables exchanging the lower communication layers so that the interface can be statically specified for the PC application and the access to PLCs of different manufacturers via a common interface becomes possible. In the course of this development a further low-level library for communication with the devices of a different automation technology manufacturer was already developed. The following figure (1) clarifies this architecture.

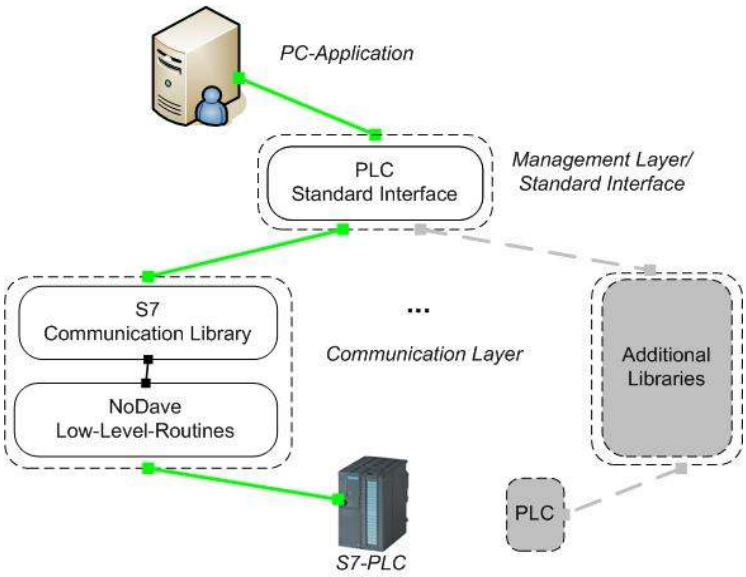

Figure 1 - Reference architecture

[8] IP: Internet Protocol
[9] IDE: Integrated Development Environment
[10] SCL: Structured Command Language

Prototypical implementation

The integration into an Ethernet network allows a modular and simple expansion or adaptation of the system as well as by using standard protocols; implementations in class libraries are already available. Here the user can acquire the data from a brand independent web service, which in turn uses a proprietary protocol to access the PLC data, in this case the S7 protocol. That way it is possible to offer consistent access to the data of PLCs of different manufacturers in a heterogeneous system via a uniform interface (figure 2 shows the setup of the prototype).

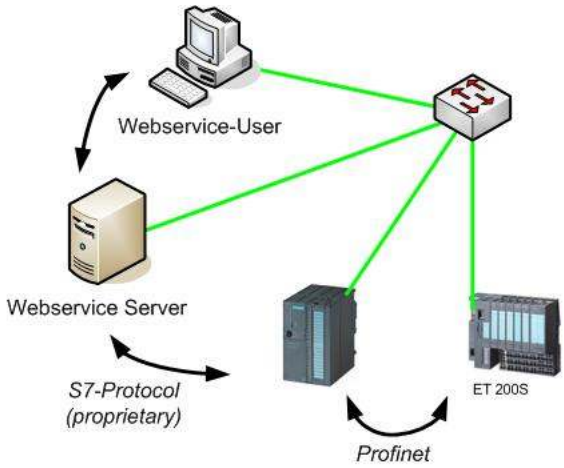

Figure 2 - Prototype setup

The symbol table from the PLC Engineering system has to be imported into the web service server and works as a database for resolving the symbolic names to the memory address within the PLC as well as for the conversion between the PLC-internal data type and a string for the web service.

The web service in this prototype offers through the SOAP[11] protocol a get and a set function for reading from and writing to the variables in the memory of the PLC. After initiating a connection to the PLC, further on only the symbolic name of the desired variable is required. All other parameters which are needed for the low-level access by the “NoDave” library are determined by the name resolution functionality based on the symbol table.

References:

[1] ZVEI – Zentralverband Elektrotechnik- und Elektroindustrie e.V. (Hrsg.), Fachverband Automation, Integrierte Technologie-Roadmap Automation 2015+, Frankfurt am Main, 2006

[11] SOAP: Simple Object Access Protocol

[2] Prof. Dr.-Ing. M. Wollschlaeger, Informationsmodelle in der Automation, Vorlesungsunterlagen, TU Dresden, 2005

[3] Prof. Dr.-Ing. K. Bender, D. Großmann, B. Danzer, FDT+EDD+OPC UA=FDD UA, Die Gleichung für eine einheitliche Gerätebeschreibung, Automatisierungstechnische Praxis (atp), 2/2007 (49), Oldenbourg, 2007

[4] Verein Deutscher Werkzeugmaschinenhersteller (Hrsg.), VDW-Bericht, Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen, Aachen, 1997

[5] Dr. S. Dominka, „Hybride Inbetriebnahme von Produktionsanlagen – von der virtuellen zur realen Inbetriebnahme“, Sierke, München, 2007

[6] Siemens AG (Hrsg.), Kommunikation mit SIMATIC S7, Online verfügbar unter: http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=de&objid=20982954&caller=view

[7] Siemens A&D (Hrsg.), Configurations zur Kommunikation, Kommunikation mit Automatisierungssystemen, , 2004, Online verfügbar: http://support.automation.siemens.com/WW/llisapi.dll/csfetch/20982954/20982954_SIMATIC_Comm_DOKU_v10_d.pdf?func=cslib.csFetch&nodeid=21169593

[8] DELTALOGIC Automatisierungstechnik GmbH (Hrsg.), ACCON-AGLink – Die vielseitige Kommunikationsbibliothek, 2008, Online verfügbar unter: http://www.deltalogic.de/content/view/21/38/lang,de/

[9] T. Hergenhahn, LIBNODAVE – Exchange Data with Siemens PLCs, 2007, Online verfügbar unter: http://libnodave.sourceforge.net/index.php

[10] Siemens AG (Hrsg.): ProdaveMPI/IE V6.0, 2005, Online verfügbar unter: http://support.automation.siemens.com

[11] Siemens AG (Hrsg.), System- und Standardfunktionen für S7-300/400 – Refenzhandbuch, 2006, Online verfügbar unter: http://support.automation.siemens.com

[12] Siemens AG (Hrsg.), Applikationen zur Kommunikation – Implementierung der Dienste FETCH/WRITE in einer S7-CPU für die offene Kommunikation über Industrial Ethernet, 2007, Online verfügbar unter: http://support.automation.siemens.com

[13] H. Berger, Siemens AG (Hrsg.), Automating with STEP7 in STL and SCL, Publics Corporate Publishing, Erlangenm Erlangen, 2007

Authors:

Dipl.-Ing. Markus Friedrich
Dipl.-Ing. Sebastian Kain
Dipl.-Ing. Martin Merz
cand. ing. Thomas Fiala

Lehrstuhl für Informationstechnik im Maschinenwesen (itm),
Prof. Dr.-Ing. K. Bender, Prof. Dr.-Ing. F. Schiller
Boltzmannstr. 15, 85747 Garching
Telefon: +49-(0)89-289-16436
Fax: +49-(0)89-289-16410
E-mail: friedrich@itm.tum.de