Connecting Programmable Logic Controllers (PLC) to control and data acquisition
A comparison of the JET and Wendelstein 7-X approach

Christine Hennig[1] and the Wendelstein 7-X CoDaC Team[1],
Klaus Kneupner [2], David Kinna[3] and JET EFDA Contributors[3,*]

[1]*Max-Planck-Institut für Plasmaphysik, Wendelsteinstrasse 1, 17491 Greifswald, Germany*
Christine.Hennig@ipp.mpg.de
[2]*Klaus Kneupner, klaus@kneupner.de*
[3] *JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK*
David.Kinna@ccfe.ac.uk
**See the Appendix of F. Romanelli et al., Proceedings of the 23rd IAEA Fusion Energy Conference 2010, Daejeon, Korea*

The use of Programmable Logic Controllers (PLC) for automation of electromechanical processes is an industrial control system technology. It is more and more in use within the fusion community. Traditionally PLC based systems are operated and maintained using proprietary SCADA systems (Supervisory Control and Data Acquisition). They are hardly ever integrated with the in fusion control and data acquisition systems.

An overview of the state of the art in fusion is given in the article.

At JET an inhouse "black box protocol" approach has been developed to communicate with any external system via a dedicated http based protocol. However, a PLC usually can't be modified to implement this special protocol. Hence, a software layer has been developed that interfaces a PLC by implementing the PLC specific communication part on one side and the black box protocol part on the other side. The software is completely data driven i.e. editing the data structure changes the logic accordingly. It can be tested using the web capability of the black box protocol. Multiple PLC types from different vendors are supported, thus multiple protocols to interface the PLC are in use. Depending on the PLC type and available tools it can be necessary to program the PLC accordingly.

Wendelstein 7-X uses another approach. For every single PLC a dedicated communication from and to CoDaC is implemented. This communication is projected (programmed) in the PLC and configurable (data driven) on the CoDaC side. The protocol is UDP based and observed via timeout mechanisms. The use of PLCs for Wendelstein 7-X is standardized. Therefore a single implementation on the CoDaC side allows communication with any PLC. Measured data from the PLC is archived in the mass data store. Set points from CoDaC can be visualized from within the PLC visualization environment.

A detailed description, discussion and comparison of the JET and W7-X approaches is given.

Keywords: Programmable Logic Controller (PLC), Control and Data Acquisition, communication, http, UDP

1. Introduction

1.1 The Fusion Experiments JET and Wendelstein 7-X

Fusion research aims to generate power by a controlled nuclear fusion process. Fusion devices in the world are unique experiments each having individual construction design and operation regime. JET – the Joint European Torus – is Europe's largest fusion device and is collectively used by more than 40 European laboratories [1]. It is operating since 1984 and has been improved ever since. JET is a tokamak type machine operating pulsed. Wendelstein 7-X (W7-X) is a fusion device currently under construction with optimized magnetic field. It is intended to demonstrate that fusion devices of the stellarator type are suitable for power plants [2]. W7-X is planned to start operation in 2014. It will be operated steady state.

1.2 The control and data acquisition systems CODAS and CoDaC

To allow scientific exploration of a unique fusion device highly specialized tools are required. These tools are often developed and maintained inhouse as is done with both JET and W7-X. Main common functionality is control and data acquisition (also known as CODAC).

CODAS & IT (Control and Data Acquisition System and Information Technology) is responsible for the design, implementation, testing, operation and maintenance of the software and hardware required for the Operation of JET. CODAS is designed for pulse based operation and supports 24/7 observation for the connected subsystems.

CoDaC (Control, Data Acquisition and Communication) provides the tools to operate W7-X in a controlled and safe way, to allow observation and data archiving and to plan and evaluate the scientific and technological results. CoDaC supports continuous operation using streaming technologies and 24/7 control and observation where applicable.

1.3 Use of PLCs

Operation of large scientific devices like JET and W7-X needs industrial standards for the automation of the electromechanical processes that “drive” the device. The use of industrial standards offers the required reliability, availability and maintainability.

PLCs (programmable logic controllers) are widespread in industries and machines. PLCs are well-adapted to a range of automation tasks. These are typically industrial processes in manufacturing where the cost of developing and maintaining the automation system is high relative to the total cost of the automation, and where changes to the system would be expected during its operational life. [3]

This makes the use of PLCs for automation of the fusion device equipment very attractive. When refurbishing existing components using a PLC is often the appropriate option. The PLC software provided by the vendor is usually specialized for factory applications and offers no easy integration to the fusion inhouse CODAC-software.

1.4 History and state of the art of PLC integration in the fusion community control and data acquisition

“The experimental activities may be partitioned into three units: electrical/electronic fundamentals, programmable logic controller (PLC) experiments, and PC control applications.” [4] The paper describes barriers to manufacturing instrumentation and control. The existence of barriers gets evidence when searching for related information. Use of PLCs within fusion is rarely described within existing papers.

[5] mentions the use of PLC but does not explain any details. [6] describes use of PLC as substitute for relay circuits and its control using RS-232C for the diagnostic interlock system. [7] describes another PLC based interlock system accessed via EPICS [8] using EtherIP. [9] is a typical paper of a control system without PLC integration. It is not present in the paper if there is a PLC in use. PLCs are not mentioned as part of control system.[10] mentions the use of PLCs within plant systems recently integrated under central CODAC. [11] describes sharing of status and control information between PLCs and a supervising master PLC using LAN.

[12] discusses possible use of EPICS in Large Fusion Experiments for slow control with respect to JET and ITER. It compares EPICS and JET CODAS and shows a possible way to combine both via a http black box protocol enhancement. [13] ITER plans to use EPICS in CODAC for slow and fast control. It standardizes to Simatic S7 PLCs and integration using EPICS. All will be tested against a MiniCODAC system providing key CODAC functionality thus allowing vendors and partners to test integration early. The necessary training and documentation will be provided by ITER partners. [14] evaluates the ITER approach on the SPA-2 project (2 switching power amplifier) for slow control only. It identified one major hurdle: using Step7 with EPICS is time consuming. There is no clear documentation available. It took much time to communicate with the PLC experts (despite trainings provided and more than over 10 years of expertise in house).

To summary the history and previous systems regarding PLC integration:

In larger experiments there is hardly any connection between slow control/SCADA and fast control and data acquisition/CODAC. To use and to unify this is rather new.

Smaller experiments often use EPICS and MDS+ [15]. However, there is rare literature if the PLC is integrated and how this is done. It looks like the PLC SCADA system is more or less autonomous. On the other hand PLC based interlock systems exist but they are separated from the CODAC systems.

Why does it come PLC integration has been so rare? How can the situation in industry overcome this?

One reason for missing PLC integration is the fact that the responsibility for the PLC and the responsibility for CODAC is traditionally spread over different departments. PLC based systems have been treated as a piece of hardware from the CODAC point of view. PLC communication has been difficult and potentially error-prone e.g. using RS-232. If done at all, this was within the responsibility of the technician responsible for the electronics. The latest improvements in Ethernet (e.g. Fast Ethernet, switching, full duplex) leaded to the appearance of Industrial Ethernet (Ethernet for automation). Nowadays, PLCs are fully industrial Ethernet capable when attached with Ethernet modules.
However, Industrial Ethernet is not as standardized as the name might imply. [16] Describes that two third of Ethernet modules use standard Ethernet TCP/IP. The remaining third splits among numerous protocol variants e.g. EtherNet/IP, profinet, ModbusTCP/IP, EthernetPowerLink, EtherCat and others.
These protocols are not compatible to each other and a huge market exists for middleware between the various protocols e.g. OPC [17].

2. PLC interfacing for JET and W7-X

2.1 Used PLC types

W7-X solely uses PLCs of type Siemens Simatic S7 and does not accept any other PLC types.
JET as a joint experiment potentially allows multiple PLC types depending on the selection of the association that developed the component. It ranges from high end to low cost PLCs and also covers Soft PLCs (a PLC emulation running on a local host computer). See Table 1.

2.2 middleware/protocol

Vendors offer different protocols to interface a PLC depending on its type. Large vendors have their own protocols e.g. Siemens S7 communication, the niche vendors tend to use standardized protocols e.g. the modbus protocol in various dialects [16]. The Application Programming Interfaces (APIs) for these protocols vary between: closed source, open source, third party, C/C++, java, http, connectionless. Some PLCs are also offering multiple access methods. See Table 1.

PLC type	Protocol	Protocol is proprietary	PLC needs special pro-gramm	API	Middle-ware runs at	Used at	Known problems
Simatic S7	Siemens S7 Communication (UDP)	no	Yes	Java standard library (java.net)	PC with JVM (windows, linux)	W7-X	None
Simatic S7	Siemens S7 communication (UDP)	no	Yes	C socket API	PC with VxWorks	W7-X	None
Simatic S7 with CP 343-1 Advanced	S7 beans	yes	No	S7 beans API [18]	Solaris machine with JVM	JET	Needs high speed data link
Schneider Modicon Premium	unkown	yes	No	FactoryCast library (com. schneiderautomation. factorycast) [19]	Solaris machine with JVM	JET	None
Festo FC640	Festo command interpreter [20] over http	no	Yes	java third party library (apache http client)	Solaris machine with JVM	JET	Not yet known
IBHSoftec	Windows dll on localhost	yes	No	JNA Java Native Access [21] interface to plc32.dll	PC Windows (host where Soft PLC is running)	JET	SoftPLC freezes sometimes

Table 1: PLC types and the subset of protocols and APIs used to interface the PLC. Comparison and problems encountered

2.3 Protocols used within CODAS/CoDaC

The JET CODAS connection to a PLC is done using the JET http based black box protocol [22]. The concept of the black box protocol is to treat a component as a black box. It allows:

- Setting and reading back of plant equipment parameters,
- Reading of plant equipment state,
- Monitoring of plant equipment status,
- Setting up of plant equipment data channels for pulse data collection before a JET pulse and reading of the collected data after the pulse (not applicable for PLCs).
- Central logging or plant equipment error and warning messages.

In addition to the black box protocol some enhancements have been added to the PLC communication for test and convenience. They are planned to be added to future black box protocol versions.

The PLC connection is done using java allowing to running the middleware code platform independent. It is implemented as eclipse application and is completely data driven by XML and initialization files. Usually there is a single connection from the PLC to CODAS.

W7-X CoDaC communicates with a PLC using the Wendelstein7-X RtDataBus in the UDP option [23], [24] which is compatible to Siemens S7 communication. It allows:

- Continuous data streaming
- Reading measured values and set points
- Setting set points and parameters

- Preparing W7-X for segmented machine operation
- Getting status Information (e.g. operational state, health information)
- Switching between autonomous and subordinated mode

The PLC connection is completely data driven via configuration database entries. It is implemented twice for the two types of control stations:

1. In Java as part of the Data Acquisition stations
2. In C++ as part of the for Real time Fast Control Stations (running the real time operating system VxWorks)

Both implementations are in use and usually multiple connections to the PLC are alive for control and data acquisition purposes.

2.4 Integration of the PLC

The control and data acquisition software of JET and Wendelstein 7-X has more similarities than differences. However, they use different vocabulary and different structuring.

Figure 1 and 2 show schematically where the PLC is integrated into the control and data acquisition systems. Both figures show also the local human machine interface (HMI) allowing for remote operation.

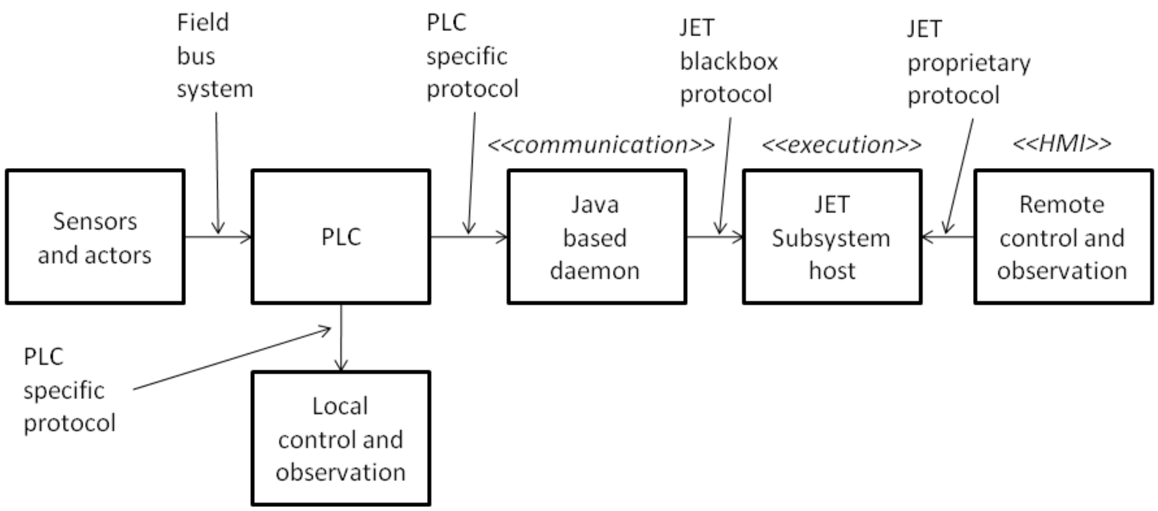

Fig. 1 Schema of PLC integration at JET with the hydra daemon

As shown in fig. 1 the black box approach separates the host computer from the actors and sensors. The additional hydra daemon wraps the black box protocol to the PLC specific protocol.

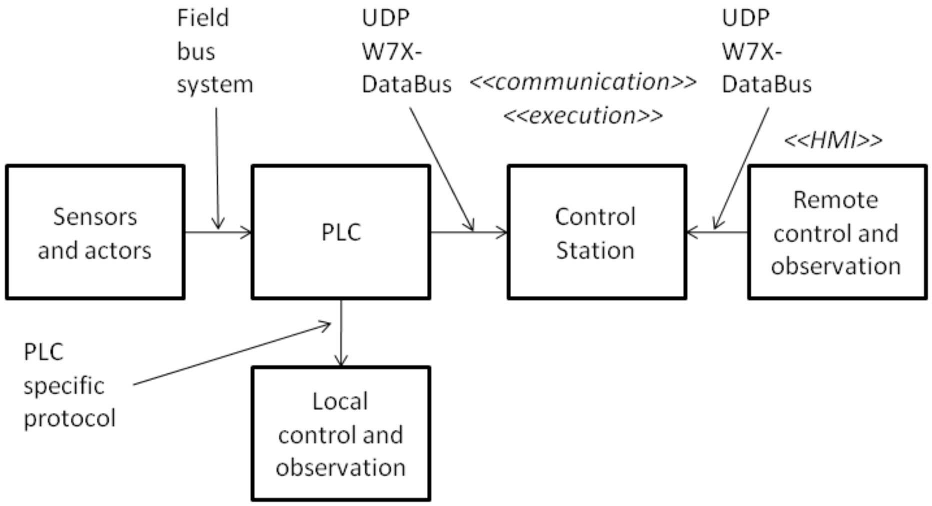

Fig. 2 Schema of PLC integration at W7-X

As shown in fig. 2 the PLC is able to communicate with the W7X DataBus protocol. In order to do so the PLC has to be programmed accordingly. This is well supported within the Simatic Step7 software.

2.5 Mapping machine / diagnostics components and PLCs

Some machine or diagnostics components have complex control programs or are physically distributed. Therefore multiple PLCs can be in use per component.

The default strategy is to identify a master PLC and communicate from CODAC to the master PLC only. The master PLC communicates to the client PLCs using proprietary PLC internal communication which is seamlessly integrated into the PLC programming environment. Drawback of this strategy is the single point of failure. This risk can be taken because the PLC hardware and network infrastructure is extremely robust. Moreover, cascading allows using cheaper PLCs of the same vendor behind the master PLC (e.g. Simatic models without advanced communication processor). Cascading is not the only option.

For the JET Neutral beam enhancement system an alternative approach has been taken. There was the requirement to minimize changes to the existing CODAS framework application which was already spread over multiple components. This resulted in multiple connections to multiple PLCs for the component.

Multiple components may share a PLC if a dedicated PLC per component is not economical. One can try to hide multiple vendors behind known PLC types as long as the PLCs are able to communicate with each other.

One can try to hide multiple vendor PLCs behind known, stable and proven PLC vendors and connections.

2.6 Performance and reliability

The connection of the PLC to the CODAS resp. CoDaC framework should be nearly as reliable and fast as the PLC itself. It proofed that all implementations are fast and reliable over months. Problems found were due to slow network connection and failures in the SoftPLC or in the host operating system.

PLCs are used for slow control only. Typical PLC cycle times are 100ms to 1second. Thus, performance requirements are rather relaxed. Potential bottlenecks are

1. The network connection. A problem occurred with the solution [18] when the existing 10Mbit/s connection was shared with interactive internet access during tests. The symptom was a loss of connection. An upgrade to a 1Gbit/s connection solved the problem. Observation of the connection was introduced and a watchdog added.
2. Velocity of the PLC. PLC hardware is usually i386 PC hardware in a reliable industrial proven version. If the PLC is already at its limit with its internal control tasks (combined with a relative high cycle frequency) it might not be able to send and receive data within every control cycle. PLCs work deterministic, Ethernet not. Hence the PLC program must be able to cope with missing or delayed data.
3. Performance of the communication software at the CODAC machine. Java based software in use with [18] and [24] works reliable and without failure detected during uptime of the machine (magnitude of months). C++ based software in use with [23][24] is as good as the java based.

The PLC itself is a proven industrial technology stable over many months (in factory automation over years). If the Ethernet connection breaks down there must be strategies to re-establish connection oriented communication. Main bottleneck is the computer that communicates with the PLC. The computer operating system requires upgrades and reboots accordingly.

Meantime between failures (MTBF) is usually in the range of the computer reboots. The communication software runs without any problem for many months. When using a connection oriented protocol the communication must be observed. Communication losses are in the order of network outages. The PLC [20] has not been operative so far, hence the MTBF is unknown. The PLC emulation [21] does not work stable. It freezes without notification for unknown reasons. Therefore the strong recommendation is to rely on PLC hardware and do not emulate PLCs by PCs for reliable tasks.

2.7 comparison of PLC integration in JET and W7-X

Both the JET and W7-X approach work quite similar: The PLC communicates with the Remote Control and Observation (see fig. 1 and 2). For JET the communication is done with the hydra daemon. The execution of commands is done in the subsystem. The daemon can run either on the subsystem host or any other host. For W7-X both command and execution are done within the control station software. The control station can be of type data acquisition or real time fast control.

Main differences are the protocols used. One can imagine to replacing one protocol by another one with any of these approaches. The JET black box separates the execution from the communication with external signals, thus being more flexible to interface third party systems as long as they implement the JET black box protocol which is the price for the flexibility.

For future experiments it is indispensable to standardize on a communication protocol. For ITER the suggested protocol is EPICS-IP. To interface the PLC either the PLC needs to be able to communicate the selected protocol or a middleware that converts between the PLC and the protocol selected has to be implemented. The first one is true for the Simatic S7 PLCs for both Epics and S7 UDP in use for W7-X. The latter is done with the hydra daemon at JET.

3. Summary and Conclusions

The paper gives an overview of the current activities for PLC integration into control and data acquisition systems for fusion devices.

To interface PLCs and thus making the control and data acquisition tools usable for components containing PLCs is an opportunity to bridge this gap. Benefits are:

- remote control the PLC from the control room with the same tools as any other control and data acquisition system, hence ease user interaction
- archive PLC data together with the operational and/or physics data

- communicate read/write and read only variables, i.e. set points and control variables/measured values which allows to control the PLC

Two approaches for connecting PLCs to fusion CODAC software frameworks have been introduced. The paper explains the details of the Jet and W7-X approaches and compares them. It also shows the relevance and open issues for ITER CODAC.

At W7-X all components use the same PLC type. Standardization of the PLC type eases the maintenance of the software.
At JET different PLC types are in use. However, all PLCs can be interfaces with a hydra daemon that hides the PLC specific part from the CODAS framework via the http black box protocol. New PLC types might need more implementation work. The existing implementations at JET show a wide variety of distributions. Nevertheless, JET could benefit from standardizing the PLC types wherever possible.
Both approaches for JET and W7-X are fully data driven. The use of java whenever possible allows a flexible and platform independent installation and distribution.
The PLC connections are extremely stable and reliable. They are nearly as stable as the host computer operating system. Any problems found are reported in table 1.

The paper discusses the potential to unify the approaches towards a common fusion PLC interface. This would require agreement on a single interface or protocol.

4. References and Resources

[1] http://www.jet.efda.org/jet
[2] http://www.ipp.mpg.de/ippcms/eng/for/projekte/w7x/index.html
[3] https://secure.wikimedia.org/wikipedia/en/wiki/Programmable_logic_controller
[4] Mrudula Ghone, Mark Schubert, and John R. Wagner: Development of a Mechatronics Laboratory—Eliminating Barriers to Manufacturing Instrumentation and Control; IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 50, NO. 2, APRIL 2003
[5] R Garbil, et al: The Tore Supra Cryoplant control system upgrade; Fusion Engineering and Design Volumes 58-59, November 2001, Pages 217-223
[6] H. Nakanishi, M. Kojima, S. Hidekuma: Distributed processing and network of data acquisition and diagnostics control for large helical device (LHD); Fusion Engineering and Design Volume 43, Issues 3-4, January 1999, Pages 293-300
[7] K.H. Kim, et al: The KSTAR integrated control system based on EPICS;
Fusion Engineering and Design Volume 81, Issues 15-17, July 2006, Pages 1829-1833
[8] Experimental Physics and Industrial Control System; http://www.aps.anl.gov/epics/
[9] Mahajan, Kirti et al: CORBA-based solution for remote participation in SST-1 tokamak control and operation; Fusion Engineering and Design 81 (2006) 2069-2072
[10] Gerhard Raupp, et al: ASDEX Upgrade CODAC overview; Fusion Engineering and Design Volume 84, Issues 7-11, June 2009, Pages 1575-1579
[11] D Zasche et al: Automatic setting of machine control with physics operation parameters at ASDEX Upgrade; Fusion Engineering and Design
Volume 48, Issues 1-2, 1 August 2000, Pages 3-8
[12] Young, ID: Possible Use of EPICS in Large Fusion Experiments ; CODAS JDN/H(06)053 Issue 1: 13/05/07
[13] F. Di Maio, et al: DEVELOPMENT OF THE ITER CODAC CORE SYSTEMS; Proceedings of ICALEPCS2009, Kobe, Japan
[14] D. Mastrovito et al: Integrating EPICS and MDSplus; Fusion Engineering and Design 81 (2006) 1847–1851
[15] http://www.mdsplus.org
[16] http://www.industrial-ethernet.org/ethernet-protocols-technologies/standard-ethernet-tcpip-dominates-industrial-installations/
[17] OLE (Object Linking and Embedding) for Process Control; http://www.opcfoundation.org/
[18] Siemens Manual SIMATIC NET S7Beans / Applets for IT-CPs C79000-G8976-C180-02; http://support.automation.siemens.com/AU/view/en/22964892
[19] http://www.schneider-electric.com
[20] Festo Manual: The CPU I/O Module FEC FC34-FST Manual FEC FC34-FST

[21] JNA java native access (download via https://jna.dev.java.net/)
[22] Hogben, C. and Griph, S.: Interfacing to JET Plant Equipment Using the HTTP Protocol; JDN/H(02)011; Issue 2 : 14th August 2008 (http://www.iop.org/Jet/fulltext/EFDR02004.PDF)
[23] H. Laqua, H. Niedermeyer, I.Willmann, Ethernet based real time control data bus, IEEE Trans. Nucl. Sci. 49 (2002) 478–482.
[24] Ch. Hennig et al: Using the network as bus system for long discharge data acquisition and data processing; Fusion Engineering and Design 81 (2006) 1735–1740