CODAC STANDARDISATION OF PLC COMMUNICATION

Sopan Pande#, Franck Di Maio, Bruno Evrard, Kirti Mahajan, Antoni Simelio,
Anders Wallander, Izuru Yonekawa,
ITER Organization, Route de Vinon sur Verdon, 13115 Saint Paul-lez-Durance, France
Prasad Sawantdesai, Tata Consultancy Services, Pune 411 057, India

Abstract

As defined by the CODAC (COntrol, Data Access and Communication) Architecture of ITER, a Plant System Host (PSH) and one or more Slow Controllers are connected over Switched Ethernet network. Siemens PLCs (Programmable Logic Controller) are standardized by CODAC as the Slow Controllers. An important part of Software Engineering for Slow Controllers is the standardization of communication between PSH and PLCs. Siemens PLCs offer various communication services such as OpenIE, Send/Receive and S7 Communication which are supported over transports such as TCP, ISO-On-TCP (RFC1006). Based on prototyping and performance evaluation, OpenIE communication over TCP was selected as it offers best performance and is also easy to use and configure. OpenIE Communication is used to implement the communication on PLCs to support the CODAC data model. The implementation is packaged with the Standard PLC Software Structure (SPSS) which is a part of official CODAC Core System release. SPSS can be easily configured by the SDD (Self-Description Data) Tools supporting PLC integration with EPICS IOCs. However, OpenIE is restricted to the PROFINET interface of PLC-CPUs and is not available on PLC-CPs (Communication Processor). Thus OpenIE can't be used for redundant PLCs and when the PROFINET connection is needed for remote IOs. So another version of SPSS developed with Send/Receive communication is used to support communication over CPs. The S7plc EPICS driver is extended to support CODAC requirements and to support redundant PLCs. This paper describes PLC communication standardization in the context of CODAC environment. Future developments of EPICS driver to support the ISO-On-TCP and S7 communication are presented briefly.

INTRODUCTION

To ensure the integrated operation of large number of ITER plant systems, CODAC has laid down uniform standards, methodologies for the Instrumentation & Control (I&C) in the Plant Control Design Handbook (PCDH) [1]. Overall I&C architecture [2] is defined with building blocks such as the PSH, Slow Controllers and Fast Controllers communicating over switched Ethernet (Fig.1). The PSH is a Red Hat Linux platform executing EPICS IOCs (Input Output Controller) and the Slow Controller is one of the SIEMENS PLCs from the ITER catalogue [3].

#sopan.pande@iter.org

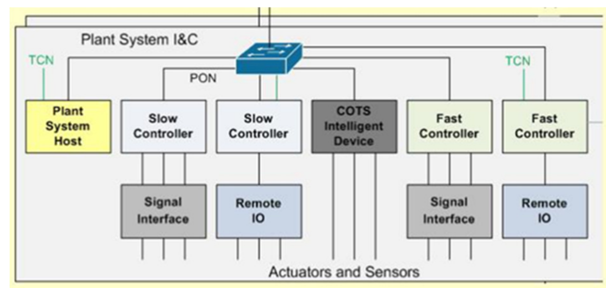


Figure 1: Plant System I&C Architecture.

CODAC standards are effectively implemented by the software tools and components of CODAC Core System v 4.1 (CCS) [4, 5]. Siemens PLC communication with EPICS IOC is one of the standard components of CCS. It is implemented by the *CODACInterface* on PLC as a part of SPSS [6] and the extended S7plc EPICS Driver which are components of CCS. Together they implement a CODAC plant system data model (Fig.2). SDD Tools of CCS namely the SDD Editor or SDD Web Application generates plant system I&C applications to easily configure these components.

CODAC standardization of PLC communication will be used for all ITER plant systems. Development, challenges and evolution of these components is described in following sections.

CODAC PLANT SYSTEM DATA MODEL

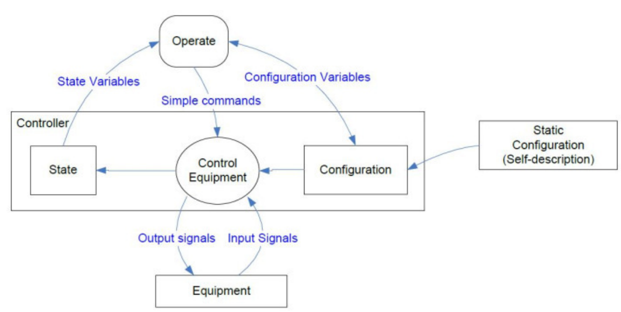


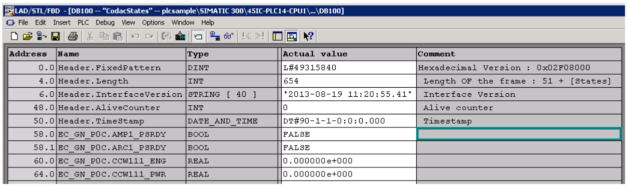
Figure 2: CODAC Plant System Data Model.

CODAC has defined the plant system data model with buffers named *State*, *Configuration* and *SimpleCommands* [7]. The *State* consists of all plant system process variables and timestamp sent periodically from PLC to PSH. The *Configuration* consists of all set-points and other values sent from PSH to the PLC. And the *SimpleCommands* is used to generate specific events on PLC from PSH.

CODAC performance requirement for *State* buffer is 8KBytes every 50ms. A specific structure is imposed on

ISBN 978-3-95450-139-7

the *State* to include a time stamp and to detect communication errors (Fig.3). The *State* has a 58 byte header and 4 byte footer with all plant system variables in between. Fixed values of header, footer and length are used to confirm the consistency of communication. A unique version identifier generated by SDD Editor is used to compare and ensure a unique connection between an IOC and a PLC. PLC time in the header provides the time-stamp for all the process variables.


LAD/STL/FBD - [DB100 -- "CodacStates" -- plcsample\SIMATIC 300\45IC-PLC14-CPU1\...\DB100]

File Edit Insert PLC Debug View Options Window Help

Address	Name	Type	Actual value	Comment
0.0	Header.FixedPattern	DINT	L#49315840	Hexadecimal Version : 0x02F08000
4.0	Header.Length	INT	654	Length OF the frame : 51 + [States]
6.0	Header.InterfaceVersion	STRING [40]	'2013-08-19 11:20:55.41'	Interface Version
48.0	Header.AliveCounter	INT	0	Alive counter
50.0	Header.TimeStamp	DATE_AND_TIME	DT#90-1-1-0:0:0.000	Timestamp
58.0	EC_GN_P0C.AMP1_PSRDY	BOOL	FALSE	
58.1	EC_GN_P0C.ARC1_PSRDY	BOOL	FALSE	
60.0	EC_GN_P0C.CCW111_ENG	REAL	0.000000e+000	
64.0	EC_GN_P0C.CCW111_PWR	REAL	0.000000e+000	

Figure 3: *State* Header.

The CODAC data model is implemented by *CODACInterface* on a PLC and S7plc EPICS Driver Extensions (Fig.4) on a PSH as described in following sections.

CODACINTERFACE ON PLC

Communication services of Siemens PLCs [8] namely the OpenIE, Send/Receive and S7 were prototyped and evaluated to select the best communication service for the PLCs (Table1).

Table 1: Siemens PLC Communication

Siemens PLC Communication Services		SEND/ RECEIVE	OpenIE	S7
Transport		TCP	TCP	ISO-On-TCP
Available on CP		Yes	No	Yes
Available on CPU		No	Yes	Yes
Channel Configuration Required		Yes	No	No
PLC Application Programming Required		Yes	Yes	No
Portability for s7-300 and S7-400		No	Yes	Yes
Maximum Block Size	S7-300	8192 Bytes	8192 Bytes	8192 Bytes
	S7-400	240 Bytes	32768 Bytes	8192 Bytes
Transmission Time for 8192 Bytes	S7-300 CPU	-	25 ms	230 ms
	S7-300 CP	-	-	120 sec
	S7-400 CPU	-	12 ms	120 ms
	S7-400 CP	134 ms	-	130 ms
EPICS Driver Available		S7plc	S7plc	No
Reliability		High	Low	High
Open Source Library Available		-	-	LIBNODAVE

OpenIE over TCP offers best performance and is very easy to configure but it is available only on CPUs and not on CPs. Communication over CPs is necessary for redundant PLC configurations and also where the Ethernet connection on the CPU is used for the PROFINET remote IOs. In such cases it is necessary to use Send/Receive for communication with PSH over CP. Send/Receive performance is slower than OpenIE and requires communication channels to be configured in STEP7. S7 Communication is also a very useful option for as it is supported by all Siemens PLCs with no programming in STEP7. It uses the ISO-On-TCP (RFC1006) transport and is very much slower than OpenIE or Send/Receive.

The SPSS [6] of CODAC is the starting point of PLC software development. It provides a common program structure for PLC software of all ITER plant systems. As a part of SPSS the *CODACInterface* implements the PLC end of CODAC data model. The *CODACInterface* is implemented with OpenIE and Send/Receive both. The required mechanism is chosen by appropriately compiling the SPSS. The SDD editor generated PLC files are imported and compiled in STEP7 to generate the *State*, *Configuration* and *SimpleCommands* DBs (Data Blocks) and configure *CODACInterface* to communicate these DBs with PSH (Fig.4).

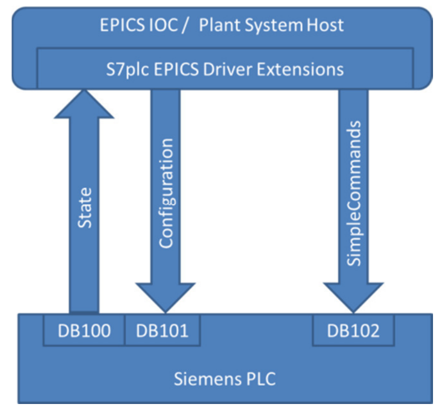


Figure 4: PLC Communication Standardization.

S7PLC EPICS DRIVER EXTENSION

S7plc [9] is a widely used open source EPICS driver that implements communication with Siemens PLCs for the EPICS IOCs supporting all PV record types. S7plc periodically receives a data buffer from the PLC to update all the input PVs mapped on this buffer. Similarly it sends the output buffer to PLC when any of the output PVs change.

S7plc is extended to support the CODAC data model with time stamping and error detection. The *States* and *Configuration* are mapped respectively on the input and output buffers on IP port 2000 of PLC. *SimpleCommands* are mapped on the output buffer on port 2001 of PLC.

State buffers received by the receive thread of S7plc are checked for correctness before they could be used for record processing. Any error with header, footer, length or version string is reported as communication alarm. All input PVs are assigned the timestamp from the *State* header. Difference in successive timestamps is monitored to detect *State* frames that may be lost or delayed in PLC.

S7plc is also extended to support safety or high availability PLCs operating in hot-standby mode where both the CPU/CPs of the redundant configuration simultaneously send same *State* buffer to the PSH. Duplicate *State* buffers from redundant CPUs are ignored by the S7plc.

Due to delay in TCP ACK (acknowledgement) from PSH to PLC, occasionally a few buffers get queued in the PLC. The queued buffers are received much faster than

ISBN 978-3-95450-139-7

S7plc could process them completely with the non-blocking `scanIoRequest` (EPICS call to trigger record processing). So `scanIoRequestBlocking` (blocking version of `scanIoRequest`) is developed and used to ensure that the received *State* is fully processed by the S7plc before it receives next buffer.

FURTHER DEVELOPMENTS

Due to the ease of use and support by all Siemens PLCs, S7 Communication is a very good option to develop low performance features such as debugging and simulation.

LIBNODAVE [10] provides an open source implementation of S7 on top of ISO-On-TCP transport. S7 implements a proprietary interface to access any Siemens PLC DB or memory resources. S7 and ISO-On-TCP are extracted from LIBNODAVE to implement an ASYN device support. ASYN [11] defines a very good framework for modular architecture of EPICS device support.

Asyn based PLC driver support is proposed and prototyped using standard EPICS IP port drivers, an interpose layer and EPICS record processing layer (Fig.5) [12]. With a separated interpose layer it possible to add, remove or replace any protocol with no changes in the IOC.

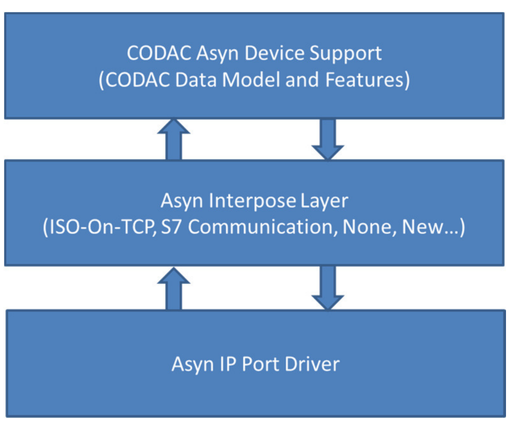


Figure 5: Asyn PLC Driver Support.

S7 and ISO-On-TCP protocols are implemented as interpose layer. In case of OpenIE or Send/Receive communication with TCP transport, the interpose layer could be ignored completely. Thus it is possible to have any protocol or no protocol as required. This Asyn PLC driver is being used to develop a prototype CODAC plant system application.

CONCLUSIONS

The *CODACInterface* implemented with OpenIE and Send/Receive is available as a part of SPSS. The S7plc EPICS Driver extensions support all CODAC features and redundant PLCs. Both components which implement standard PLC communication for CODAC I&C applications are part of official CODAC Core System version 4.1 [5], making it is possible to support CODAC data model for all Slow Controller architectures.

S7 and ISO-On-TCP protocols are prototyped based on ASYN architecture. They are not part of the official release but are being used for special communication requirements.

DISCLAIMER

The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

REFERENCES

[1] Plant Control Design Handbook (PCDH), https://www.iter.org/org/team/chd/cid/codac/plantcontrolhandbook
[2] Plant System I&C Architecture.
[3] ITER catalogue for Slow controllers PLC.
[4] CODAC Core System Overview.
[5] F. Di Maio et al., "CODAC Core System, the ITER software distribution for I&C", Proc. of ICALEPCS 2013, San Francisco, http://jacow.org
[6] PLC Software Engineering Handbook.
[7] The CODAC - Plant System Interface.
[8] STEP7 – Programming with Step7, http://www.automation.siemens.com
[9] S7plc EPICS driver documentation, http://epics.web.psi.ch/style/software/s7plc/s7plc.html
[10] LIBNODAVE - Exchange data with Siemens PLCs, http://libnodave.sourceforge.net/
[11] asynDriver: Asynchronous Driver Support, http://www.aps.anl.gov/epics/modules/soft/asyn/R4-18/asynDriver.html
[12] Driver Support for Modbus Protocol under EPICS, http://cars9.uchicago.edu/software/epics/modbusDoc.html

ISBN 978-3-95450-139-7