ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA080

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

A COMMUNICATION PROTOCOL FOR MOTION CONTROL APPLICATIONS AT THE JCNS NEUTRON INSTRUMENTS

H. Kleines†, F. Suxdorf, Forschungszentrum Jülich, Germany

Abstract

Main focus of slow control in neutron scattering is motion control for the movement of around 25 mechanical axes in a typical neutron instrument. The implementation of motion control functions in the JCNS neutron instruments at the FRM II research reactor in Garching, Germany, is based on Siemens S7 PLCs. A communication protocol called PMcomm which is optimized for motion control applications in neutron instruments has been developed at JCNS. PMcomm (PROFI motion communication) is based on PROFINET or PROFIBUS as the underlying transport protocol in order to facilitate the easy integration into the PLC world. It relies on the producer/consumer communication mechanism of PROFINET and PROFIBUS for the efficient direct access to often-used data like positions or status information. Coordinated movement of groups of axes is facilitated by a generic controller/axes model that abstracts from the specifics of the underlying motion control hardware. Simplicity was a major design goal of the protocol in order to allow an efficient and easy implementation on PLCs.

INRODUCTION

JCNS, the neutron science division of Forschungszentrum Jülich, developed and operates 15 neutron instruments at its outstations ILL in Grenoble, the Spallation Neutron Source in Oak Ridge and at the FRM-II in Garching near Munich. Further neutron instruments are being developed for the future European Spallation Source in Lund. The control and data acquisition systems of these neutron instruments are responsible for a variety of tasks including detector readout, vacuum systems, cryogenic systems, sample environment devices and motion of many mechanical axes.

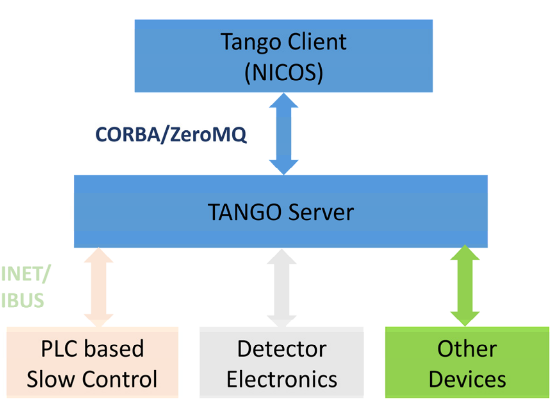

Figure 1: Jülich-Munich Standard System Architecture.

In order to reduce the overall development effort und the number of spare parts, these control systems are highly standardized and follow a common framework, the so-called Jülich-Münich standard [1], that evolved gradually over a period of more than 20 years. Originally being based on the TACO framework, it is now based on the successor TANGO [2], as shown in Fig. 1. Main client application is the standardized measurement program NICOS [3] providing a GUI interface as well as a scripting interface for scan definition and execution. All slow control tasks - especially motion - are implemented with industrial PLC technologies including fieldbus communication and decentral peripheral systems in the frontend. Main motivations for this approach are:

- low prices induced by mass market,
- inherent robustness,
- long term availability and support from manufacturer,
- powerful development tools.

Because of its strong market position, Siemens has been selected as the PLC vendor for all JCNS instruments.

In the past the communication of PLCs with peripheral devices and with supervisory computers was implemented with PROFIBUS DP, since this field bus standard was naturally supported by Siemens PLCs. In the recent years Ethernet-based industrial communication systems like PROFINET IO [4], Ethernet/IP, Ethercat, Powerlink or Modbus TCP became increasingly important for the communication with industrial devices and today PROFINET IO (see section III) is the most commonly used industrial Ethernet system and inherently supported by all PLCs from Siemens. As a consequence, all new developments for JCNS instruments are based on PROFINET IO and PROFIBUS is being replaced by PROFINET in older instruments.

For the communication between motion related PLCs and controlling computers running the corresponding TANGO servers, a dedicated application protocol called PMComm (PROFI Motion Communication) has been developed. Originally designed for PROFIBUS DP it supports also PROFINET IO, since both fieldbus systems share an identical communication model, requiring just a minor software adaptation on the PLC side as well as on the host side.

PROFINET IO

Originally defined by PI (PROFIBUS and PROFINET International), PROFINET IO is now internationally standardized in IEC 61158 and IEC 61785. It is based on 100 Mbit/s Ethernet supporting wireless, copper and optical media in full conformance to the corresponding IEEE standards. As a consequence, a PROFINET communication interface can be used simultaneously for PROFINET real time communication as well as for standard internet

† h.kleines@fz-juelich.de

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA080

communication – e.g. for web services. PROFINET IO defines several conformance classes in order to support soft real time as well as hard real time communication, called IRT (isochronous real time) mode. In IRT mode, communication cycle times down to 31.25 µs with a jitter of less than 1 µs are supported. IRT mode relies on PTP (Precision Time Protocol) according to IEEE 1588 for time synchronization.

In order to facilitate the easy migration from PROFIBUS to PROFINET, it shares basic architectural principles with PROFIBUS DP. Similar to PROFIBUS DP, PROFINET IO aims at application scenarios, where a central station communicates with decentral field devices. As a consequence, each station in a PROFIBUS IO system can take one of the following roles according to Fig. 2:

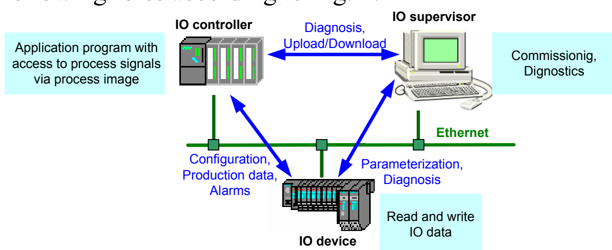

Figure 2: PROFINET IO communication model.

- IO controller: The IO controller represents an intelligent central station, like a PLC. It is responsible for the configuration and parameterization of its associated devices and controls the transfer of process data.
- IO device: The IO device represents a field device, like an analogue input unit. It cyclically transmits collected process data to the IO controller and vice versa. It also provides diagnostic or alarm information to the IO controller.
- IO supervisor: The IO supervisor represents an engineering station for programming, configuration or diagnostics, e.g. a PLC programming tool.

Similar to PROFIBUS DP, PROFINET IO is based on a consistent model of the IO device structure and capabilities, described by so-called GSDML files. An IO device may be modular and is composed of one or more so-called slots, which may have subslots. Each slot or subslot represents an IO module and has a fixed number of input and output bits. The input data of the IO device is the sequence of all inputs of slots and subslots, according to their position in the device. The same holds for the output data. Also all diagnostic or alarm data reference slots or subslots.

Similar to PROFIBUS DP, the transfer of process data relies on a consumer/producer model with cyclic data exchange between IO controllers and their IO devices. As illustrated in Fig. 3 the transmission occurs from and to predefined IO areas. These IO areas are read and written by the application processes in IO controller and in IO device synchronously or asynchronously to the communication cycle. This memory-based communication model corresponds in a very natural way to internal PLC operation and makes software implementation of data exchange extremely simple.

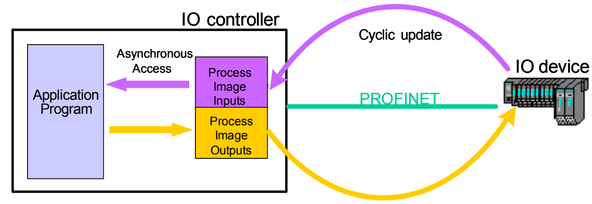

Figure 3: Cyclic message exchange.

IMPLEMENTATION OF PROFINET IO FOR SUPERVISORY COMPUTERS

For the implementation of the PROFINET IO interface in supervisory computers a variety of controllers for PCI, CompactPCI and PCIe with Linux support are available from different vendors. Additionally, there are industrial computers on the market with PROFINET IO directly integrated on the main board as well as pure software implementations for standard Ethernet network interface cards, e. g. the Siemens product PROFINET Driver in source code, which also supports Linux.

For ComapctPCI, which is required for the JCNS instruments, several well established vendors – like Hilscher or Kunbus – offer PROFINET IO controller implementations. Since PROFINET IO requires a consistent configuration of the controller (in our case the PC) and the device (in our case the PLC) we would have to use tools from different vendors for this purpose, which is quite error-prone. So we intended to use a Siemens product also for the controller, which could be configured directly from the TIA Portal PLC project. Since no CompactPCI controller is available from Siemens, we selected the product CP1604 in PC104+ formfactor. The CP1604 supports PROFINET IO controller as well as device functionality and comes with Linux support, including device driver, utilities and examples programs in source code.

For the integration of the CP1604 we implemented a generic CompactPCI base board for PC104+ mezzanines, as shown in Fig. 4. Core of the board is the IDT TSI 350 PCI bridge chip, supporting 32 Bit / 66 MHz PCI. As shown in Fig. 5, the 4 port RJ45 connector required for CP1604 has been directly integrated on the base board and is interfaced via a connector to the mezzanine, thus avoiding a ribbon cable. Due to the PC104+ connectors, two CompactPCI slots are required for a standard mezzanine. For single slot configurations, we exchanged the female/male connectors on the CP1604 by a single female connector.

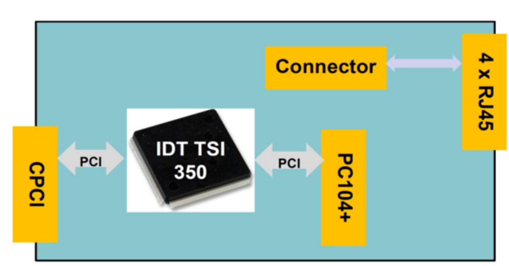

Figure 4: Simplified block diagram of the new CompactPCI carrier for PC104+.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA080
Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Figure 5: Photo of the new CompactPCI carrier for PC104+.

PMCOMM CONCEPTS

PMcomm is designed for scenarios, where the motion hardware (motion controllers, encoder readouts,…) is attached to a PLC, either locally as PLC modules or remotely, e.g. as decentral periphery systems via a fieldbus, as shown in Fig. 6.

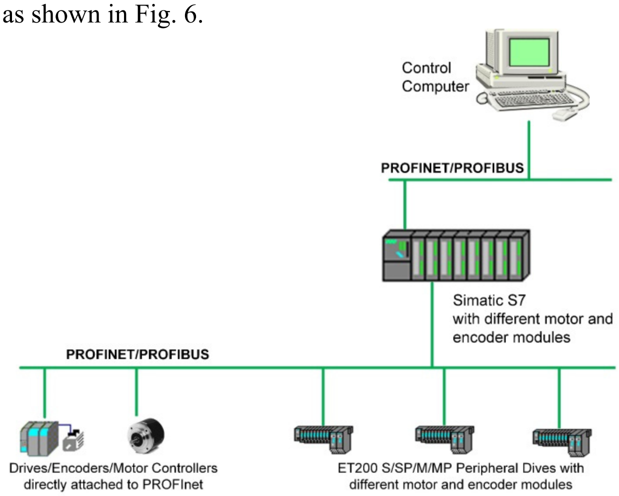

Figure 6: PMcomm hardware configuration scenario.

PMcomm has been defined for the communication between the PLC and its controlling computer. It defines an abstract controller/axis model for motion tasks, thereby abstracting from the specific motion hardware which is actually used. Each axis resides in a controller and a controller may have several axes, in order to support systems with synchronized axes. Execution of a controller start service will synchronously move all subordinate axes according to a predefined functional relation between these axes. Services defined by PMComm include standard motion functions like referencing, stopping, relative and absolute movement of axes. Additionally, there are services for diagnostics, state information, configuration retrieval and change. Services for synchronized movement have rarely been used and the download service for NC data has never been used.

The PLC application program is responsible for the mapping between the abstract controller/axes model to the specific motion hardware. Additionally, it may implement application functions or safety interlocks which are hidden from the controlling computer. PMcomm is based on PROFIBUS DP or PROFINET IO, since these protocols are naturally supported by Siemens S7 PLCs and provide a very simple programming interface on the PLC side as well as on the Linux side. For the same reason JCNS exclusively uses PROFIBUS DP and PROFINET IO for the decentral distribution of motion hardware in the field. But this is completely transparent to PMcomm and in principle other interfacing techniques could be used for decentral motion devices.

PMCOMM PROTOCOL

In PMcomm the computer always takes the role of the PROFINET IO controller (or PROFIBUS master) and the PLC takes the role of the PROFINET IO device (or PROFIBUS slave).

PMcomm relies heavily on the producer/consumer communication model of PROFIBUS and PROFINET. This allows the permanent mapping of important data like axis positions and states into the IO areas described above, as shown in Fig. 7. Thereby, efficient accesses to these data are possible via memory read and write operations which are completely asynchronous to the operation of the PLC program. For less often used services, a transaction mechanisms has been defined for sending requests to the PLC and for receiving confirmations from the PLC. As shown in Fig. 7, a reserved 32 Byte space has been reserved at the beginning of the IO areas for this purpose.

Additional IO data not directly related to motion - like analogue and digital IOs - can be directly mapped into a reserved space at the end of the IO areas.

Due to the limited IO area size of PROFIBUS (244 Bytes) a maximum of 40 axes can be supported in a PROFIBUS system. If more axes have to be supported by a single PLC, it as to be extended by an additional PROFIBUS communication controller. Due to the extended IO space of 1.5 kBytes, this is not an issue anymore in PROFINET IO.

Generally, the asynchronous, memory-based approach of PMcomm provides a simple and efficient programming interface on the PLC side as well as on the computer side.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA080

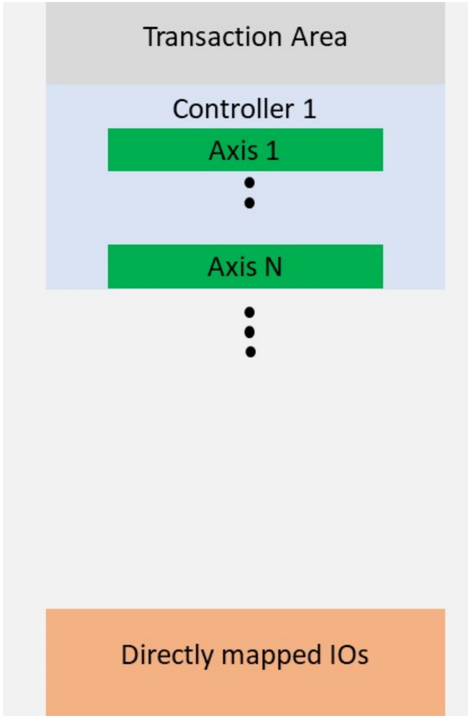

Figure 7: Layout of the PROFINET/BUS IO areas.

CONCLUSION AND OUTLOOK

PMcomm has successfully been used for the communication with motion PLC in all JCNS instruments for about 20 years and provides an easy and efficient access to motion functions. During this time it continuously evolved and was extended by an interface to PROFINET.

Because of the limited IO area space of PROFINET the directly mapped data is much squeezed. The additional IO area space of PROFINET will allow extended functionalities. Therefore a major redesign of PMcomm is intendend, with the following goals:

- Better self-description of motion devices, e.g. axis names, axis units.
- Improved performance by mapping an increased amount of data directly to the IO areas, thus reducing the need for slower transactions.
- Support of 64 bit position data instead of just 32 bit.

REFERENCES

[1] H. Kleines *et al.*, "Implementation of the Control and Data Acquisition Systems for Neutron Scattering Experiments at the New Jülich Center for Neutron Science According to the Jülich-Munich Standard", in *Proc. 10th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'05)*, Geneva, Switzerland, Oct. 2005, paper P1_104.

[2] Official TANGO homepage, `http://www.tango-controls.org`

[3] T. Unruh, "Instrument Control at the FRM-II using TACO and NICOS", in *Proceedings of the Nobugs 2002*, Gaithersburg, 2002.

[4] International Electrotechnical Commission, "Digital data communication for measurement and control - Fieldbus for use in industrial control systems", IEC 61158, 2015.

© Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.