PROFINET: An Integration Platform for heterogeneous Industrial Communication Systems

Juergen Jasperneite
Phoenix Contact GmbH & Co. KG
Dringenauer Strasse 30
D-31812 Bad Pyrmont
jjasperneite@phoenixcontact.com

Joachim Feld
Siemens AG
Gleiwitzer Strasse 555
D-90475 Nürnberg
joachim.feld@siemens.com

Abstract

Ethernet has become so popular that it has also been further developed for use at field level in automation technology under realtime conditions. However, for reasons of function and investment protection, the introduction of realtime Ethernet in automation solutions is likely to be a gradual process. Suitable solutions for the integration of existing fieldbus technology into the realtime Ethernet network are therefore required. This paper describes two possible integration concepts of the PROFINET IO realtime Ethernet standard using INTERBUS and PROFIBUS as an example. It shows that it is possible, despite very different device models of the considered fieldbus systems, to implement a high level of integration.

1. Introduction

For industrial communication within the automation domain, fieldbus systems and sensor networks have been standardized in IEC 61158. Another important current extension of fieldbus technology is the transmission of safety-oriented data. On the other hand, LANs (Local Area Networks) based on Ethernet TCP/IP are increasingly used in automation technology. This is due to a constant fall in prices in the IT market for Ethernet, the high bandwidth available, switching technology, and Ethernet-enabled products that meet industrial environmental requirements (e.g. [9, 11]). The use of Ethernet at field level in automation technology was previously opposed due to the lack of realtime capability. In the meantime, however, a range of realtime initiatives has been developed. One of the most promising realtime Ethernet standards is PROFINET [6].

However, for reasons of investment protection, the introduction of realtime Ethernet in automation solutions is likely to be a gradual process. It is therefore important that existing fieldbus systems can be integrated effectively into the Ethernet protocol. PROFIBUS ([10]) and INTERBUS ([3]), the two most important fieldbus systems for manufacturing industries, use PROFINET as an integration platform. This corresponds to a central requirement of the German Motor Industry Federation (VDA), which uses both fieldbus systems in its plants. The VDA asked automotive suppliers to develop common solutions based on Ethernet. In the following, two different integration concepts are described for the PROFIBUS and INTERBUS fieldbus systems. These integration concepts are generally also suitable for other fieldbus systems.

The paper is organized as follows: Section 2 and 3 give a brief introduction to PROFINET and the general Proxy architecture. Afterwards a description of the proxy architecture follows in section 3. In section 4 the implementation of the mapping from INTERBUS and PROFIBUS to PROFINET IO is presented. The paper ends with the conclusions.

2 Brief Introduction to PROFINET

PROFINET is an initiative to adapt Ethernet to the next generation of industrial automation [2]. PROFINET consists of several functionalities such as distributed automation (PROFINET CBA), decentralized field devices (PROFINET IO), network management,installation guidelines and web integration. All these different functionalities will help to make standard switched Ethernet [7, 8] easier to use in industrial automation. PROFINET IO enables the use of decentralized field devices with Ethernet. With decentralized field devices all automation devices including IO-devices, valve islands, frequency converters etc. can easily be used in a homogeneous network infrastructure. PROFINET IO defines three different types of functionality:

- The IO-controller is a controlling device which is associated with one or more IO-devices (field devices).
- The IO-device is a field device which is connected to the actors and sensors in the field level and delivers these data to the IO-controller.
- The IO-supervisor is an engineering device which manages the provision of configuration data (pa-

0-7803-9402-X/05/$20.00 © 2005 IEEE

rameter sets) and the collection of diagnostic data for/from IO-controllers and/or IO-devices.

PROFINET IO offers a cyclic exchange of more than 1000 inputs and outputs with 32 field devices in less than 1 ms. The data exchange is based on a provider/consumer mechanism. PROFINET IO allows a mono-controller or multi-controller operation, where more than one IO-controller can be connected to one IO-device. Also an isochronous mode with a jitter less than 1 microsecond signalling to the application is specified (Isochronous Realtime (IRT)). For engineering aspects of the field devices, PROFINET uses the xml-based device description GSDML (Generic Station Description Markup Language).

3 General Proxy Architectures

The concept of a fieldbus integration must take into account the following aspects:

- Exchange of cyclic process data, acyclic parameter and alarms
- Functional safety : The specifications of fieldbus systems are extended by the characteristics of safe data communication. This characteristic must be provided also in the combination of realtime Ethernet and Fieldbus.
- Diagnostics : All fieldbus systems have mechanisms for bus diagnosis. This characteristic must be preserved in a heterogeneous network.
- Determinism: The end-to-end latency must not exceed the requirements of factory automation (typical range of 5 to 10 ms).
- Bus configuration: Each field bus system has specific configuration parameters.
- Engineering aspects: The user demands only one configuration tool.

From the networking point of view the proxy represents a gateway between the Ethernet-based PROFINET system and the fieldbus system (see Fig. 1).

For the integration of a fieldbus system to PROFINET two modeling concepts exists:

- Multi-device approach: mapping fieldbus devices to multiple logical PROFINET IO-devices
- Slot/Sub-slot approach: mapping fieldbus devices to the model of a modular PROFINET IO-device

3.1 Multi-device approach

With the multi-device architecture of PROFINET-IO (see Fig. 2) it is possible to realize several logical IO-devices and IO-controllers within one physical device. To do this, some additional protocol elements are defined for

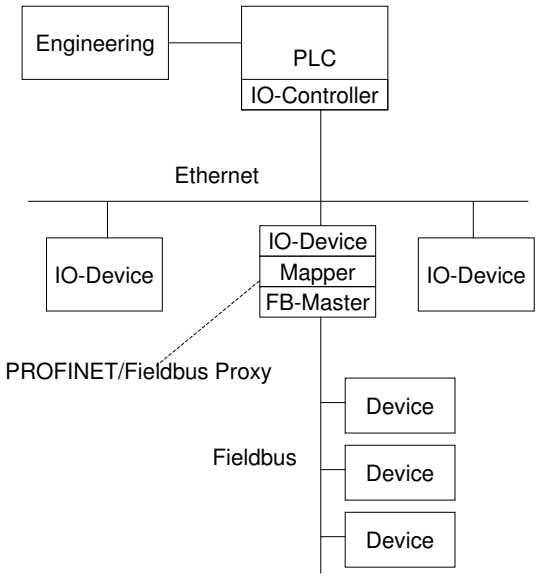
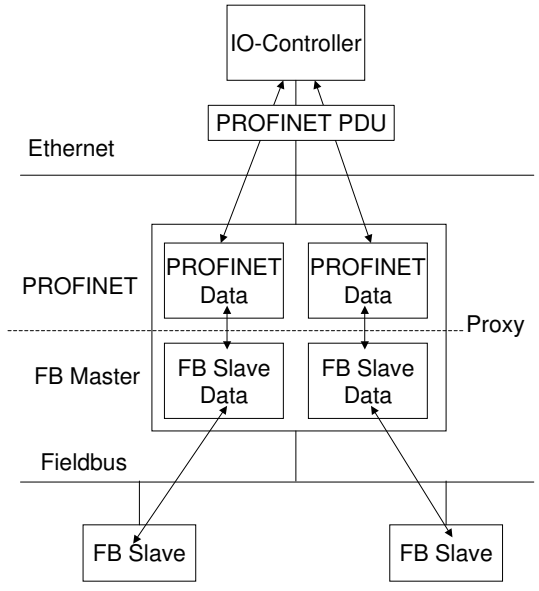


Figure 1. Fieldbus Integration to PROFINET

the context management, which is based on a connectionless Remote Procedure Call (RPC). The 16 byte protocol elements Interface-UUID, Type-UUI' and Object-UUID are defined for structuring the RPC interface and can be used for addressing different logical IO-devices or IO-controllers within one physical device. The disadvantage of this approach is, that for every logical device, a separate Ethernet frame will be used. The advantage of this ap-



FB: Fieldbus

Figure 2. Multi-device model of PROFINET IO

proach is, that especially if the fieldbus consists of the concept of modular slaves (like PROFIBUS does), the user's view of the devices does not change. This concerns espe-

cially the following aspects:

- Diagnostic: If diagnosis on the field bus devices is organized by means of slots and modules the user, e.g. in the PLC program, has the same view regardless if the device is directly coupled to the PLC via the fieldbus (e.g. PROFIBUS) or via PROFINET. The already available PLC programs and concepts for diagnosis can be reused.

- Maintenance: Because the geographical addressing scheme is not transformed through the link, all diagnosis information (especially geographical addresses) can be delivered to the user up to HMI systems without retransforming through the PLC. Example: Consider a PROFIBUS Device with at least 5 slots. If slot 4 is pulled this information can easily be transferred over the proxy. The corresponding slot 4 of the proxy IO-Device is plugged and the alarm is sent to the PROFINET IO-Controller. The status message sent to the HMI system *Slot 4 pulled* still considers the correct slot.

The Multidevice approach allows a mapping of every slot based device model on the fieldbus side of the link without loss of information.

3.2 Slot/Sub-slot approach

Another kind of modelling devices in PROFINET IO is the slot/sub-slot approach, which is used for modular slaves. The PROFINET IO-device is composed of different structural units to group application objects and provide a certain level of abstraction. These structural units may reflect hardware components or virtual functional units of the field device. The aim is to provide a suitable set of address parameters. These structural units are referred to as slots, sub-slots, and channels which may reflect also physical units, subunits, or a single connection point of an IO-device. Figure 3 shows the structure of this model. The structural units slot and subslot are accessed by means of slot number or subslot number within the PROFINET IO address model. Slot 0 in combination with a particular Subslot (e.g. 1) must always be used to represent the IO-device itself referred to as Device Access Point (DAP). A DAP contains the global data of an IO-device. Furthermore, the subslot Number 0 in combination with a particular Slot Number shall represent the dedicated module itself. This definition implies that a real subslot 0 does not exist and must not contain IO channels. A slot, in most cases slot 0, may contain up to 16 further special subslots referred to as Interface subslots. These subslots define the remote access to IP address parameters and the station name. A slot, in most cases slot 0, may contain for each interface up to 255 subslots referred to as port subslots. This approach uses only one Ethernet frame per device and is therefore more efficient than the multidevice approach concerning the communication aspects. The main disadvantage of this approach is that multiple modular devices on the fieldbus have to be mapped to one device in PROFINET. The same mapping that is applied in the proxy has to be applied on the PLC or the HMI in reverse order to reassign the e.g. diagnosis information to the user's view of his plant. Example: Device 5 Slot 6 on the Fieldbus line is mapped to Slot 35 on the PROFINET IO-device of the link. If Device 5 Slot 6 is pulled, the diagnosis information sent to the PROFINET IO-Controller (and furthermore to the HMI) is Slot 35. To repair the correct module, a reverse mapping from Slot 35 to Device 5 Slot 6 has to take place.

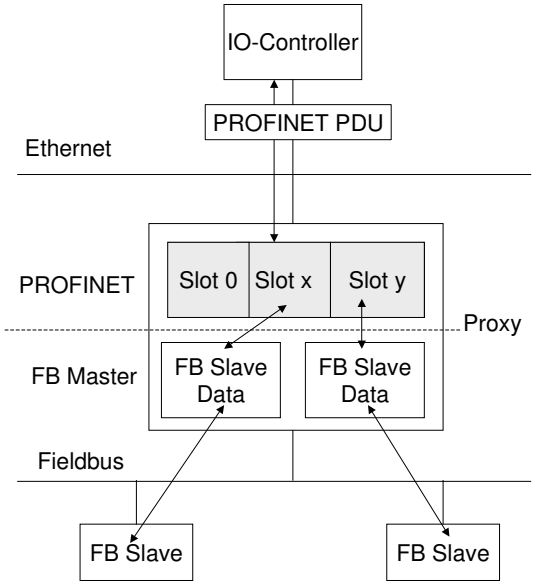


Figure 3. Slot/Sub-slot model of PROFINET-IO

4 Examples of realizations

In this section the two different mapping models are explained with examples of two implementations:

- INTERBUS - PROFINET Proxy using the slot/subslot approach

- PROFIBUS - PROFINET Proxy using the multidevice approach

4.1 INTERBUS PROFINET Proxy

In this section we describe the implementation of an INTERBUS proxy.

4.1.1 BRIEF INTRODUCTION TO INTERBUS

INTERBUS is an international standardized fieldbus system according to IEC-61158 (type 8) [4]. It has been designed as a fast sensor/actuator bus for transmitting process data in industrial environments with 7.5 Mio. installed nodes (status: Q1/2005). Due to its transmission

procedure and ring topology, INTERBUS offers features such as fast, cyclic, and time-equidistant process data transmission, diagnostics to minimize downtime and easy operation and installation. In addition, it meets the optimum requirements for fiber optic technology. The INTERBUS master/slave system enables the connection of up to 512 devices, across 16 network levels. Unlike in other systems where data is assigned by entering a bus address using DIP or rotary switches on each individual device, in the INTERBUS system data is automatically assigned to devices using their physical location in the system. In the INTERBUS system, data is transmitted according to the summation frame method. INTERBUS has a ring structure. The ring structure allows INTERBUS to send and receive data simultaneously. The master controls all devices in an INTERBUS system. The protocol architecture of INTERBUS provides the cyclic process data channel (PD) and an acyclic parameter data channel, using the services of the Peripheral Message Specification (PMS). It is also possible to exchange safety-related data in INTERBUS cycles in addition to standard data. This is done by a special safety protocol layer on top of the PD-channel. INTERBUS uses the bus-independent xml-based device description FDCML (Fieldbus Description Configuration Markup Language), according to ISO 15745-3 [1]. FDCML enables the different views of a field device to be described due to the generic device model used. Some examples include identification, connectivity, device info functions, diagnostic information, and mechanical description of a device. In configuration software, this electronic device description is used for configuration, startup, and other engineering aspects. More details about the INTERBUS system can be found on the home page of the INTERBUS Club [3].

4.1.2 INTERBUS PROFINET Proxy

Figure 4 shows the internal structure of the proxy device that realizes the mapping between PROFINET IO and INTERBUS. An INTERBUS master offers the functionality of the connected slaves through the following logical interfaces:

- Network Management: This interface is used to manage the INTERBUS-configuration. There are functions to load and activate the expected configuration, to activate or deactivate devices, etc. These functions are available by a special communication protocol called mailbox protocol. This protocol runs over a defined interface called mailbox interface (MXI).

- Process Data Channel: Via this interface the cyclic IO data values of all INTERBUS slaves can be accessed. There are mechanisms for asynchrony or synchronized data exchange.

- Parameter Data Channel: The parameter data channel can be used to access the functionalityof the acyclic parameter data transmission between the INTERBUS master and the INTERBUS slaves. This functionality is also available through the MXI.

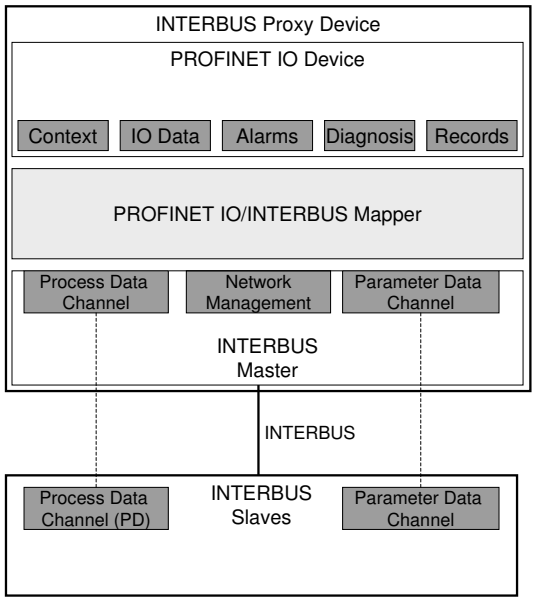


Figure 4. Architecture of the proxy device

The services of a PROFINET IO-device are offered by the following application service elements (ASE), as defined in ISO/IEC 9545. This ASEs are relevant for the use of an underlying INTERBUS-System.

- Context: services for handling the configuration of a modular slave as expected configuration, connected configuration, identification differences, etc.

- IO Data: services for exchanging cyclic IO data and status information of IO data.

- Alarm: services for sending and receiving acyclic alarms.

- Diagnosis: services for reading and writing device diagnosis.

- Record data: services for reading and writing device-dependent parameter data.

The task of the PROFINET IO/INTERBUS mapper is to connect these interfaces in several use cases such as cyclic data transfer, startup and bus error.

4.1.3 INTERBUS PROFINET Mapping

The mapping of INTERBUS to PROFINET IO is based on the well-defined slot/sub-slot model (see Figure 5). Slot 1 of the PROFINET IO device model is reserved for the INTERBUS master. The following registers are mapped to input data items:

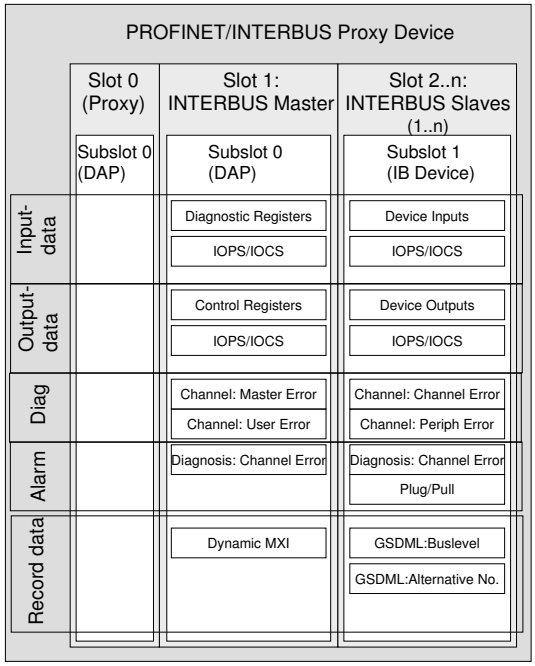


Figure 5. Mapping of INTERBUS to PROFINET IO

- Diagnostic status register (unsigned 16): This register contains the current status of the INTERBUS system. There are defined bits for bus errors, peripheral faults, etc.
- Diagnostic parameter register (unsigned 16): This register contains more information about special error conditions. In one example, it contains the slot number of the faulty module in the event of a bus error.

The output data of slot 1 is used to define the standard function registers of INTERBUS. The following registers are mapped to output data items:

- Standard function start register (unsigned 16): This register contains single bits used to trigger certain INTERBUS functions such as the acknowledgment of bus errors or the activation/deactivation of single devices.
- Standard function parameter register (unsigned 16): This register contains more information about certain actions required by standard functions as explained above. For example, it contains the slot number of the module that has to be activated or deactivated. All events which do not affect a specific slot, such as general parameterization errors, are mapped to the channel diagnostics of slot 1. The control system is informed of each diagnostic event by a corresponding diagnostic alarm.

All INTERBUS slaves connected to the master use slots $2..N$. The order of the slots corresponds to the structure of the INTERBUS system. Empty slots are not permitted. The cyclic process data of a slot is assigned directly to the input and output data of the individual INTERBUS slaves, whereas the PROFINET IO parameters are used for setting device-specific data (e.g., ID code or process data length) and for mapping device-specific PMS parameters.

A key feature of the INTERBUS system is its excellent diagnostic properties. All diagnostic messages are therefore available as PROFINET IO channel diagnostics and sent to the control system using diagnostic alarms. The control system then displays the diagnostic event in plain text. PROFINET IO devices are described in GSDML (Generic Station Description Markup Language), which is a further development of GSD (Generic Station Description) used in the PROFIBUS system. The file must include the slots of an IO device as well as all the modules, which can be plugged in the slots. Since the slaves of the INTERBUS system are mapped to the modules of the IO device during fieldbus integration, the GSDML file for the proxy should also include all INTERBUS devices available on the market. This is neither practical nor useful. Instead, the file contains universal modules with INTERBUS parameters that can be set by the user. In addition, modules from the FDCML files for INTERBUS slaves can be inserted in the GSDML file for the proxy. FDCML (Field Device Configuration Markup Language) used for the INTERBUS system is also based on XML technology, is multilingual, and has comprehensive description options for INTERBUS devices. The GSDML file for the proxy enables the user to set up the INTERBUS system in any programming software using the PROFINET IO configurator, by simply inserting the modules into slots $2..N$ according to the bus configuration. The INTERBUS system is parameterized during control system startup. Separate configuration in an additional software tool is not required, which makes startup much easier.

4.2 PROFIBUS PROFINET Proxy

In this section we describe the implementation of a PROFIBUS proxy.

4.2.1 Brief introduction to PROFIBUS

PROFIBUS is an international standardized fieldbus system according to IEC-61158 (type 3) [5]. It has been designed as a fast IO bus for transmitting process data in industrial environments with 10 Mio. installed nodes (status: Q1/2004). PROFIBUS supports fast cyclic data exchange between Master and slave as well as a parametrization channel for setup and configuration of the system. Diagnostics events on the DP slave (e.g. Pull/Plug of a module or cable break) can lead to alarms that are transferred acyclically to the DP master. For reasons of scalability different characteristics of DP slaves with optimized functionalities are exist (see Figure 6).

For reasons of configuration and diagnostics the capa-

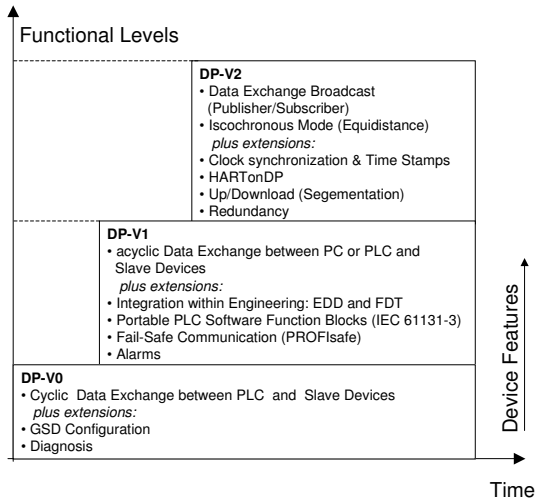


Figure 6. Characteristics of DP slaves

bilities of a DP slave are described using a standardized description file (GSD). The GSD contains the configuration data of a DP slave based on the generic station model of PROFIBUS DP. More details about the PROFIBUS system can be found on the home page of the PROFIBUS User Organization [10].

4.2.2 PROFIBUS PROFINET Mapping

The PROFIBUS PROFINET Proxy enables an IO-controller or IO-supervisor to have a transparent view of the DP slaves of the lower-level PROFIBUS DP line. This is made possible by the Proxy (hereafter also referred to as the physical device) mapping each DP slave of its DP master system as an independent IO-device instance in PROFINET IO. For diagnosis and parameterization purposes the Proxy itself is mapped additionally as an IO-device instance in PROFINET IO.

The following address information is necessary for an IO-controller or IO-supervisor for the unique addressing of the individual IO-device instances (proxies) of a physical device:

- Device name and IP address of the Proxy
- IO-Device interface UUID. The access of an IO-controller or IO-supervisor to an IO-device instance of a physical device takes place through the IO-device interface specified in [6] with a specific RPC interface UUID defined for PROFINET IO.
- IO-Device instance Object UUID. The individual IO-device instances are uniquely identified through the object UUID defined in RPC. The element *Local index* of the PROFINET Object UUID is used to represent the PROFIBUS Mac address of the underlying PROFIBUS DP slave. The Device ID represents the PROFIBUS ident number of the underlying PROFIBUS slave. Thus the Object UUID is generic and an easy access through the PROFINET IO interface functions (e.g. ReadRecord) is possible. The user only has to know the characteristics of the underlying PROFIBUS devices.

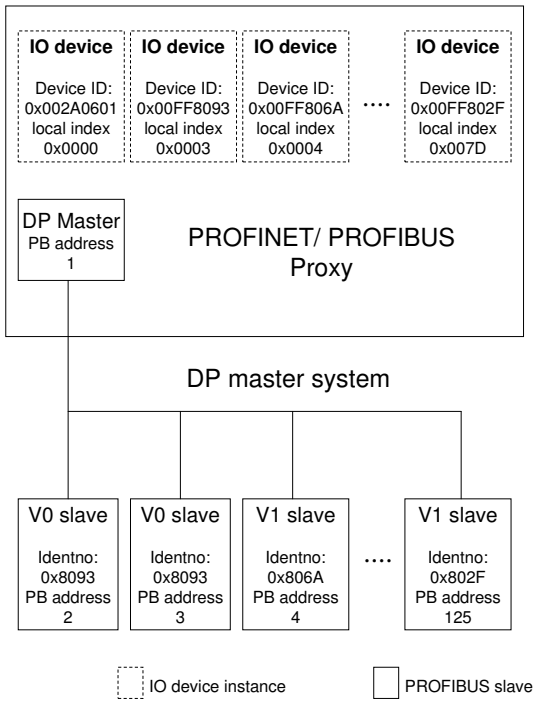


Figure 7. Mapping of PROFIBUS to PROFINET IO

Each module of the DP slave is mapped as a module of the IO-device instance, comprising of a submodule on subslot 1 (default submodule) as the carrier of the entire working data, alarms and data records. The module's slot in the IO-device instance corresponds to the module's slot of the mapped DP slave. The slot numbers of the DP master system are issued DPV1-compatible and therefore correspond to the modeling of the DP slave vendor (see Figure 8).

Startup of the DP slave
During startup the different startup-states from PROFIBUS and PROFINET have to be mapped in a correct manner. A simple 1:1 mapping is not sufficient. The proxy has to assure that the application on the IO-Controller gets the correct startup information from the underlying DP slave. Otherwise a correct and transparent diagnosis will not be possible.

Cyclic Data exchange
Figure 9 shows the data flow of the cyclic data between IO-Controller and DP slave.

The cyclic data exchange on the PROFIBUS DP is transferred into a cyclic data exchange on PROFINET IO. The PROFINET IO quality code IOPS and IOCS is either added from the DP status information or used to transfer

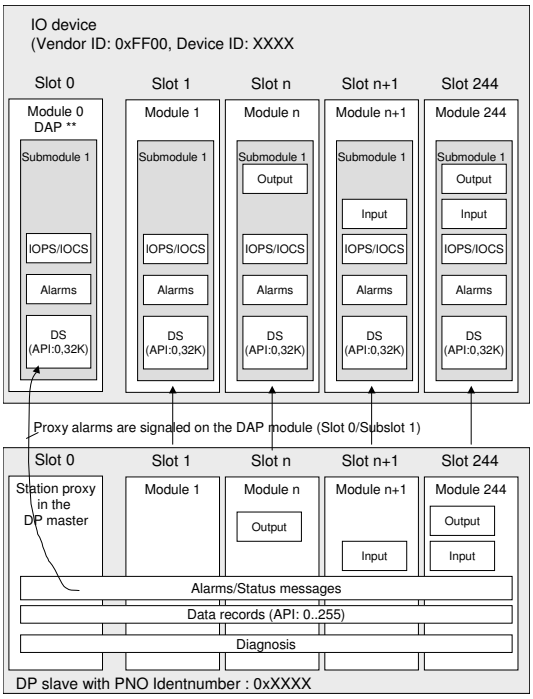


Figure 8. Station configuration of DP slave proxies

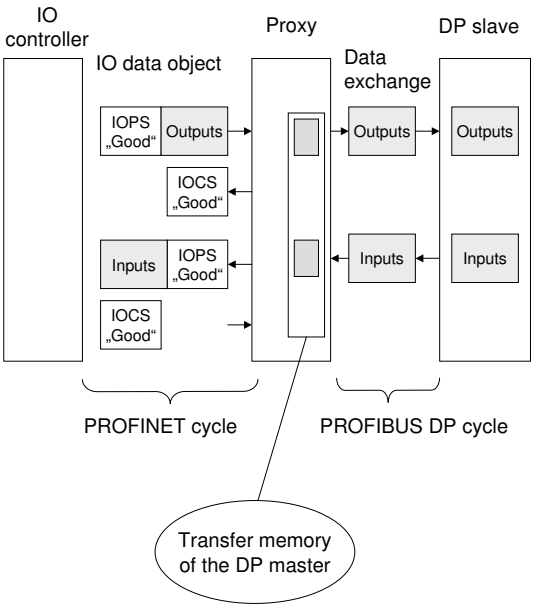


Figure 9. Transmission of I/O data from IO-controller ↔ DP slave

the correct data to the DP slaves. Details are described in the next chapters.

IOxS of the outputs
When IOPS = Good, the data received from the IO-controller are adopted in the transfer memory of the DP master. When IOPS = Bad, the last value is held or a configured substitute value is written in the transfer memory of the DP master, depending on the substitute value strategy. With the IOCS of the outputs the proxy acknowledges the successful transmission of the output data by the IO-controller into the transfer memory of the DP master. Transmission of data on the PROFIBUS is still not assured, however. In addition the IOCS of the outputs shows the current availability of the module in the view of the DP master. The initial status of a module from the DP master view is non-available (corresponds to IOPS = Bad).

The following events change the status of a module in the DP master from non-available to available and thus lead to an IOCS = Good:

- Plug alarm received on PROFIBUS DP
- PROFIBUS DP slave enters data transfer phase (at least one DataExchange telegram was transmitted successfully to the DP slave) and the module exists.

The following events change the status of a module in the DP master from available to non-available and thus lead to an IOCS = Bad:

- Pull alarm received from PROFIBUS DP
- Station failure detected on PROFIBUS DP
- IO-controller failure detected

In the operating status *Clear* of the DP master, the IOCS of the outputs can be *Good* (if module available). The current output data of the IO-controller are adopted in the transfer memory of the DP master. However, no transmission of the DP master's transfer memory takes place on the PROFIBUS.

IOxS of the inputs
The IOPS of the inputs shows the availability of the module in the view of the DP master. With IOPS = Good, the last received input data of the DP slave (current content of the transfer memory of the DP master) are forwarded to the IO-controller in addition. The initial status of a module from the DP master view is not available" (corresponds to IOPS = "Bad")

The following events change the status of a module in the DP master from *non available* to *available* and thus lead to an IOPS = *Good*:

- Plug alarm received on PROFIBUS DP
- PROFIBUS DP station enters data transfer phase (at least one DataExchange telegram was transmitted successfully to the DP slave) and module exists

The following events change the status of a module in the DP master from *available* to *non-available* and thus lead to an IOPS = *Bad*:

- Pull alarm received on PROFIBUS DP

- Station failure detected on PROFIBUS DP
- IO-controller failure detected

The IOCS of the inputs is ignored.

Alarms and Diagnosis
The following figure shows the transformation of the PROFIBUS DP V1 alarm request block in a PROFINET IO alarm request block. The AdditionalAlarmInfo is

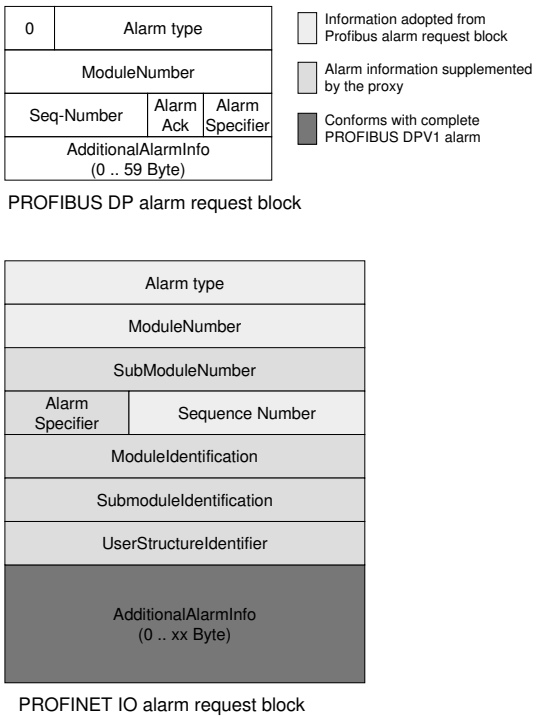


Figure 10. Implementation of the PROFIBUS DPV1 alarms

transferred transparently by the PROFINET IO alarm request block. Using the standardized UserStructureIdentifier PROFIBUS_DPV1_ALARM (see [6]), it is very easy to decode the alarm information using the already available function blocks or user programs, e.g. a PLC. Migration can thus very easyily be achieved without further efforts.

By using the Type mode of the DP V1 slave, it can be arranged that no alarm will be lost and additionally that there is no take over of different alarms from a DP slave. All alarms are processed in the order in which they are sent.

The link creates proxy diagnosis alarms on the DAP (slot 0 / subslot 1) for *coming / going* diagnosis messages of a V0 slave. The additional alarm information contains the complete PROFIBUS DP diagnosis telegram and is marked with the UserStructureIdentifier PROFIBUS_DP_SLAVE_DIAG (see [6]). Thus, not only DP/V1 slaves are mappable through the proxy but also DP/V0 slaves, which are still the majority of the devices currently sold and used in the field level of the factory.

5 Conclusions

For many applications realtime Ethernet provides an alternative to fieldbus systems. For reasons of investment protection, a migration path from the first generation of fieldbus systems towards realtime Ethernet is necessary. Using the INTERBUS and PROFIBUS fieldbus systems as an example, we have shown two different integration approaches for the realtime Ethernet standard PROFINET. With the approaches described here, it is possible to configure and operate the fieldbus systems without further fieldbus-specific engineering tools. Although there are differences between the two fieldbus systems, the modeling method used enables all the properties of both fieldbus systems to be mapped to a single PROFINET network. PROFINET is thus a suitable integration platform for heterogeneous industrial communication systems. With the introduction of realtime Ethernet, users will not be faced with an 'and/or' decision, rather they can determine to what extent they wish to use fieldbuses and realtime Ethernet according to their specific requirements. The next step is to integrate the measures required for functional safety as well as further fieldbus systems.

References

[1] FDCML. *Device description language*. `www.fdmcl.org`, 2005.

[2] J. Feld. PROFINET- Scalable Factory Communication for all applications. In *5th IEEE International Workshop on Factory Communication Systems (WFCS'2004)* , Vienna, Sept. 2004.

[3] IB Club. *INTERBUS Club*. Blomberg, 2005. `www.interbusclub.com`.

[4] IEC, Geneva. *Digital data communication for measurement and control – Fieldbus for use in industrial control systems*, 2001. IEC 61158 Ed.3 CDV, Type 8 (INTERBUS).

[5] IEC, Geneva. *Digital data communication for measurement and control – Fieldbus for use in industrial control systems*, 2001. IEC 61158 Ed.3 CDV, Type 3 (PROFIBUS-DP).

[6] IEC, Geneva. *PROFINET-IO*, 2004. IEC 61158/Ed. 4-359.

[7] IEEE, New York. *Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, Frame Extension for Virtual Bridged Local Area Networks (VLAN) Tagging on 802.3 Networks*, November 1998. IEEE Std. 802.3ac.

[8] IEEE. *Media Access Control (MAC) Bridges*. IEEE, New York, 1998. ANSI/IEEE Std 802.1D-1998.

[9] Phoenix Contact. *Industrial Ethernet Products*. `www.ethernetrail.com`, 2005.

[10] PNO. *Profibus User Organisation*. Karlsruhe, 2005. `www.profibus.com`.

[11] Siemens. *Industrial Ethernet Products*. `www.siemens.com/profinet.`, 2005.