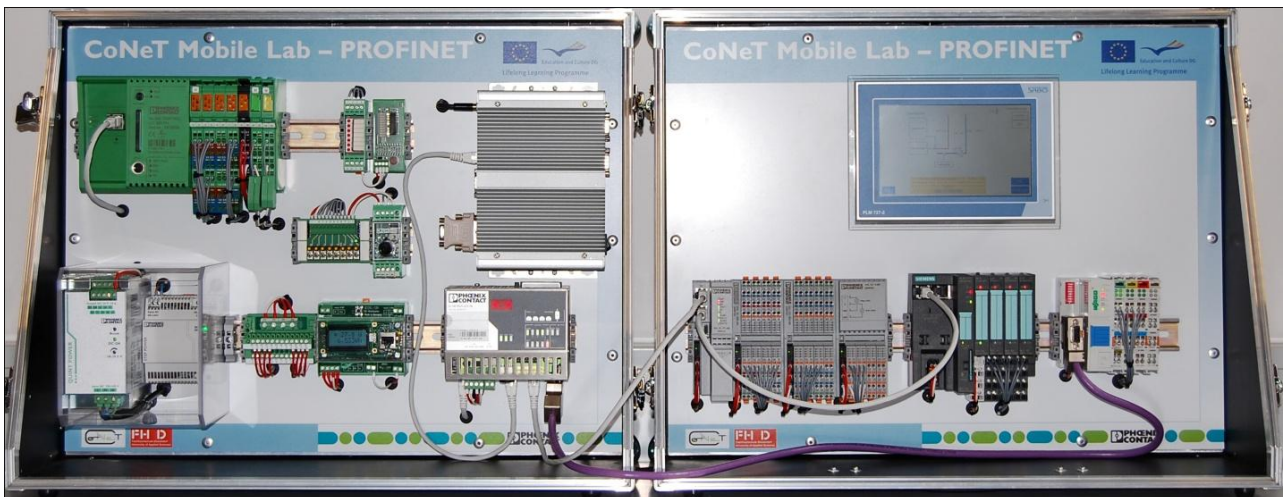


# CoNeT Mobile Lab 3

## PROFINET ON PHOENIX CONTACT PLATFORM



- PROFINET engineering -

Revision 1.0

Co-operative Network Training



University of Applied Sciences Duesseldorf  
Process Informatics Laboratory (Pi-LAB)  
<http://www.pi-lab.de>  
Contact: telelabor@fh-duesseldorf.de



## Contents

<b>PROFINET IO OVERVIEW .....</b>	<b>5</b>
Device Roles.....	5
Services of PROFINET IO .....	5
Device Model of an IO Device .....	6
Communication Services .....	7
Questions.....	7
<b>PROFINET IO COMMUNICATION.....</b>	<b>9</b>
Non-Real Time .....	9
Real Time.....	10
Isochronous Real Time .....	10
Questions.....	11
<b>PROFINET IO GSDML .....</b>	<b>13</b>
Naming GSD Files .....	13
Unique Device Identification.....	13
Structure of a GSD file .....	13
Profile Header .....	14
Profile Body .....	14
Questions.....	15
<b>PROFINET IO ENGINEERING.....</b>	<b>16</b>
Engineering Steps.....	16
Address Assignment.....	16
Device Replacement.....	17
Diagnostics and Alarms .....	17
Questions.....	19
<b>PROFINET CBA OVERVIEW .....</b>	<b>20</b>
Technological Modules .....	20

**Component Model..... 21**

**Automation Structure..... 21**

**Questions..... 22**

**PROFINET CBA ENGINEERING ..... 23**

**Creation of Components..... 23**

**Establishment of Connections..... 24**

**Downloading to PROFINET Devices ..... 24**

**Questions..... 25**

**PROFINET CBA RUNTIME MODEL ..... 26**

**TCP/IP Stack..... 26**

**Remote Procedure Call (RPC)..... 26**

**Distributed Component Object Model (DCOM)..... 27**

**Auto Marshaler (AM)..... 27**

**Active Control Connection Object (ACCO) ..... 27**

**RT-Auto (Runtime Automation Object) ..... 27**

**Questions..... 28**

**PROFINET CBA PCD..... 29**

**PROFINET CBA Integration with PROFIBUS..... 29**

    Connection using proxies ..... 29

    Connection using Integration of Fieldbus Applications ..... 30

**Understanding Questions ..... 31**

**PROFINET SUMMARY ..... 32**

## PROFINET IO Overview

In the previous section, the basics and advantages of PROFINET were discussed. In this section, you will take a deeper look at the details of PROFINET. Let's begin with the PROFINET IO.

PROFINET IO is the communication concept for distributed IO devices. Unlike the Master/Slave communication concept in PROFIBUS, the communication concept in PROFINET IO uses the so-called **Consumer/Provider model**. The device receiving data is called the 'consumer' and the device sending data is called the 'provider'. There are many devices with different features used in PROFINET IO. Similar and simple devices are grouped together and connected to distributed IO devices.

### Device Roles

- **IO Controller:** An IO Controller is usually a **PLC (Programmable Logic Controller)** on which the automation program runs. This controller is used to address the connected devices, meaning that this IO Controller exchanges the input and output signals with assigned field devices. In comparison with PROFIBUS, this would be a *Master class 1*.
- **IO Supervisor:** An IO Supervisor is usually an **engineering station**. This could be a programming device, PC or HMI device for commissioning or diagnostics. In comparison with PROFIBUS, this would be a *Master class 2*.
- **IO Device:** An IO Device is a **distributed field device** assigned to one of the IO controllers. In comparison with PROFIBUS, this would be a *Slave*.

A subsystem contains at least one IO Controller and one or more IO Devices. The connection between these devices occurs **cyclically** by IO data transfer and **acyclically** by alarms and control data transfer.

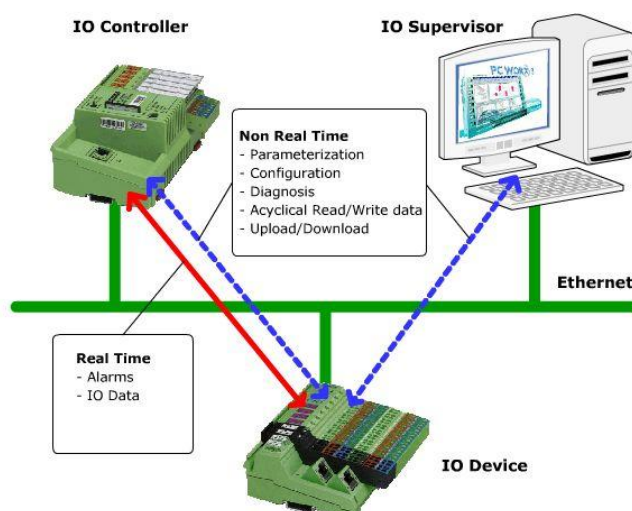


Fig. 1 Device roles in PROFINET

### Services of PROFINET IO

- **I/O Data Objects:** I/O data objects are transported **cyclically** between a provider (IO De-

vice) and a consumer (IO Controller). This transmission occurs without acknowledgment. The interval is configured using engineering tool.

- **Record Data Objects:** Record data objects are used to set the parameters on IO Devices, configure them and read their status information. The record data objects use **acyclic** data exchange by means of **read/write services**. Examples for data records are diagnostic information, parameterization, I&M (Identification and Maintenance), I/O data objects, etc.
- **Alarm Data Objects:** Alarms are used to transfer events that have to be acknowledged by the application process. There are two kinds of events, **system-defined and user-defined**. System-defined events are, e.g., insertion and removal of modules. User-defined events are, e.g., defective load voltage or temperature too high. Diagnostic and process alarms can be prioritized differently by the user.

## Device Model of an IO Device

An IO Device has a modular structure (Fig. 2). It consists of slots. You can insert modules/submodules into these slots. The modules/submodules have channels that are used to read and send output process signals. The structure of the inputs and outputs is mapped to these channels and set by the device manufacturer. Modules and submodules can be inserted or removed during runtime, if necessary.

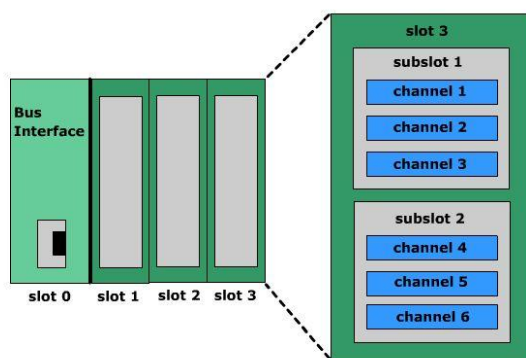


Fig. 2 Device model of an IO device

- **Slot:** A slot is a physical place of an IO Device into which you can insert a module. There are several subslots in different slots. These subslots contain the data for a cyclic data exchange. The manufacturer determines the capabilities of a module and defines the technical data in the GSD file.
- **Subslot:** A subslot is used for addressing. A slot must contain at least one subslot that contains channels. Therefore, we can group the channels on the same subslot. The definition is contained in the GSD file from the manufacturer.
- **Index:** The index specifies an acyclic read/write service for a specified subslot. The manufacturer defines the access possibilities for a slot and its subslots. Access can be made from the distributed IO Controllers, whereas the write cycle or alarm can be mapped to only one IO Controller. The read access, on the other hand, can be accessed by multiple IO Controllers.
- For the simple integration of application profiles, PROFINET IO provides special addressing elements, the so-called **API (Application Programming Interface)**. It is therefore possible to use different APIs to support multiple application profiles in one device.

## Communication Services

In order to enable cyclic or acyclic data exchange between an IO Controller/Supervisor and an IO Device, an IO Controller must set up the necessary communications paths on the system startup. Fig. 3 shows the communication services between a PROFINET IO controller and an IO device. To do this, the IO Controller sets up the connection based on data from the engineering tool. The **Application Relation (AR)** contains all the data necessary for the establishment of the data exchange. Within an AR, there are one or more APIs that allow fine-tuning of the application group. An AR can contain one or more **Communication Relations (CR)**.

Following are the CRs that are available for every API:

- One or more IO Communication Relations, for cyclic data exchange.
- Alarm Communication Relation, for events.
- Record Data Communication Relation, for acyclic data exchange.

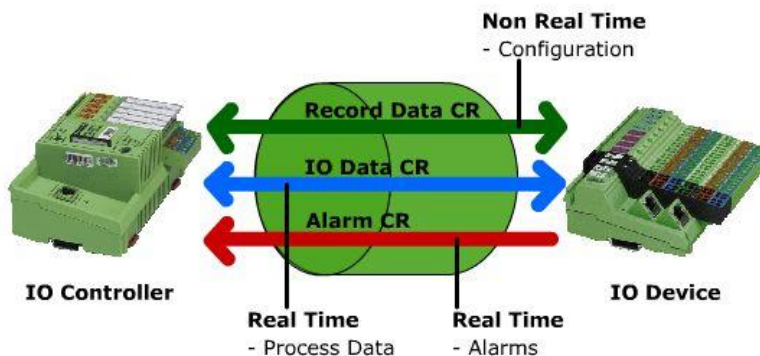


Fig. 3 Communication services

## Questions

Which of the following device roles is not included in PROFINET?

1. IO Controller
2. IO Monitoring, there is no device role named IO Monitoring.
3. IO Device
4. IO Supervisor

The record data objects use acyclic data exchange.

1. True
2. False.

Which of the following statements is true?

1. A slot contains only one subslot.
2. A subslot contains only one channel.
3. A channel contains only one slot.
4. None of them is true.

If you were a device manufacturer, where would you define the technical data of a module?

1. GSC file

2. PSD file
3. GSD file
4. PSC file

An Application Relation can contain one or more Communication Relations.

1. True
2. False



## PROFINET IO Communication

In the previous section, we discussed the communication concept used in PROFINET, Non-Real Time, Real Time and Isochronous Real Time. Furthermore, communication in PROFINET is based on the **switched Ethernet** mechanism (Fig. 4). Switched Ethernet means that every PROFINET device is connected directly to every other device. Another name for this is **point-to-point communication**. This can be actualized by the use of **switches**. A switch can be a standalone device or integrated into a PROFINET device. A switch can have several ports into which we can connect devices.

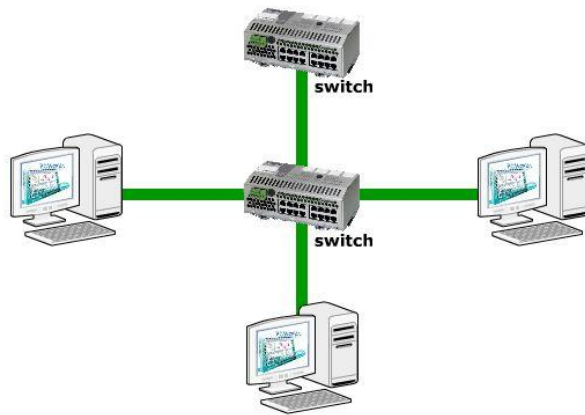


Fig. 4 Switched connections

One device is linked into a port of a switch and other devices, such as other PROFINET devices or even other switches, can be linked into other ports of a switch, too. In this way, we can combine almost any number of PROFINET devices into a large network.

Processing time for data transmission can be divided into 5 parts (Fig. 5). First, the provider provides the data (T1) and passes it to the communication stack (T2). The data is transferred to the consumer (T3). The consumer receives the data through the communication stack (T4) and finally it goes to data processing (T5).

The time used in these 5 steps will be summed up and considered whether it is in the specific interval or not. Then we can call it Non-Real Time, Real Time or Isochronous Real Time.

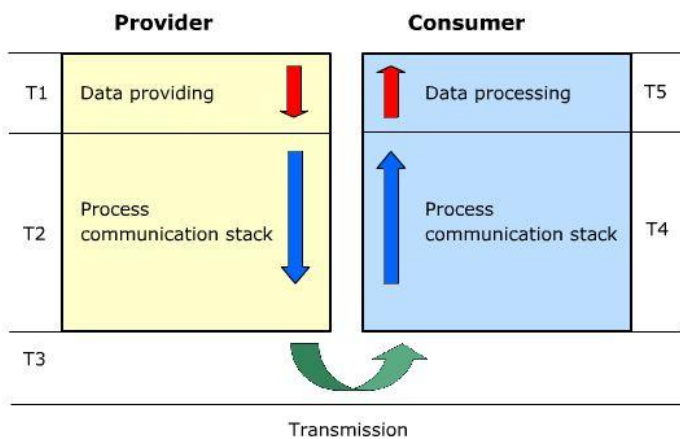


Fig. 5 Parts of the data transmission

## Non-Real Time

In PROFINET, for the **acyclical, non-time-critical processes**, such as parameterization, configuration, or HMI, we use communications with the standard Ethernet mechanisms over

TCP/IP or UDP/IP according to the international standard IEEE 802.3. This kind of communication in PROFINET is also called NRT (**Non-Real Time**) communication.

Address assignment is done by using both IP and MAC addresses. IP addresses are used for communication between two or more networks. The MAC addresses are used to identify the devices within a network. Please note that all of the PROFINET devices must support data communication over UDP/IP.

Typical cycle times are 100 milliseconds.

## Real Time

For the transmission of **cyclical, time-critical process data**, RT (**Real Time**) communication is used (Fig. 6). In this way, the same performance class as in existing fieldbus systems is achieved. The requirements of **Factory Automation** are thus fully covered.

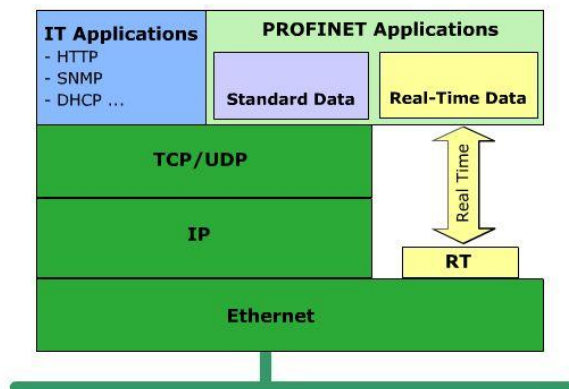


Fig. 6 RT communication in PROFINET

To enable RT, PROFINET abandons some parts of IP, TCP and UDP. The mechanisms of the Ethernet (Layer 2 of ISO/OSI reference model) are suitable for this purpose. RT communications can run parallel with NRT ones.

There are three options for RT communications:

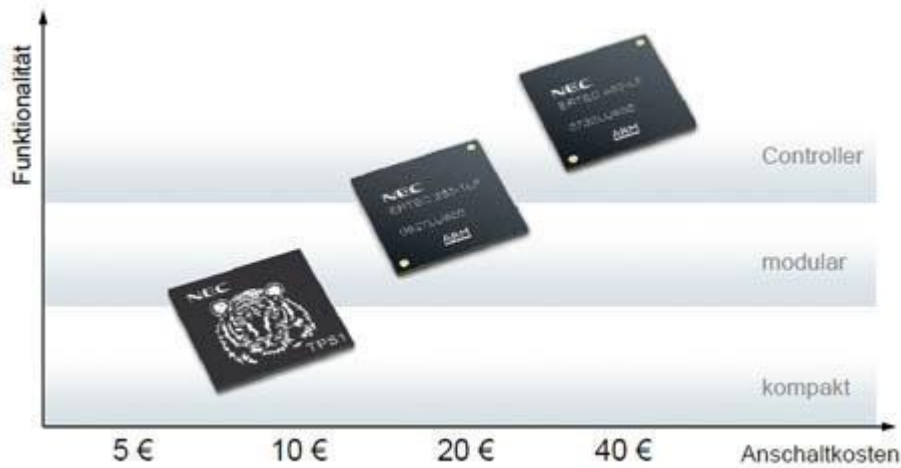
- RT communication within a network: Here, the address information is removed.
- RT communication between networks: Here, RT over UDP is available.
- Data multicasts with RT: Here, for cyclical data exchange, RT over UDP is available.

The typical cycle times are 5 - 10 milliseconds.

## Isochronous Real Time

For **clock-synchronized applications**, especially in the area of **Motion Control** applications and **high-performance factory automation** applications, the hardware-based IRT (**Isochronous Real Time**) is implemented (Fig. 7). ASICs (**A**pplication-**S**pecific **I**ntegrated **C**ircuits) with switch functionality and cycle synchronization are used to actualize this type of communication.

Phoenix Contact develops a new Tiger-Chip TPS 1. It should be available from the 3rd Q of 2011. The chip from low-cost segment (ca.10€) is fully compatible with ASICs Ertec 200, Ertec 400 and suitable for RT, IRT and as well for all possible real-time extensions



The communication cycle is split into a deterministic part and an open part. The deterministic channel carries cyclical IRT frames, while the TCP/IP and RT frames are transported on the open channel. In this way, both types of data transmission exist together without interfering with one another. Please note that if the IRT communication is necessary, the bandwidth must be divided into a deterministic part and an open part during the engineering phase.

The typical cycle time is less than 1 millisecond with jitter less than 1 microsecond.

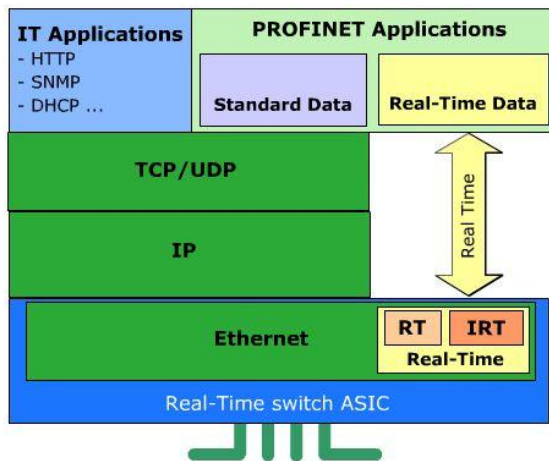


Fig. 7 IRT communication in PROFINET

## Questions

How can one achieve point-to-point communication?

1. By using hubs
2. By using TCP
3. By using switches
4. By using repeaters

If you want to configure a device, which performance class is used?

1. NRT.
2. RT.
3. IRT.

4. RT class 1.

NRT and RT communications can run parallel.

5. True
6. False.

IRT is only software-based.

1. True.
2. False
- 3.

What do you have to do if you want to use IRT communications?

1. The wires must be divided into two parts.
2. The wires must be divided into three parts.
3. The bandwidth must be divided into two parts.
4. The bandwidth must be divided into three parts.

## PROFINET IO GSDML

Every PROFINET IO field device must be described with a GSD file. A GSD (**General Station Description**) file is XML-based and contains all the technical information and functions of the device. This information is relevant for engineering and data exchange. GSD files are provided by the device manufacturer. The GSD file is multilingual and can be edited with almost all text editor programs. Since a GSD file is written in XML, we call the language describing GSD files the GSDML (**General Station Description Markup Language**).

### Naming GSD Files

There is a standard for naming a GSD file as follows:

**GSDML-[GSD schema version]-[manufacturer name]-[device family name]-[date].xml**

Where the keywords in the squared brackets are defined as follows:

- GSD schema version is the version of the used schema.
- Manufacturer’s name is the name of the device manufacturer. It can contain hyphens and spaces.
- Device family name is the name of the device family. It can contain hyphens and spaces.
- Date has the format yyymmdd. Different GSD files with the same date and device family name are not allowed.

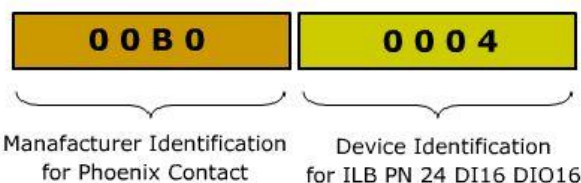
A valid GSD file name can look like this:

**GSDML-V1.0-Phoenix Contact-ILB PN 24 DI16 DIO16-2TX-V1.0-20051206.xml**

### Unique Device Identification

Each PROFINET IO field device is identified by a worldwide unique device identification number (Fig. 8). This number contains two parts, the **Vendor\_ID** and **Device\_ID**. Each part of the number is 16 bits long.

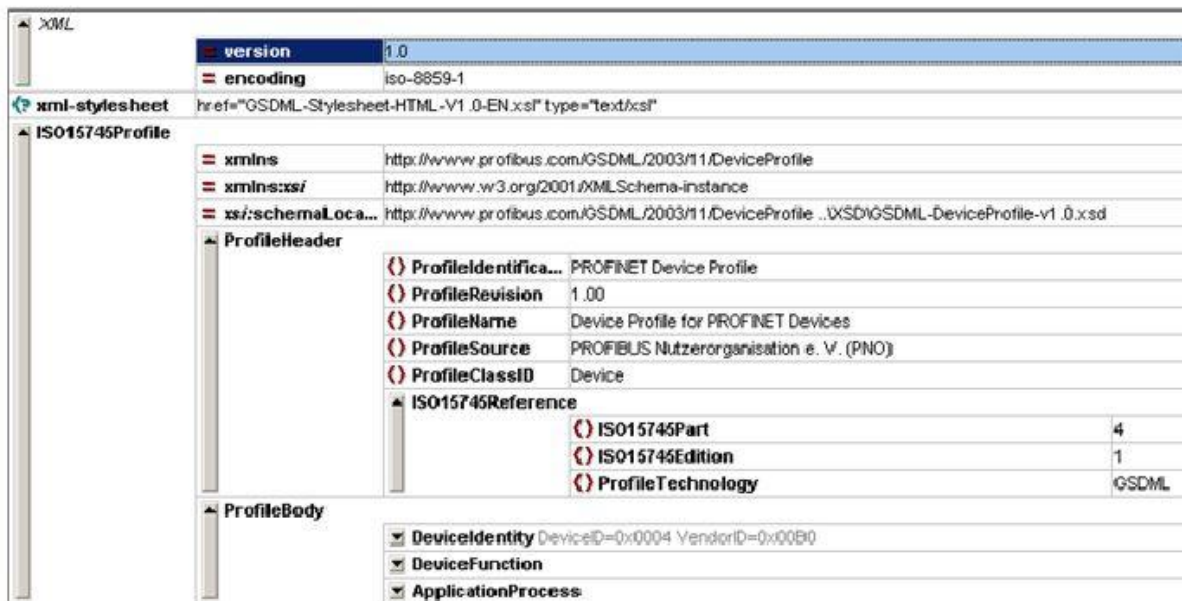
- **Vendor\_ID**: Is used to identify the manufacturer and can be obtained from the PROFIBUS Nutzerorganisation e.V. A device manufacturer needs only one Vendor\_ID.
- **Device\_ID**: Is used to identify the field device and can be assigned by the manufacturer.



**Fig. 8** Structure of the unique device identification number

### Structure of a GSD file

The GSD is based on the ISO 15745 standard and contains two parts, **profile header** and **profile body**. Fig. 9 shows a GSD structure.



version	1.0
encoding	iso-8859-1
xml:stylesheet	href="GSDML-Stylesheet-HTML-V1.0-EN.xsl" type="text/xsl"
xmlns	http://www.profinet.com/GSDML/2003/11/DeviceProfile
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.profinet.com/GSDML/2003/11/DeviceProfile ..\XSD\GSDML-DeviceProfile-v1.0.xsd
ProfileHeader	<ul style="list-style-type: none"> <li>ProfileIdentification: PROFINET Device Profile</li> <li>ProfileRevision: 1.00</li> <li>ProfileName: Device Profile for PROFINET Devices</li> <li>ProfileSource: PROFIBUS Nutzerorganisation e. V. (PNO)</li> <li>ProfileClassID: Device</li> <li>ISO15745Reference:           <ul style="list-style-type: none"> <li>ISO15745Part: 4</li> <li>ISO15745Edition: 1</li> <li>ProfileTechnology: GSDML</li> </ul> </li> </ul>
ProfileBody	<ul style="list-style-type: none"> <li>DeviceIdentity: DeviceID=0x0004 VendorID=0x00B0</li> <li>DeviceFunction</li> <li>ApplicationProcess</li> </ul>

Fig. 9 Structure of a GSD file

## Profile Header

This part contains the common definitions specifying the functions of the field device. The reference to the international standard ISO 15745 is also contained.

```

<ProfileHeader>
  <ProfileIdentification>PROFINET Device
  Profile</ProfileIdentification>
  <ProfileRevision>1.00</ProfileRevision>
  <ProfileName>Device Profile for PROFINET Devices</ProfileName>
  <ProfileSource>PROFIBUS Nutzerorganisation e. V. (PI)</ProfileSource>
  <ProfileClassID>Device</ProfileClassID>
  <ISO15745Reference>
    <ISO15745Part>4</ISO15745Part>
    <ISO15745Edition>1</ISO15745Edition>
    <ProfileTechnology>GSDML</ProfileTechnology>
  </ISO15745Reference>
</ProfileHeader>
  
```

## Profile Body

The information in this sector describes the technical possibilities of the field device. All of the modules, submodules, initial pre-allocation and diagnostics are described here. It can be further divided into:

- **DeviceIdentity** block (VendorID, DeviceID, InfoText, VendorName)
- **DeviceFunction** block (Family, MainFamily, ProductFamily)
- ApplicationProcess block
  - **DeviceAccessPointList**: contains the description of the degree of expansion of the individual interface modules.

- **ChannelDiagList**: contains the channel for errors and error text.
- **GraphicsList**: contains reference to the graphic representing the device.
- **CategoryList**: is used to build categories for the engineering tool.
- **ExternalTextList**: contains all text that can be referenced by the other segments.

## Questions

Can you use NotePad editor from Windows XP to edit a GSD file?

1. Yes
2. No.

Which choice is correct for **yyyymmdd**?

1. 2007-10-23
2. 20072310
3. 20071023
4. 23102007

A manufacturer needs more than one **Vendor\_ID**.

1. True.
2. False

How many bits does a unique identification number contain?

1. 8 bits
2. 16 bits
3. 32 bits
4. 64 bits

Which part of a GSD file describes the technical possibilities of the device?

1. ISO 15745
2. ISO/OSI reference model
3. Profile Header
4. Profile Body

## PROFINET IO Engineering

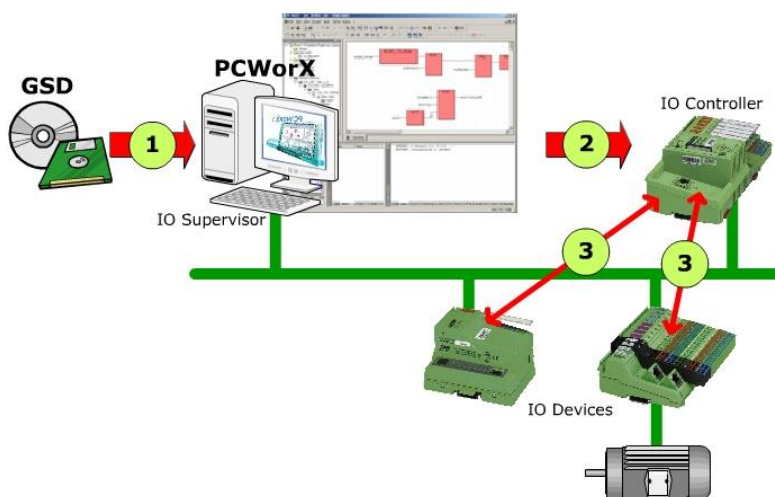
From the previous sections, you may already have an idea of what we need to do to commission a PROFINET IO system. If you are familiar with PROFIBUS DP, you may find the PROFINET engineering straightforward.

Basically, we need the following components in order to set up a system:

- An engineering tool: This is an IO Supervisor, such as PCWorX or STEP 7.
- An IO Controller: We need at least one IO Controller for a system.
- An IO Device: We need at least one IO Device.
- GSD files: They describe the properties of PROFINET devices, and contain all the information required for configuration.

### Engineering Steps

- The GSD files are imported into the engineering tool. In the engineering tool, the decentralized field devices are assigned to one or more IO Controllers. In this step, the IO Device is to be configured to the actual system network based on the content in the GSD file. The IO Device is simultaneously integrated, appropriately parameterized and configured into the PROFINET topology.
- After completion of the engineering process, the configuration information and user programs are transferred to the IO Controller.
- The IO Controller independently takes over the data exchange with the decentralized IO Device automatically.



**Fig. 10** PROFINET IO engineering steps

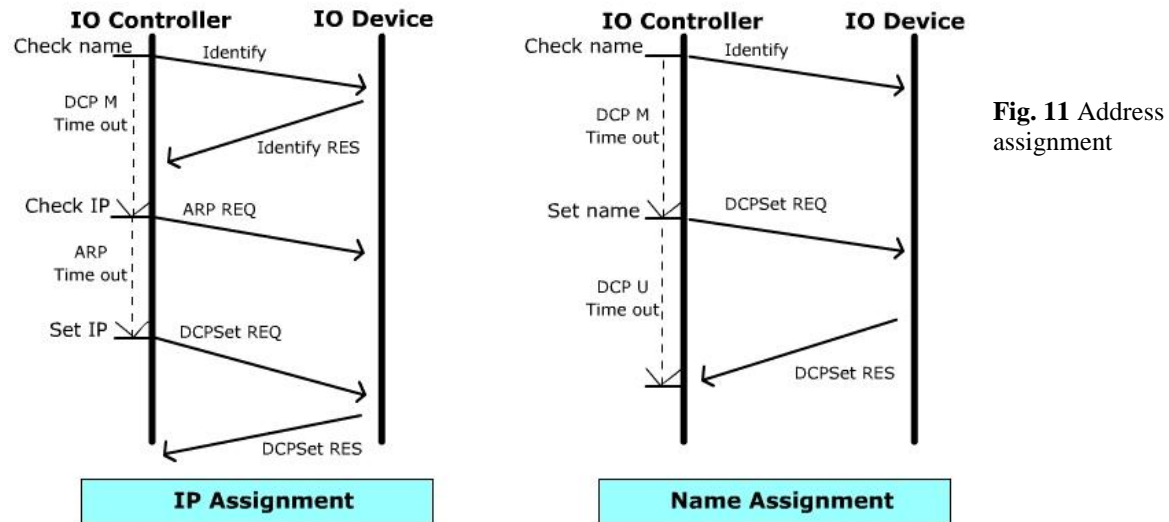
### Address Assignment

The addressing of the field devices on IP-based communication can be done by using an IP address. For the address assignment, PROFINET uses the **DCP** (Discovery and Configuration Protocol).



As delivered, every field device has, among other things, a MAC address and a symbolic name. We can use this information to assign the name of the respective field device. The address assignment can be done using the DCP protocol as follows (Fig. 11):

- Assignment of the unique name for the respective field device;
- Assignment of an IP address from an IO Controller before system startup.



**Fig. 11** Address assignment

## Device Replacement

Should a device module or a complete device be defective, it can be rapidly replaced without renewed configuration of the spare part. This is because exchangeable media such as MMC (Multi Media Card) and C-Plug enable the simple swapping of field devices and network components without the need for a programming device.

If you remove the memory card from a PROFINET device and insert it into a different PROFINET device, the device name is transferred. In the case of active network components such as switches, the full configuration data is directly on the C-Plug.

After replacement, the IO Controller automatically performs parameterization and configuration of the new device, and assigns its new IP address. The cyclical user data exchange is then restored.

## Diagnostics and Alarms

PROFINET device diagnostics take place at three levels:

- The error diagnostics in the device, such as a failure of the station.
- The error diagnostics in the module location, for example damage to the analog input module.
- The error diagnostics on the channel, for example a wire breakage.

In addition to the device, in PROFINET the **network infrastructure** can be diagnosed via the **SNMP** protocol.

In the example in Fig. 12, a wire between an IO Device and a switch has been interrupted. The switch generates an alarm in the IO Controller, which also contains information about the port affected. Additionally, diagnostic information can also be read directly from an IO

Supervisor.

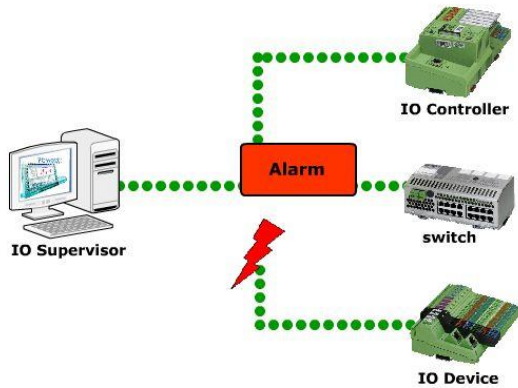


Fig. 12 Example of an alarm detection

The user has two possibilities to access the PROFINET diagnostics.

- **Diagnostics via the IO Controller:** The IO Device sends the diagnostics information to the IO Controller, which deposits this in the SZL (System Status Lists). The IO Supervisor accesses the SZL in the IO Controller. Any malfunctions can then be visualized in the HMI (Human Machine Interface). In addition, there can be a response to the error in the application program.

- **Diagnostics directly from the IO Device:** The engineering tool (IO Supervisor) can also read the status of the IO Device directly. To do this, the IO Device must be connected to the Ethernet.

PROFINET enables one further option for diagnostics. You can read directly from the status LEDs on the device whether undisturbed communication is possible and whether data can be sent or received.

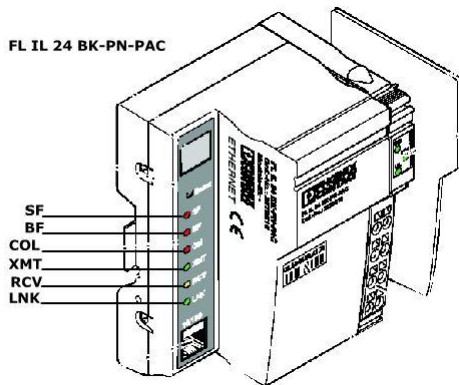


Fig. 13 FL IL 24 BK-PN-PAC

Table. 1 shows the LEDs on Inline Bus Coupler FL IL 24 BK-PN-PAC (Fig. 13) from Phoenix Contact which are provided for this purpose:

Table. 1 Diagnostics and alarm LEDs in FL IL 24 BK-PN-PAC

Description	Color	Status	Meaning
SF	Red	ON	System error present (incorrect parameterization, bus error, peripheral fault)
		OFF	No system error, INTERBUS running without errors
BF	Red	ON	No link status available
		Flashing	Link status available, no communication connection to the IO controller, connection establishment is currently active
		OFF	The IO controller has established an active communication connection to the IO device
COL	Red	ON	Collision of data telegrams

		OFF	Transmission of telegrams without a collision (if LNK LED active)
<b>XMT</b>	Green	ON	Data telegrams are being sent
		OFF	Data telegrams are not being sent
<b>RCV</b>	Yellow	ON	Data telegrams are being received
		OFF	Data telegrams are not being received
<b>LNK</b>	Green	ON	Physical network connection ready to operate
		OFF	Physical network connection interrupted or not present

## Questions

Do you need the GSD files after the IO Controller takes over data exchange with the IO Device?

1. Yes.
2. No.

Which protocol does PROFINET use for address assignment?

1. TCP.
2. UDP.
3. DCP.
4. IP.

The device diagnostics can be done at..

1. 3 levels.
2. 4 levels.
3. 5 levels
4. 6 levels.

An IO Supervisor can access diagnostic information from an IO Device directly.

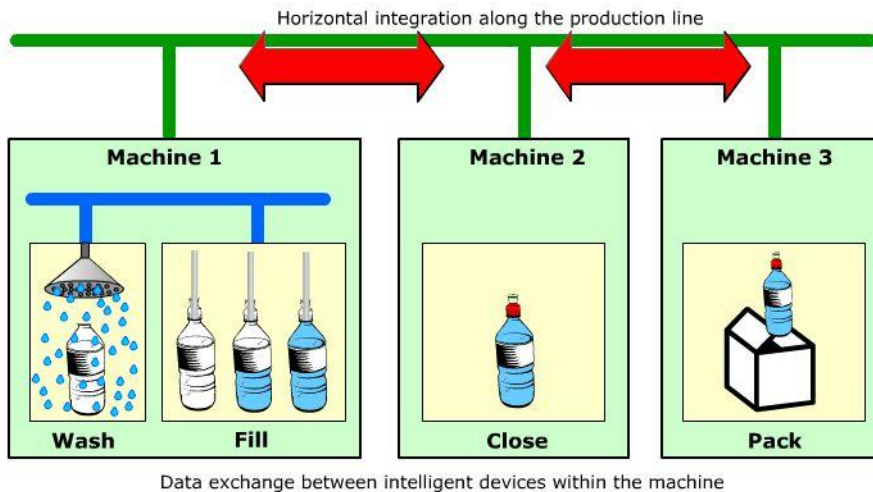
1. True.
2. False.

LEDs are used for diagnostics, too.

1. True
2. False

## PROFINET CBA Overview

While the PROFINET IO provides a modularized automation concept **within** a plant, the PROFINET CBA (Component **B**ase **A**utomation) provides a modularized automation concept **between** the plants (Fig. 14). The different subsystems or machines can be developed, tested and commissioned by one or more manufacturers with one standard. The horizontal scalability can be achieved this way.



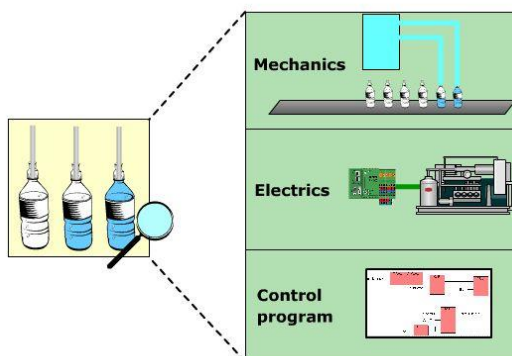
**Fig. 14** Horizontal scalability in an automation system

## Technological Modules

A technological module in PROFINET CBA consists of

- Mechanical,
- Electrical or Electronics,
- Control Logic or Software

The three components of a technological module describe the function of an automated plant or machine (Fig. 15). They operate with each other during the product manufacturing process. Thus, a technological module represents a system-specific plant or machine completely.



**Fig. 15** Three components of a technological module

## Component Model

The PROFINET CBA is based on a **component model** (Fig. 16), i.e. it consists of several small sub-units. These sub-units coordinate among themselves exchanging their information and act autonomously as **technological modules**.

Components such as machines or plants or their parts will be considered as the technological modules. The functions of these components are encapsulated in the unique COM objects, called **PROFINET components**. COM (Component Object Model) was developed by Microsoft using the object orientated concept. With COM, you can develop the components, which are based on ready-made components, easily.

The PROFINET components are also the COM objects, but their functions are specific to PROFINET. A PROFINET component can be used as a black box with uniquely defined interfaces. Therefore, you can connect the PROFINET components to each other in any combination.

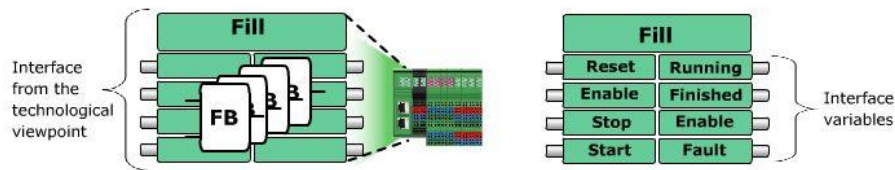


Fig. 16 Component model of PROFINET CBA

For configuration and interconnection with other PROFINET devices, only the variables that are required for interaction with each other can be accessed. The definition for the communication is defined in the COM objects, this includes;

- Communication within one process,
- Communication between two processes within one device,
- Communication between two processes on different devices.

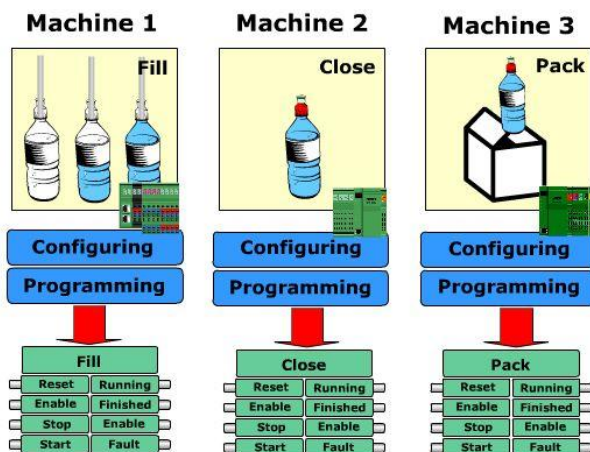


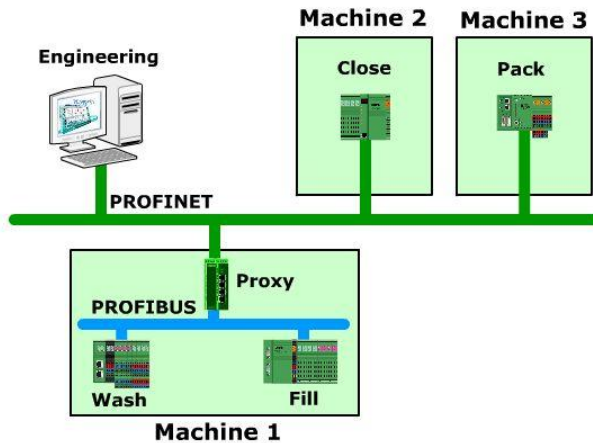
Fig. 17 Example for a distributed automation system

This kind of distributed automation system (Fig. 17) design is necessary in case you want to have modularization of your systems and machines. Moreover, the reuse of your system or machine parts is possible, which means a reduction in engineering costs.

## Automation Structure

Like PROFINET IO, PROFINET CBA components are connected using industrial Ethernet communication mechanisms. The existing PROFIBUS components can be integrated into this

system by using the suitable proxy (Fig. 18).



**Fig. 18** Automation structure with PROFINET and PROFIBUS

## Questions

What can you do with PROFINET CBA?

1. Build a modularized system within a plant
2. Build a modularized system between the plants
3. Build a decentralized system for field devices
4. Build a decentralized system for IO devices

What does a technological module represent?

1. A production machine during the production process
2. A field device during cyclical data exchange
3. A field device during acyclical data exchange
4. An IO Controller during the startup process

What is the concept of COM used for?

1. A decentralized concept
2. A centralized concept
3. An object orientated concept
4. A connection-oriented concept

A PROFINET component can be used as a black box.

1. True
2. False

We use a gateway to integrate PROFIBUS into PROFINET.

1. True
2. False

## PROFINET CBA Engineering

For the users, it is important that the engineering tool for PROFINET is easy to use and has standardized characteristics. For this purpose, a **vendor-independent engineering concept** is defined. This concept is based on the **engineering object model**. That means you can use the components from different vendors as well as vendor-specific or user-specific functional extensions with one engineering tool. The products from different vendors can be easily diagnosed and integrated into one installation using **connection editor**. With the connection editor, you can establish the connection relations between components by drawing the line between them. The connection editor will then check the connection plausibility.

In PROFINET CBA, the data exchange between nodes is ensured by the configured connections. An XML-based file describes the technical functions of a PROFINET CBA device. This file is called **PCD (PROFINET Component Description)** and is provided by the device manufacturer.

A system-wide application can be built in three phases (Fig. 19):

- Creation of components;
- Establishment of connections;
- Download of connection information into the PROFINET devices.

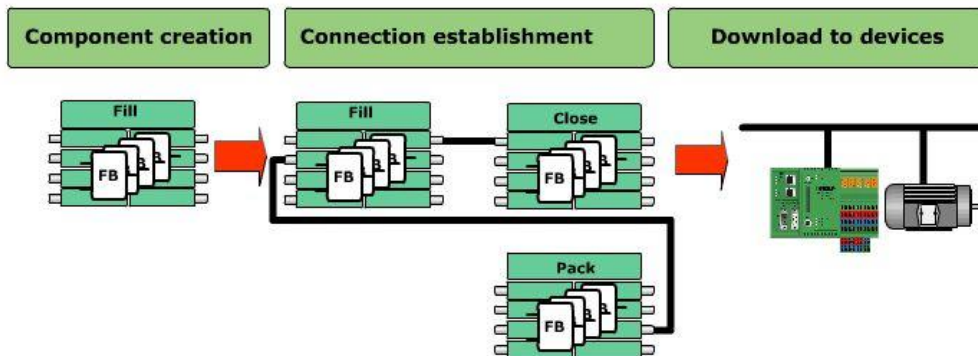
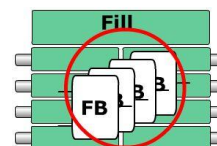


Fig. 19 Three phases of a PROFINET CBA application

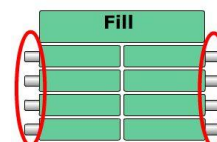
### Creation of Components

The creation of PROFINET components as an image of the technological modules can be done in three steps:

The vendor-specific tool can be used to configure and parameterize the devices. The device manufacturer can use the old software tool by simply extending a new component interface. In this way, existing programming knowledge can still be used.



The component interface will be defined. The interface consists of several variables, such as Name, Type, PROFINET direction and Comment



The PROFINET component will then be generated with encapsulated variables. The properties of the component include Component Name, Version Number and Storage Location.



After this, you will have a PCD file, which can then be imported into the library of the connection editor.

### Establishment of Connections

The **connection editor** is used to interconnect the PROFINET components, which are represented by a PCD file (Fig 20). The components can be selected from a library into the application by clicking on a mouse.

This step simplifies the interconnection process by using a simple graphics-based configuration. Complex programming and knowledge of communication functions in the device is no longer required.

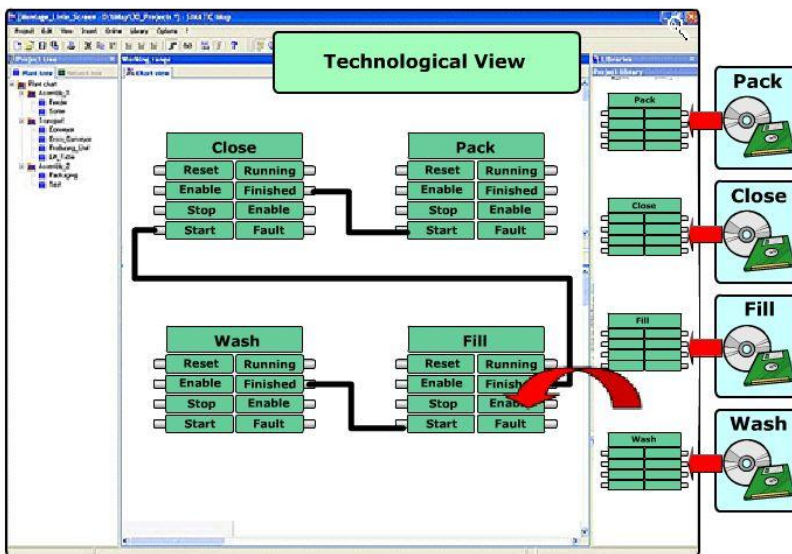


Fig. 20 Establishment of connections in the connection editor

By using connection editor, we can combine the individual distributed applications throughout the system. The different PROFINET components from different manufacturers can be interconnected together. The frequency of transfer can be specified at this point and the connection editor will check to make sure that only the same data types are interconnected.

### Downloading to PROFINET Devices

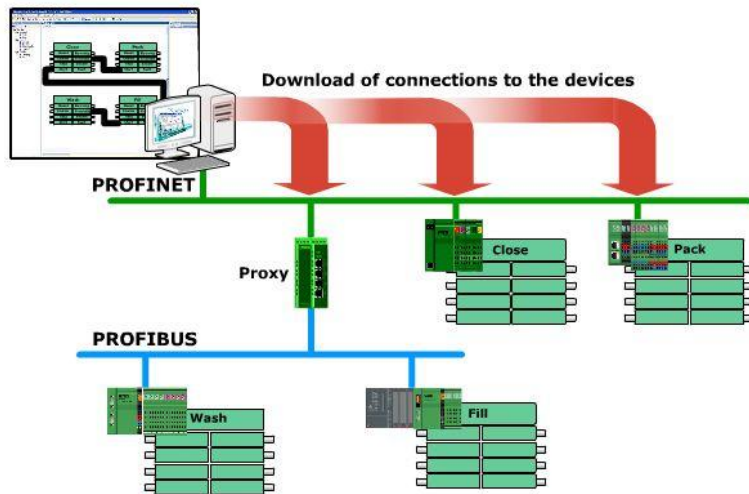
The connection information will be downloaded into the appropriate PROFINET devices according to the component connection plan (Fig. 21). That means that each component knows its own communications partner, communications relationships and the information to be exchanged.

The establishment of the communications connections to the partner and the data exchange occur automatically. The connection information is loaded into each consumer, which then independently creates and monitors the communications connections to the participating partner devices.

PROFINET CBA communications are based on the TCP/IP or UDP/IP for Non-Real Time



communications and RT for Real Time communications.



**Fig. 21** Downloading the connection information

## Questions

What is an engineering tool used for in PROFINET CBA?

1. Binding Editor
2. Communication Editor
3. Connection Editor
4. Combining Editor

What kind of format does a PCD file have?

1. HTML
2. GSDML
3. FDML
4. XML

The components can be connected through interfaces.

1. True
2. False

How can one establish the connections in PROFINET CBA?

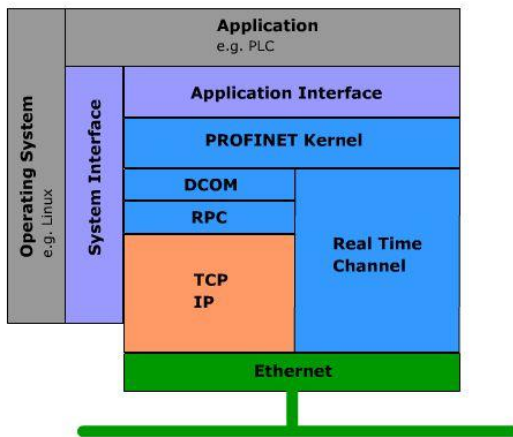
1. By using the complex programming.
2. By using the graphics-based configuration tool.
3. By using knowledge of the functions of the devices.
4. By using C programming language.

A data exchange occurs automatically after the connection information is loaded to the consumer.

1. True
2. False

## PROFINET CBA Runtime Model

In the PROFINET component model (Fig. 22), the DCOM (**D**istributed **C**omponent **O**bject **M**odel) is used and defined as a shared application protocol that is based on TCP/IP. DCOM uses the standardized RPC (**R**emote **P**rocedure **C**all) protocol to enable communication between PROFINET components over the network.



DCOM is included by default in the available software stack. If there is a **need for using** time-critical applications, the PROFINET real-time channel will be used. The configuration of data transfer can be done in the configuration tool. The transfer rate is also called QoS (Quality of Service).

Fig. 22 PROFINET CBA runtime model

Every exchanged data frame will be identified with a so-called QC (Quality Code). With this code, the user can decide how to process the data. **The codes are shown in Fig. 23.**

Table of quality codes

80	The value is good
00	The value is bad
4C	The value is an initial value
44	Last valid value is retained

Fig. 23 Table of quality codes

## TCP/IP Stack

The Transport Control Protocol (TCP) and Internet Protocol (IP) are the part of Ethernet standard. TCP/IP enables the connections between devices, manages connections and routes data packets from one endpoint to another. The TCP/IP stack is not included in the PROFINET Runtime Model. It is usually an off-the-shelf software component and is often included with the RTOS (**R**eal **T**ime **O**perating **S**ystem).

## Remote Procedure Call (RPC)

The Remote Procedure Call transports DCOM requests between PROFINET runtime software and the TCP/IP stack. RPC is a standard component of Microsoft Windows. For embedded systems running a non-windows OS, the PROFINET runtime software includes a standalone RPC. RPC uses the connection-based messaging of the TCP/IP stack and memory management, task synchronization and critical resource management services of the RTOS. To integrate RPC into a PROFINET device, both the interface to TCP/IP and the RTOS must be customized. If procedure tracing is implemented by the PROFINET device, the RPC must be customized to include the standard interfaces to the diagnostic services.

## Distributed Component Object Model (DCOM)

The Distributed Component Object Model manages the distribution of the properties, methods and events of the interfaces included in a PROFINET device. DCOM is a standard component of Microsoft Windows. For embedded systems running a non-windows OS, the PROFINET runtime software includes a standalone DCOM. The DCOM component must be integrated with four other components; the RPC, the RTOS, the DCOM Application Interface and the Diagnostic module. The standalone DCOM provided by PROFIBUS International contains the standard RPC API. Much more complex is the integration of DCOM and the services provided by the RTOS. Since DCOM calls operate asynchronously, the RTOS services use callback routines, necessitating a highly customized interface between the RTOS and DCOM.

## Auto Marshaler (AM)

The Auto Marshaler serializes the components of a parameter call prior to handing the marshaled call to DCOM. Marshalling on the Client side includes placing parameters in a message and converting the local data type to the data type of the network RPC. Marshalling on the Server side includes removing parameters from the message and converting the data type into the data type of the local Server. The Auto Marshaler performs both Client and Server operations. When the Auto Marshaler initiates a method in an interface, it functions as a Client. When the Auto Marshaler receives an access to a method, it functions as a Server. Auto Marshaling insulates the local Client or Server from the details of the call interface. Clients and Servers can call all methods as if they are local methods. Auto Marshaling is part of DCOM and is highly integrated with the DCOM provided in the PROFINET runtime software. Integration activities for this component include RTOS and Diagnostic services integration.

## Active Control Connection Object (ACCO)

The Active Control Connection Object is a software component in the PROFINET Runtime Software core (Fig. 24). The ACCO ensures the coordination of the data exchange in a PROFINET CBA device. It is integrated into every PROFINET CBA device and used to monitor the communication relations. It can be either a provider or a consumer.

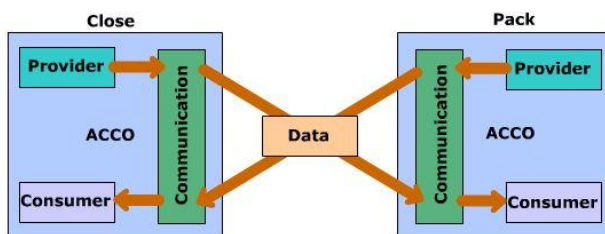
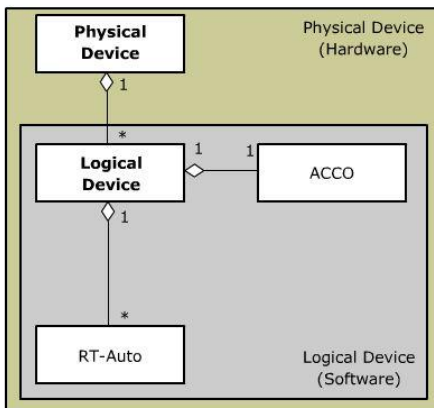


Fig. 24 Coordination of the data exchange between two ACCOs

As a provider, the ACCO makes sure that the requested data are sent automatically on time. As a consumer, the ACCO establishes and monitors the communication with the relevant provider based on the configured QoS.

## RT-Auto (Runtime Automation Object)

An RT-Auto is an executable program with a corresponding data range and contains the executable function of the field device (Fig. 25). More than one RT-Auto can be implemented in a field device. That means different functions are assigned to different RT-Autos.



**Fig. 25** Runtime Automation Object in the PROFINET CBA

## Questions

What protocol does DCOM use to enable communication?

1. RPC
2. DHCP
3. DCP
4. HTTP

What can we do by looking at Quality Code?

1. Locate the device
2. Change the cyclical time for data exchange
3. Decide how to handle the received data
4. By means of QC, you can decide how to process the data.
5. Assign the field device name

For non-windows embedded systems, RPC is integrated as standalone.

1. True
2. False

A TCP/IP stack is part of PROFINET Runtime Model.

1. True
2. False

What does ACCO do as a consumer?

1. Assign IP address for the field device
2. Assign device name for the field device
3. Look up the routing table
4. Monitor the communication with the provider

## PROFINET CBA PCD

PCD (PROFINET Component Description) is an XML file that describes the functions and objects of a PROFINET CBA component. By using XML, you can describe the data in the vendor- and platform-independent format.

We usually create a PCD file after we create the user software by using the vendor-specific tool (e.g. STEP 7 by Siemens) for the respective field device. The PCD is then stored in the engineering system and can be loaded into the engineering tool (e.g. Siemens iMap) if you want to interconnect them. Alternatively, you can obtain the vendor-independent PROFINET Component Editor from PI that can be used to create the PCD file.

A PCD file contains the following information:

- Description of the component as a library element
  - Component Identification
  - Component Name (e.g. Fill)
- Description of a hardware
  - IP address storage
  - Access to diagnostic data
  - Downloading of interconnections
- Description of the software functionality
  - Assignment between hardware and software
  - Component interface
  - Properties of variables (1..n)
    - Name (e.g. Start)
    - Data type (e.g. Boolean)
    - Direction (e.g. Input or Output)
- Storage location of the component project
  - e.g. with STEP 7

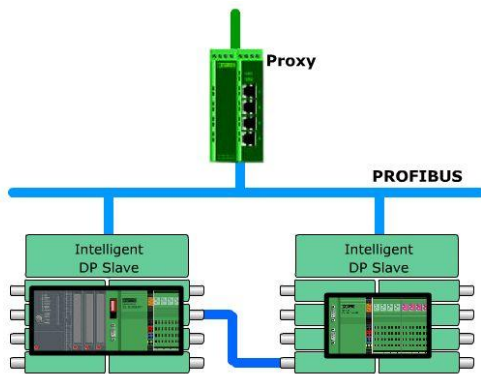
## PROFINET CBA Integration with PROFIBUS

If there is an existing PROFIBUS system, you have two possibilities to integrate it into the PROFINET system:

- By using **proxy**: You can connect the fieldbus devices through proxies.
- By integrating **fieldbus applications**: You can connect the entire fieldbus applications.

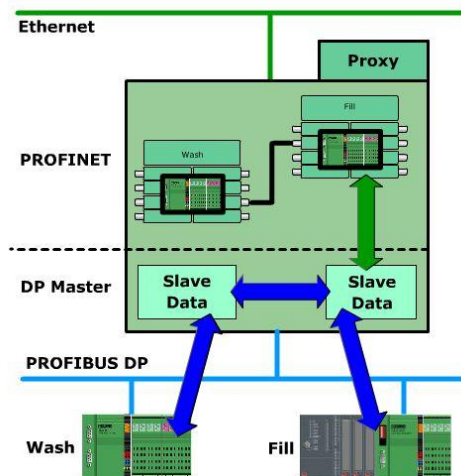
### Connection using proxies

The proxy concept in PROFINET CBA is similar to the proxy concept in PROFINET IO (Fig. 26). A proxy enables the integration of existing fieldbus systems into the PROFINET system in a simple and transparent way. The proxy maintains a transparent conversion of communications (no protocol tunneling) between the networks.



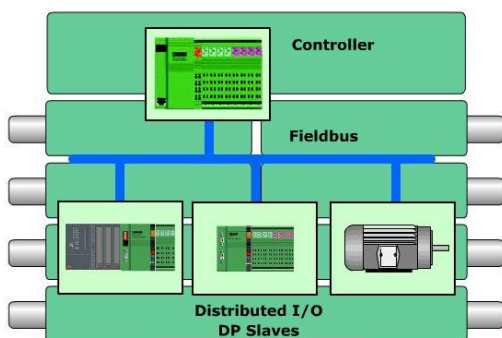
**Fig. 26** Connection of a PROFIBUS in a PROFINET CBA system by a proxy

In PROFIBUS, the proxy is a PROFIBUS master on one side, coordinating the data exchange between the PROFIBUS nodes, and on the other side, it is an Ethernet node with PROFINET communications (Fig. 27). Proxies can be implemented as e.g. PLCs, PC-based controllers, or pure gateways. In the component view, intelligent slaves are used as independent PROFINET components. In the PROFINET connection editor, such PROFIBUS components are not distinguishable from the components on the Ethernet. Proxies make communications between devices on different bus systems transparently possible.



**Fig. 27** Component structure of a PROFIBUS – PROFINET proxy

### Connection using Integration of Fieldbus Applications



**Fig. 28** Connection using integration of fieldbus application

An entire fieldbus application can be modeled as a PROFINET component in the framework of the component model (Fig. 28). This is important when an already existing system has to be extended using PROFINET. It is not important which fieldbus is used to automate the segment. For communications between the existing system and PROFINET, the fieldbus master in the PROFINET component has to be PROFINET-capable. Thus, the existing fieldbus mechanisms are used within the components, and PROFINET mechanisms outside the components.

This migration option ensures that the user (system operator or builder) can protect the investment made in existing systems and wiring. Moreover, existing know-how of user programs is preserved. This makes the seamless transition to new system segments with PROFINET possible.

## Questions

Which of the following statements is correct?

1. A PCD file is based on XML.
2. An XML format enables platform independence.
3. A PCD file describes the functions of PROFINET CBA.
4. All of them are correct

A proxy is used to connect PROFINET with PROFIBUS.

1. True
2. False

What **cannot** be implemented as a proxy?

1. Switches
2. PLCs
3. PC-based controllers
4. Gateways

What does a proxy actually do?

1. Maintain communications
2. Maintain the transparent conversion of communication between networks.
3. Maintain the transparent connections between field devices
4. Maintain the connections between IO Controllers

With the proxy concept, you can protect your existing investment.

1. True
2. False

## PROFINET Summary

There are two performance classes of PROFINET, the PROFINET IO and PROFINET CBA. PROFINET IO is designed for data exchange with decentralized field devices. PROFINET CBA is designed for distributed automation systems. The PROFINET CBA concept satisfies the current market requirements.

The PROFINET IO and its functionalities were discussed. In the case of managing process data, we can distinguish three types of device roles; IO Controller, IO Supervisor and IO Device. PROFINET IO uses the Provider/Consumer model. The connections between them can be Non-Real Time (NRT), Real Time (RT) and Isochronous Real Time (IRT) depending on the requirements for data transmission.

A GSD file is used to describe the functions and options of PROFINET IO devices. It is written in XML and therefore vendor- and platform-independent. In the engineering tool, you can easily implement these GSD files from the library.

In the section on PROFINET IO Engineering, we explained how to commission a PROFINET IO system, beginning from the implementation of GSD files, then projecting hardware onto the engineering tool and finally downloading the user program to the appropriate devices. With the Telelabor portal, you can see for yourself how to commission such a system, live on the Internet.

PROFINET CBA is designed for distributed automation systems. This increases the system scalability and reduces the cost. PROFINET CBA uses the component concept that is based on DCOM, which is developed by Microsoft. The components are based on the so-called technological modules, which act autonomously.

A PCD file was used to describe the technical functions of a PROFINET CBA device. Again, this file is based on XML and stored in the library. To interconnect the components, we just need to connect the graphical components without having to carry out complex programming and without the need for prior knowledge of the devices. After interconnection, the user program can be downloaded to the appropriate devices and they begin to transmit data automatically.

For those, who have existing PROFIBUS systems, you can easily integrate your PROFIBUS into PROFINET by using the proxies. Especially in the case of PROFINET CBA, you can integrate the entire fieldbus applications into the PROFINET. In this way, you can protect your already existing investment and know-how, while updating to new technology.

The use of information technology in industrial automation is growing fast and becoming more and more widespread. The Ethernet standard enables the seamless integration of IT from the office area to the automation area. This through-connection is very useful for SCADA (Supervisory Control And Data Acquisition) systems and reduces the cost of, e.g., wiring and engineering. The continuous further development of PROFINET assures the users of a long-term perspective for the implementation of their automation tasks. The future of industrial automation is also secured.