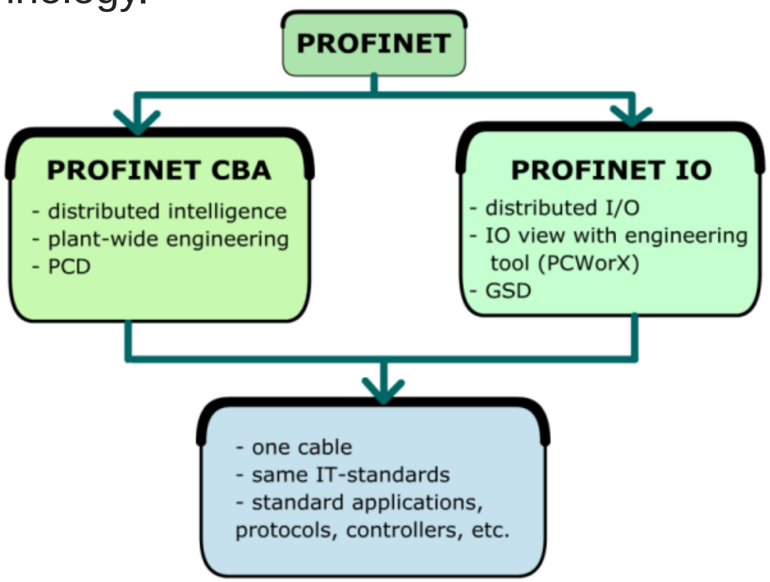
CoNeT Mobile Lab 3:
PROFINET on Phoenix Contact Platform
- ENGINEERING -

Introduction
1 Profinet IO
2 Profinet CBA
3 Profinet live

CoNeT - Co-operative Network Training

PROFINET in one statement

PROFINET is the open Industrial Ethernet Standard for Automation of PROFIBUS & PROFINET International and covers all requirements of all branches of Automation Technology.



→ Is based on Industrial Ethernet
→ Uses TCP/IP and IT standards
→ Is automation in real-time
→ Enables seamless integration of fieldbusses

- You can use PROFINET for your factory and process automations that require response time under 100 milliseconds.
- Other applications, such as drive technology applications or clock-synchronized Motion Control, can be served with response time in less than 1 millisecond.
- To run safety applications, you can use PROFIsafe

Milestones

Technical milestones to market leadership

2006 TCI – Tool Calling Interface available
2005 PROFIdrive and PROFIsafe for PROFINET available
2004 PROFINET IO based on real-time (RT) and isochronous real-time (IRT)
2003 PROFIsafe appears on the market
2002 PROFINET part of IEC 61158/IEC 61784: 10 profiles available
2001 Presentation of the Ethernet-based PROFINET (CBA)

Organisational milestones to market leadership

2007 1 Million PROFINET devices installed
2006 Founding of PI Training Centres
2004 PROFINET Marketing Working Group in Japan and USA
2002 First PROFINET Competence Center and Test Laboratory

With 1.400 members around the world, PROFIBUS & PROFINET International (PI) is the world's largest organisation for industrial communication. The PI Support Center (PISC) in Karlsruhe, Germany, is the international point of contact and the communication center.

PI support

- 25 Regional PI Associations (RPA)
- 36 PI Competence Centers (PICC) in 21 countries
- 15 PI Training Centers (PITC) in 8 countries
- 10 PI Test Laboratories (PITL) for certification tests

Worldwide support of PI

1 PROFINET IO

1.1 Overview

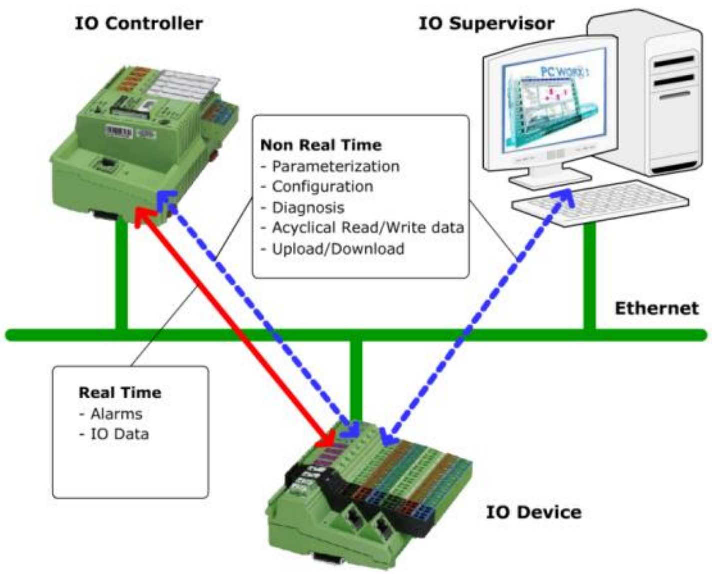
1.2 Communication

1.3 GSD

1.4 Engineering

Device Roles

PROFINET IO is the communication concept for distributed IO devices. PROFINET IO uses the Consumer/Provider model. The device receiving data is called 'consumer' and the device sending data is called 'provider'.



- IO Controller is usually a Programmable Logic Controller (**PLC**) 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.
- IO Supervisor is usually an engineering station. This could be a programming device, PC or HMI device for commissioning or diagnostics.
- IO Device is a distributed field device assigned to one of the IO controllers.

Services of Profinet IO

- I/O Data Objects: I/O data objects are transported cyclically between a provider (IO Device) and a consumer (IO Controller). This transmission occurs without acknowledgment. The interval is configured using engineering tool (IO Supervisor).

- 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, identification and maintenance, I/O data objects, etc.

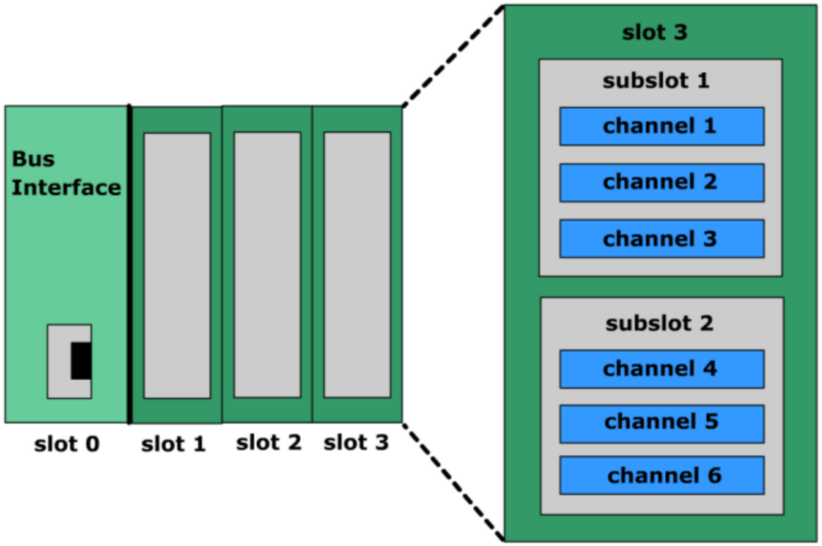
- Alarm Data Objects: Alarms are used to transfer events that have to be acknowledged by the application process. There are two kinds of events:

  System-defined → e.g., insertion and removal of modules
  User-defined → 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. It consists of slots.



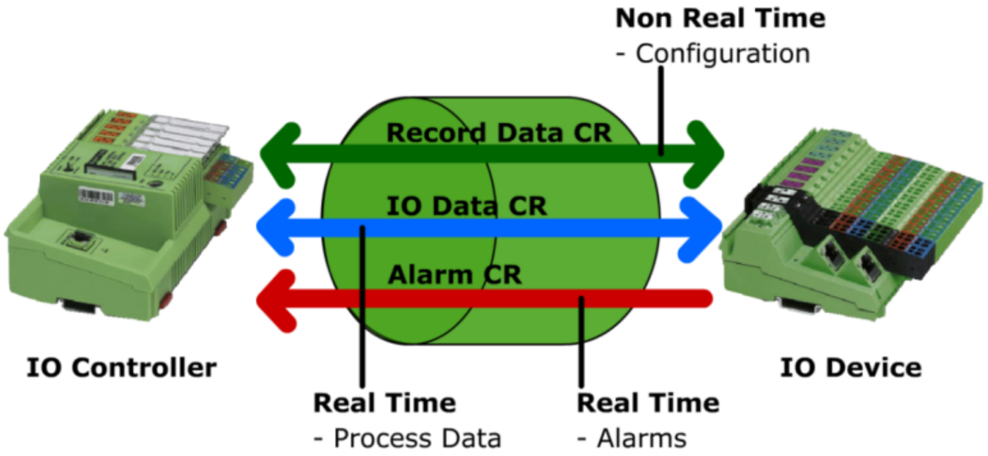
- Slot: Physical place of an IO device. The vendor determines the technical capabilities of a module and defines them in a GSD file.
- Subslot: Is used for addressing. A slot must contain at least one subslot that contains channels. Subslots contain the data for a cyclic data exchange. The definition is contained in the GSD file.
- Channel: They are used to read and send 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.

Communication Services

In order to enable the cyclic or acyclic data exchange between an IO Controller/Supervisor and an IO Device, an 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 **API**s (Application Programmers Interface) that allow fine-tuning of the appliction group. An AR can contain one or more Communication Relations (**CR**).



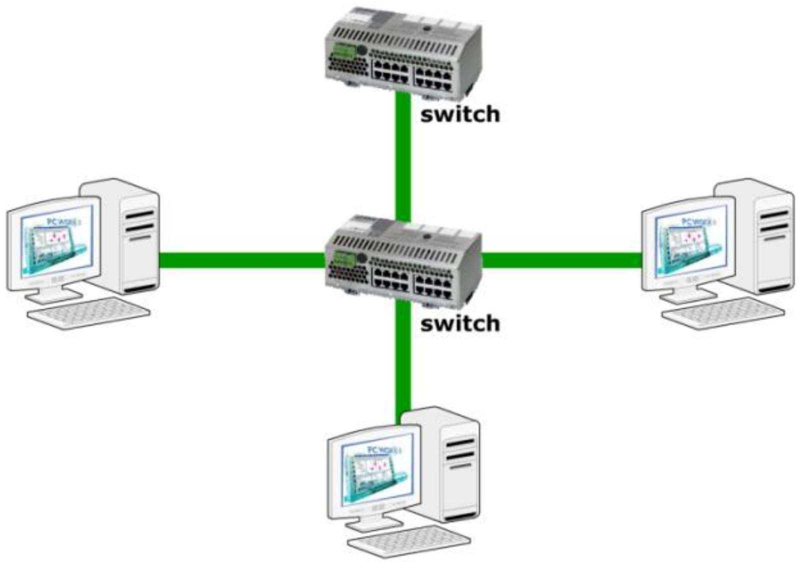
The CRs that are available for every API:

- One or more IO CRs, for cyclic data exchange
- Alarm CR, for events
- Record Data CR, for acyclic data exchange

Communication overview

Communication in PROFINET is based on the **switched Ethernet** mechanism. Switched Ethernet means that every PROFINET device is connected directly together by a **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.

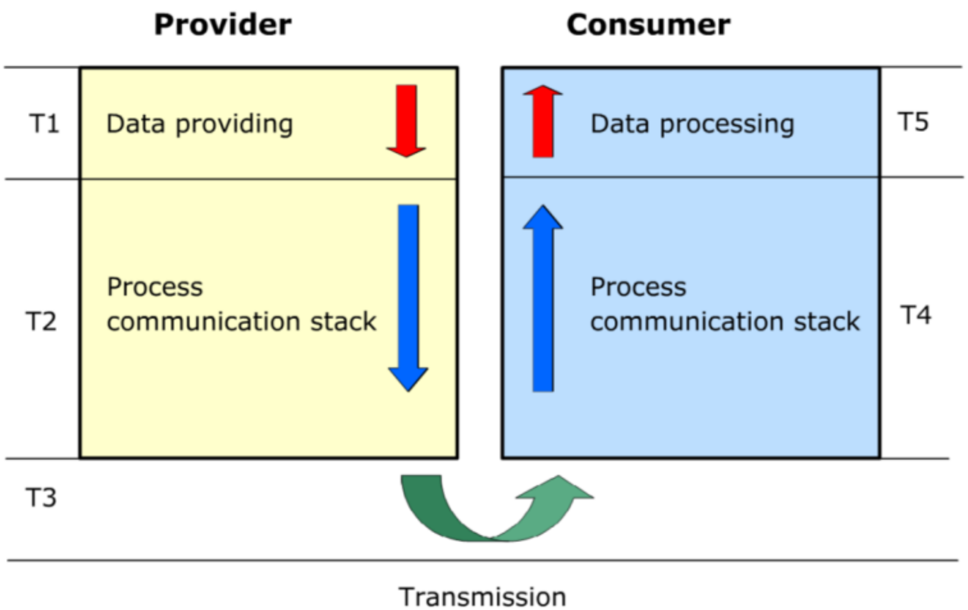


Processing time for data transmission can be divided into five parts:

- T1: Providing the data
- T2: Passing the communication stack
- T3: Transferring to the consumer
- T4: Receiving by the consumer
- T5: Data processing

Non-Real Time

In PROFINET, for the acyclical, non-time-critical processes, such as parameterization, configuration, or HMI, communication with the standard Ethernet mechanisms over TCP/IP or UDP/IP according to the international standard IEEE 802.3 is used. This kind of communication is called Non Real Time communication (**NRT**) .



IP addresses are used for communication between two or more networks.

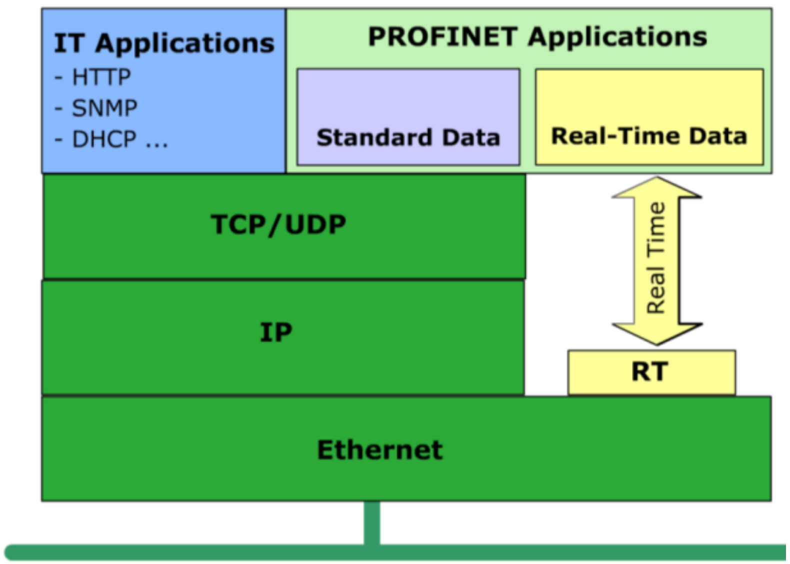
The MAC addresses are used to identify the devices within a network.

All PROFINET devices must support data communication over UDP/IP.

The typical cycle times for NRT are 100 ms.

Real Time

For the transmission of cyclical, time-critical process data, the Real Time communication (**RT**) is used. In this way, the same performance class as in existing fieldbus systems is achieved. 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.



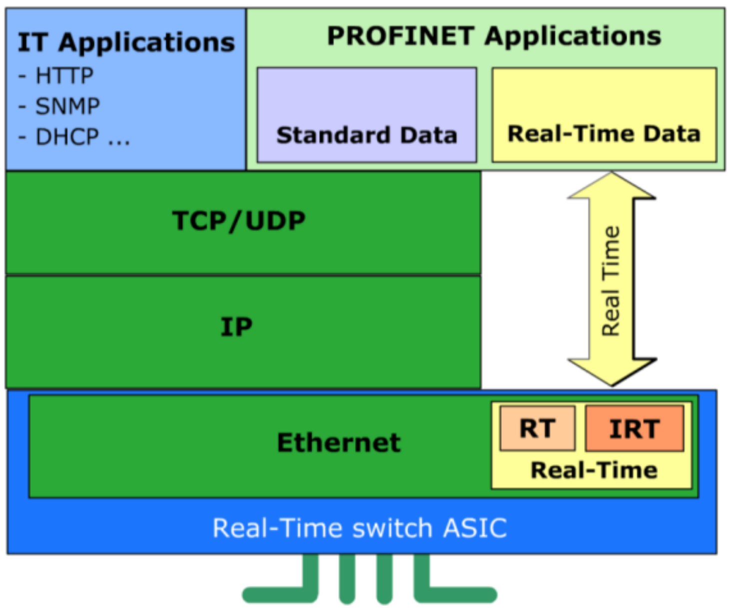
There are three options for RT communications:

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

The typical cycle times are 5 - 10 ms.

Isochronous Real Time

For the clock-synchronized applications, especially in the area of Motion Control applications, a hardware-based Isochronous Real Time (**IRT**) is implemented. We use **ASIC**s (Application-Specific Integrated Circuit) with switch functionality and cycle synchronization to actualize this communication.



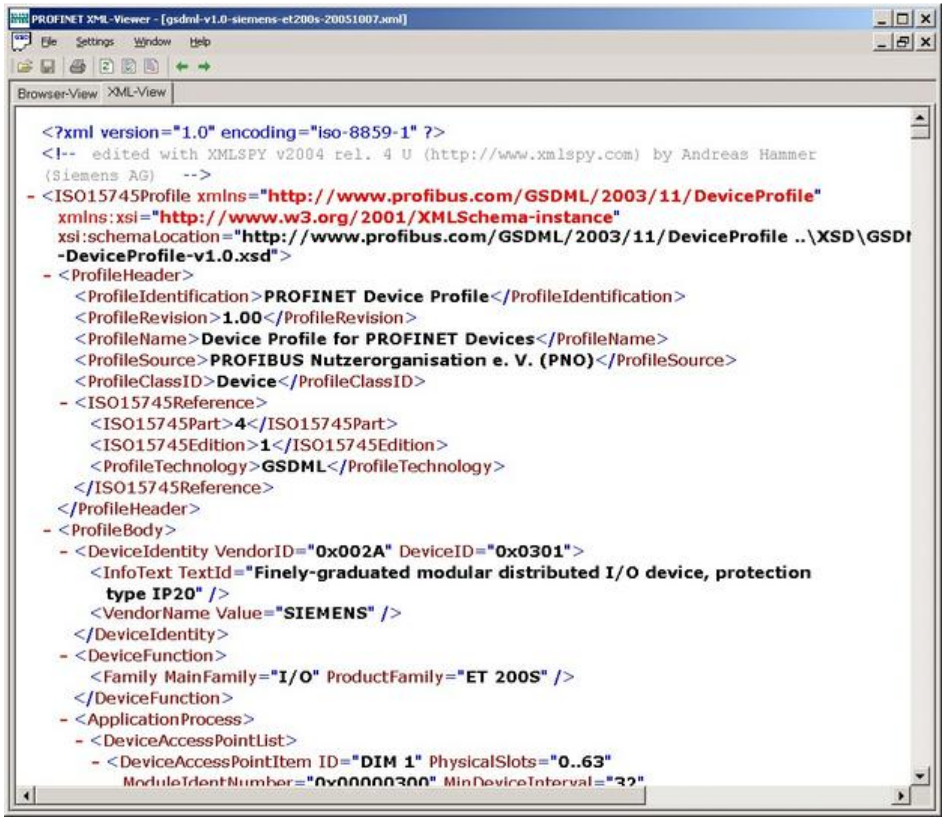
The communication cycle is split into a deterministic part and an open part. The deterministic channel carries cyclical IRT frames, while the TCP/IP and RT frames are transported on the open channel. In this way, both types of data transmission exist together without interfering with one another.

Note: If 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 ms with jitter less than 1 µs.

GSD overview

Every PROFINET IO field device must be described with a General Station Description file (**GSD** file). GSD 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.

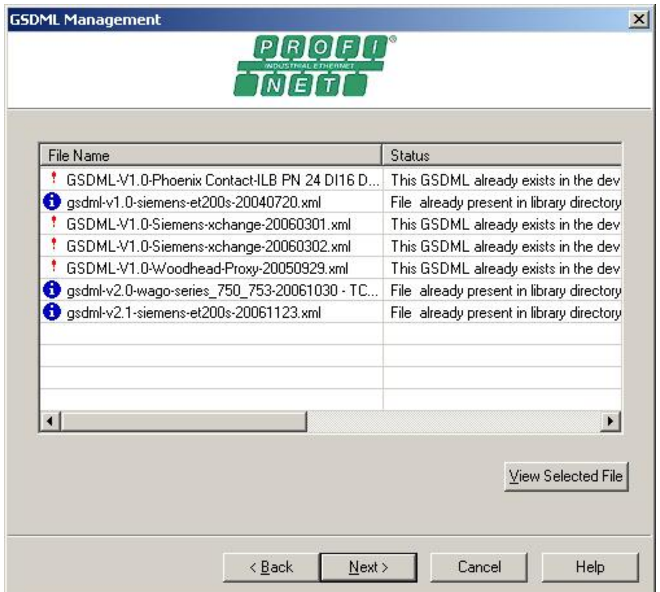
The GSD file is multilingual and can be edited with most text editor programs.

Since a GSD file is written in XML, the language describing GSD files is called the General Station Description Markup Language (**GSDML**).

Naming of GSD files

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

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



`GSD schema version:` The version of the used schema.

`Manufacturer name:` The name of the device manufacturer. It can contain hyphens and spaces.

`Device family name:` The name of the device family. It can contain hyphens and spaces.

`Date`: has the format yyyymmdd. Different GSD files with the same date and device family name are not allowed.

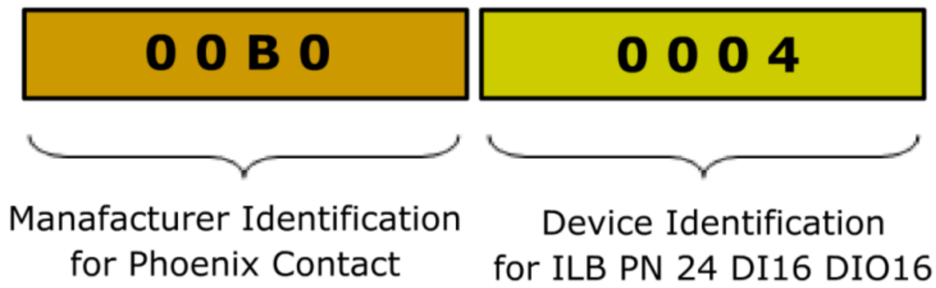
A valid GSD file name can look like this:

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

Some GSD files apply to a family of products, and each individual item, takes up a section of the file

Unique Device Identification

Each PROFINET IO field device is identified by a worldwide unique device identification number. This number contains two parts, the Vendor_ID and Device_ID. Each part of the number is 16 bits long.

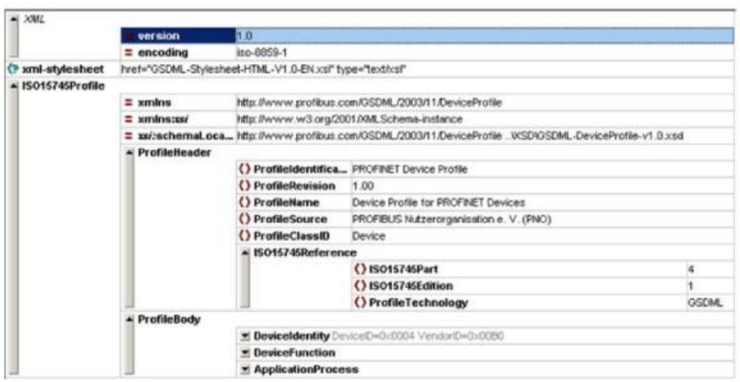


- Vendor_ID: Is used to identify the manufacturer and can be obtained from the PROFIBUS Nutzerorganisation e.V. A device manufacturer needs only one Vendor_ID.
- Device_ID: Is used to identify the field device and can be assigned by the manufacturer.

Structure of a GSD file 1

The GSD is based on the ISO 15745 standard and contains two parts: profile header and profile body.

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



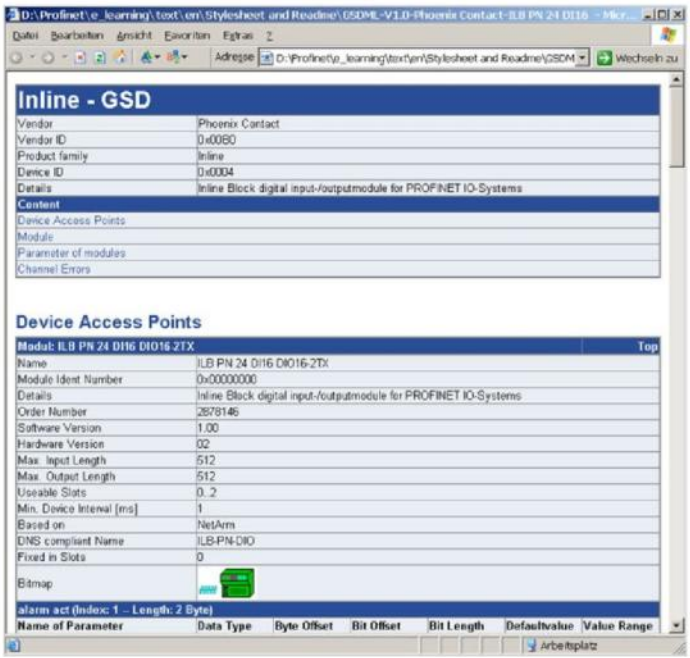
Profile Header contains the common definitions specifying the functions of the field device. The reference to the international standard ISO 15745 is also found here.

Profile Body 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

Structure of a GSD file 2

ApplicationProcess block contains the following parts:



Example of a GSD file with StyleSheet from Phoenix Contact

- DeviceAccessPointList: Description of the degree of expansion of the individual interface modules.
- ChannelDiagList: Channel for errors and error text.
- GraphicsList: Reference to the graphic representing the device.
- CategoryList: Is used to build categories for the engineering tool.
- ExternalTextList: All text that can be referenced by the other segments.

Note: The GSD files and BMP file belong to Phoenix Contact GmBh, while the Style Sheet file belongs to the PI. All rights remain with the respective authors.

Engineering overview

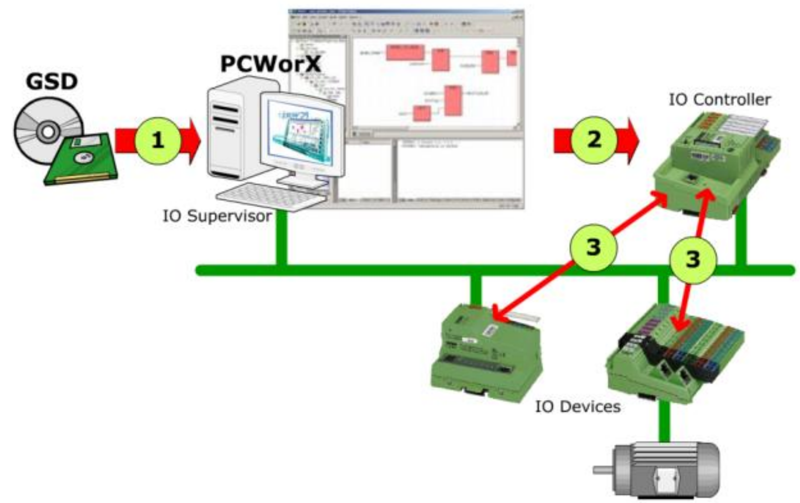
Basically, in order to set up a system the following components are needed:

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

Engineering steps

STEP 1: 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.

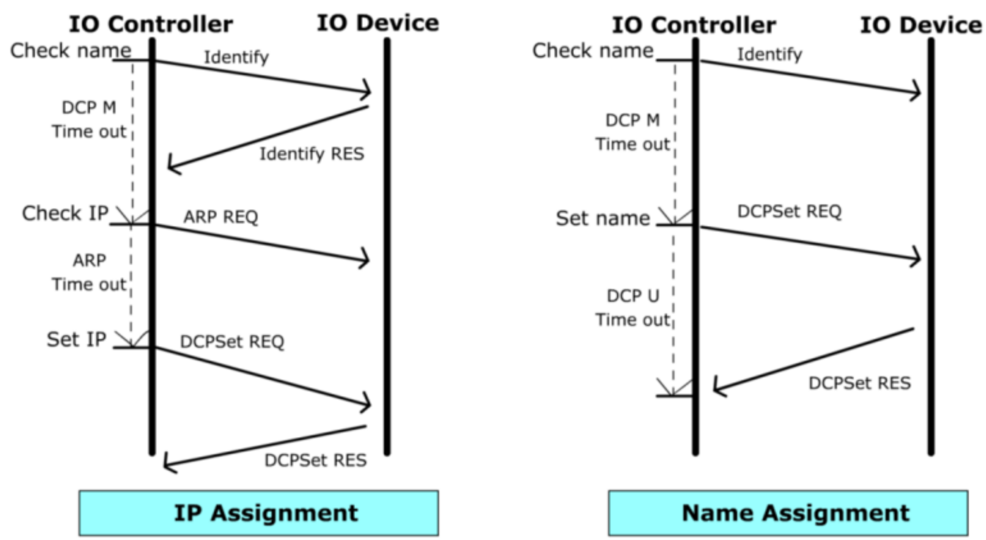
STEP 2: After completion of the engineering process, the configuration information and user programs are transferred to the IO Controller.



STEP 3: The IO Controller independently takes over the data exchange with the decentralized IO Device automatically.

Address Assignment

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



As delivered, every field device has, among other things, a MAC address and a symbolic name. This information can be used to assign the name of the respective field device.

The address assignment can be done using the DCP protocol as follows:

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

Device replacement

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

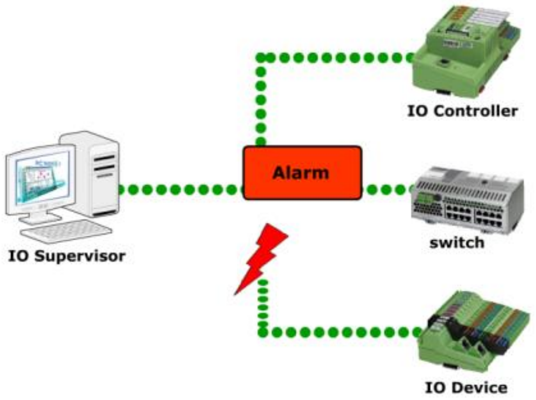
If you remove the memory card from a PROFINET device and insert it into a different PROFINET device, the device name is transferred. In the case of active network components such as switches, the full configuration data is directly on the C-Plug.

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 1

In PROFINET IO there are three main error diagnostics:

- 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 the example, a wire between an IO Device and a switch has been interrupted. The switch generates an alarm and sends it to the IO Controller, which also contains information about the port affected.

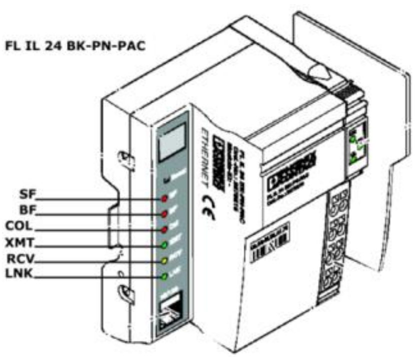
In addition to the device, in PROFINET the network infrastructure can be diagnosed via the SNMP protocol.

Additionally, diagnostic information can also be read directly from an IO Supervisor.

Diagnostics and Alarms 2

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 System Status List (**SSL**). The IO Supervisor accesses the SSL in the IO Controller. The malfunctions can then be visualized in the **HMI** (Human Machine Interface).
- 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 Supervisor and IO Device must be connected to the Ethernet.



Status LEDs Inline Bus Coupler FL IL 24 BK-PN-PAC from Phoenix Contact

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)
XMT	Green	ON	Data telegrams are being sent
		OFF	Data telegrams are not being sent
RCV	Yellow	ON	Data telegrams are being received
		OFF	Data telegrams are not being received
LNK	Green	ON	Physical network connection ready to operate
		OFF	Physical network connection interrupted or not present

A further option is the reading of the errors directly from the status LEDs on the device.

2 PROFINET CBA

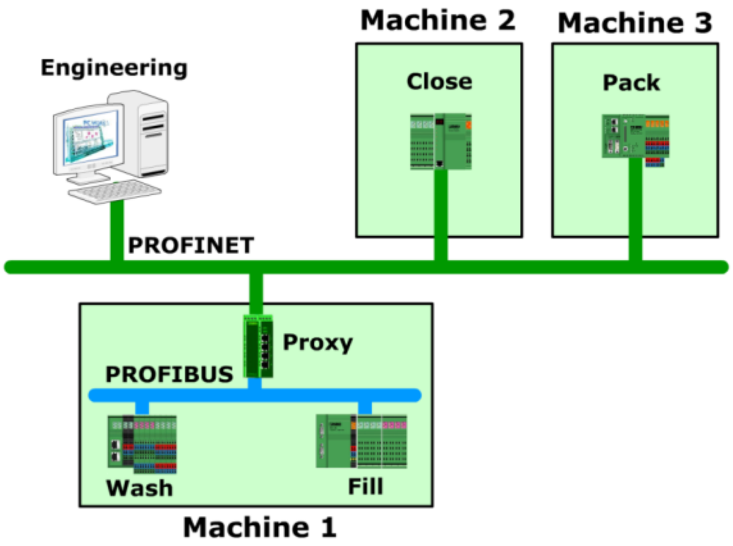
2.1 Overview

2.2 Engineering

2.3 Runtime model

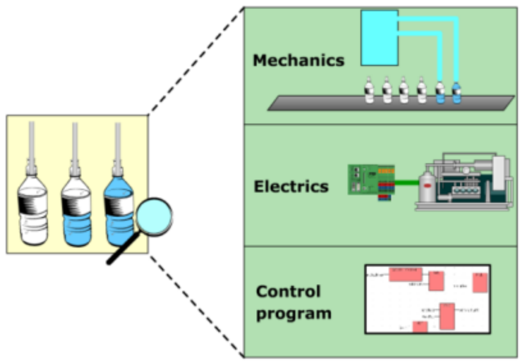
2.4 Component description

2.5 PROFINET CBA and PROFIBUS



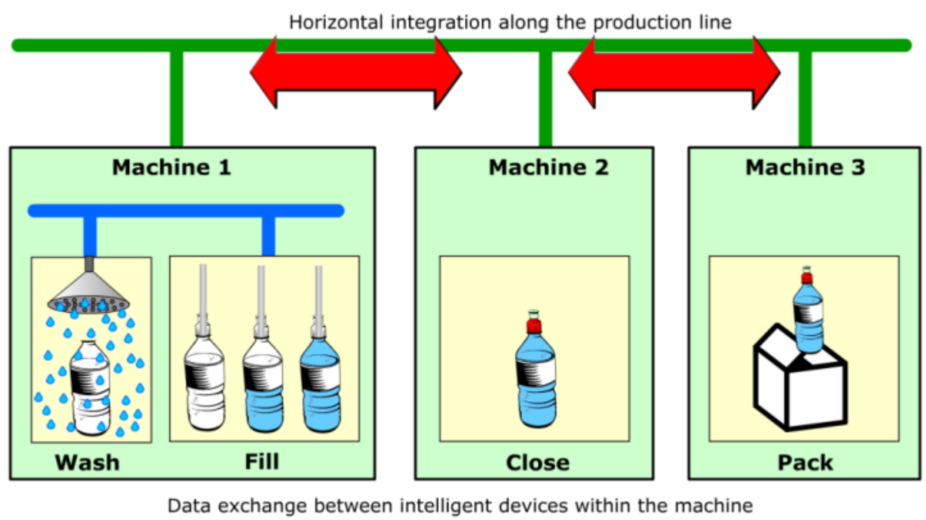
Technological Modules

PROFINET Component Base Automation (**CBA**) provides a modularized automation concept between the plants. A technological module in PROFINET CBA consists of Mechanical, Electrical or Electronics and Control Logic or Software.



The three components of a technological module describe the function of an automated plant or machine. They operate with each other during the product manufacturing process.

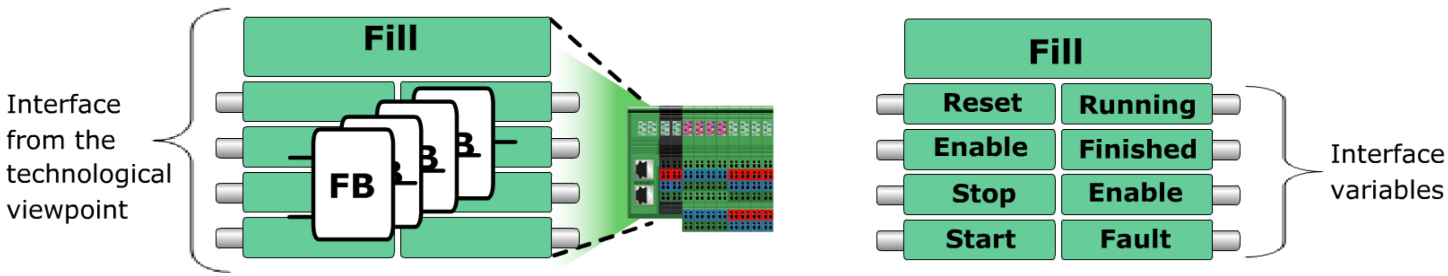
Thus, a technological module represents a system-specific plant or machine completely.



Component model 1

The PROFINET CBA is based on a component model, 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 are considered as technological modules. The functions of these components are encapsulated in the unique COM objects, called PROFINET components.

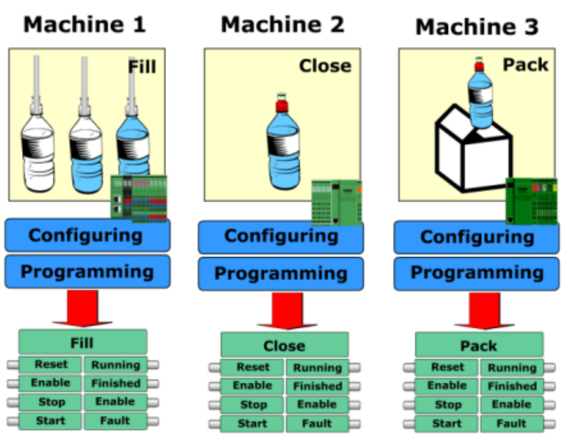


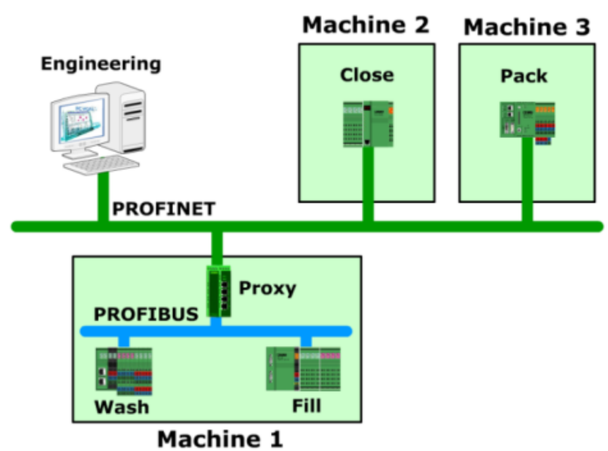
COM (Component Object Model) was developed by Microsoft. PROFINET components are also COM objects. A PROFINET component can be used as a black box with uniquely defined interfaces. Therefore, you can connect PROFINET components to any other PROFINET components.

Component model 2

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 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

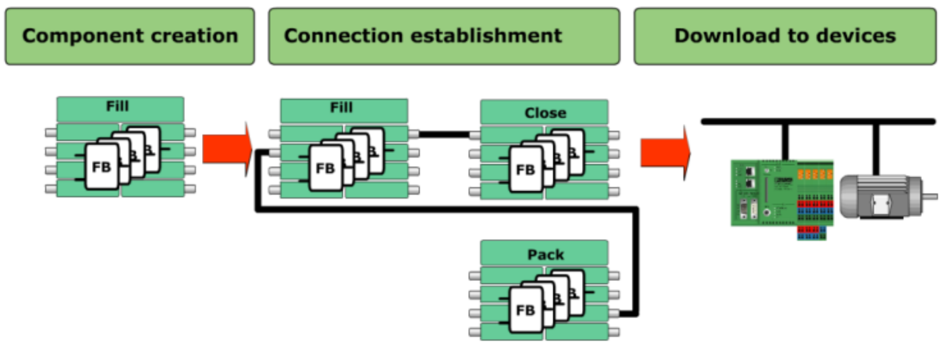




This distributed automation system design is necessary to allow modularization of your systems and machines. Moreover, the reuse of system or machine parts is possible, which can result in engineering cost reduction.

Engineering

For the engineering of PROFINET a vendor-independent engineering concept is defined. This concept is based on an engineering object model. You can use the components from different vendors as well as vendor-specific or user-specific functional extensions with one engineering tool.



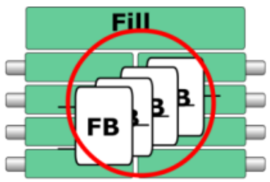
A system-wide application can be built in three phases:

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

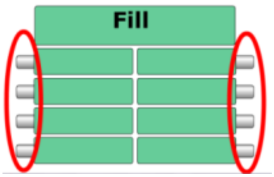
The data exchange between nodes is ensured by the configured connections. An XML-based file describes the technical functions of a PROFINET CBA device. This file is called PROFINET Component Description (**PCD**) and is provided by the device manufacturer.

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 simply by extending the new component interface. In this way, existing programming know-how can be used and protected.



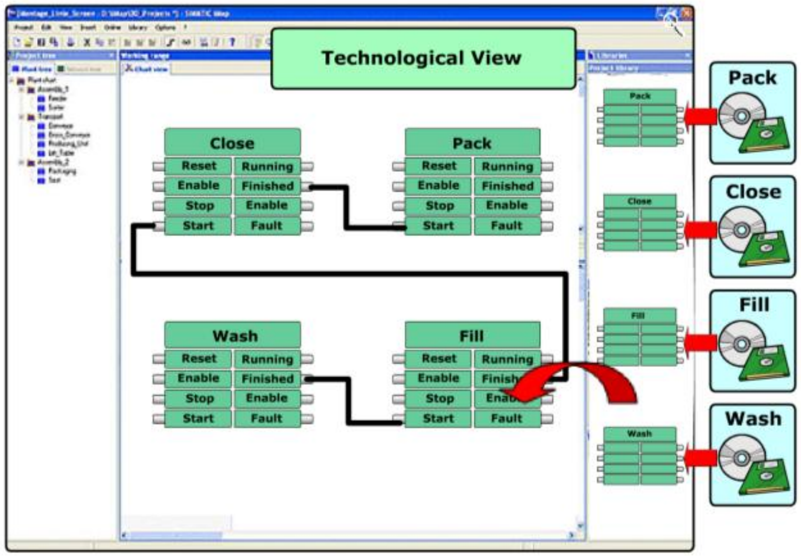
- The component interface will be defined. The interface consists of several variables, such as Name, Type, PROFINET direction and Comment.

- 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

We will use the connection editor to interconnect the PROFINET components, which are represented by a PCD file. The components can be selected from a library into the application by using a mouse click.

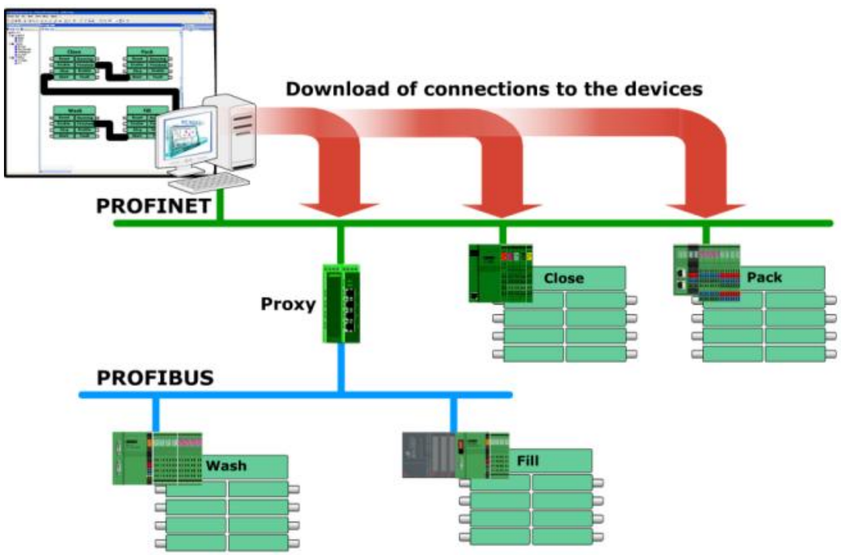


This step simplifies the interconnection process by using the simple graphics-based configuration. Complex programming and knowledge of communication functions in the device is no longer required.

By using a connection editor, you can combine the individual distributed applications throughout the system. Different PROFINET components from different manufacturers can be interconnected together. You can specify the frequency of transfer in 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. That means that each component knows its own communications partner, communications relationships and the information to be exchanged.



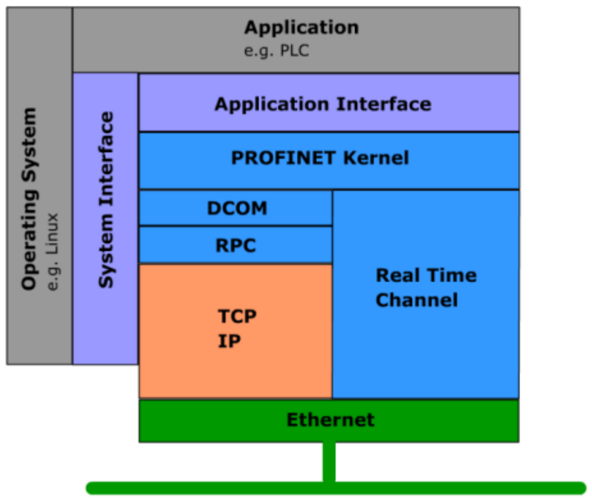
The establishment of the communications connections to the partner and the data exchange occur automatically.

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 work based on the TCP/IP or UDP/IP for Non-Real Time communications and RT for Real Time communications.

Runtime Model

In the PROFINET component model, the Distributed Component Object Model (**DCOM**) is used and defined as a shared application protocol that is based on TCP/IP. DCOM uses the standardized Remote Procedure Call (**RPC**) to enable communication between PROFINET components over the network.



DCOM is included by default in the available software stack. If there is a need to use time-critical applications, the PROFINET RT channel will be used.

The configuration of data transfer can be done in the configuration tool. The transfer rate is also called Quality of Service (**QoS**).

Every exchanged data frame will be identified with a Quality Code (**QC**). With this code, the user can decide how to process the data.

80	**The value is good.**
00	**The value is bad.**
4c	**The value is an initial value.**
44	**Last valid value is retained.**

TCP/IP Stack and DCOM

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** (Real Time Operating System).

The DCOM 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.

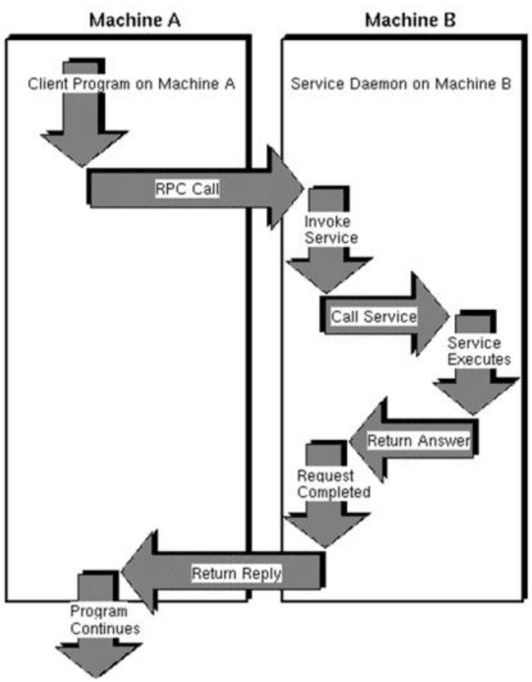
The DCOM component must be integrated with four other components:

- RPC
- RTOS
- DCOM Application Interface
- Diagnostic module

The standalone DCOM provided by PI 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.

Remote Procedure Call and Auto Marshaler

The RPC transports DCOM requests between PROFINET runtime software and the TCP/IP stack. RPC is a standard component of Microsoft Windows.



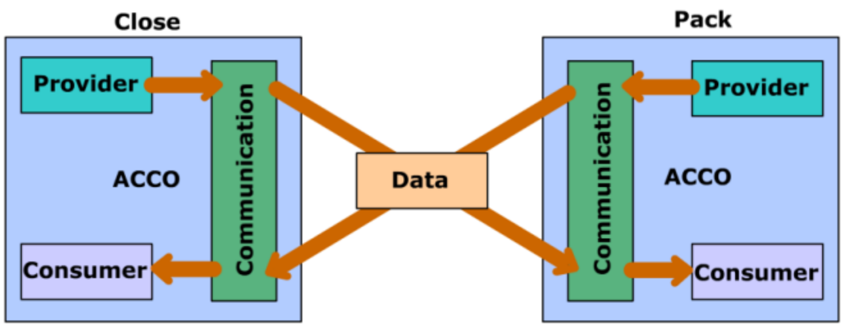
For embedded systems running a non-windows OS, the PROFINET runtime software includes a standalone RPC.

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.

The Auto Marshaler (**AM**) serializes the components of a parameter call prior to handing the marshaled call to DCOM. AM 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.

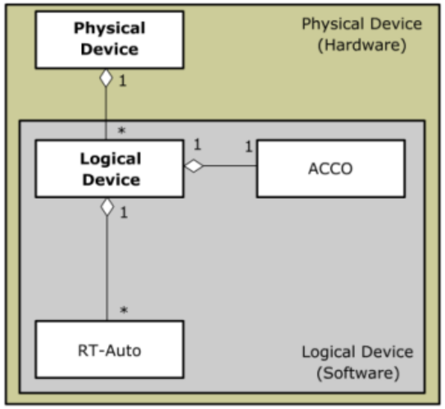
ACCO and RT-Auto

The Active Control Connection Object (**ACCO**) is a software component in the PROFINET Runtime Software core. The ACCO ensures the coordination of the data exchange in a PROFINET CBA device. It is integrated in every PROFINET CBA device and used to monitor the communication relations.



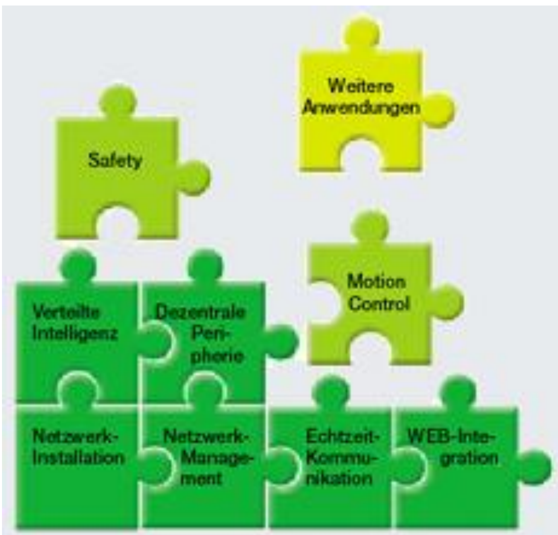
As a provider, the ACCO makes sure that the requested data are sent automatically on time. As a consumer, the ACCO establishes and monitors the communication with the relevant provider based on the configured QoS.

A RunTime Automation Object (**RT-Auto**) is an executable program with a corresponding data range and contains the executable function of the field device. More than one RT-Auto can be implemented in a field device. Different functions are assigned to different RT-Autos.



PROFINET CBA PCD

PROFINET Component Description (**PCD**) 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.



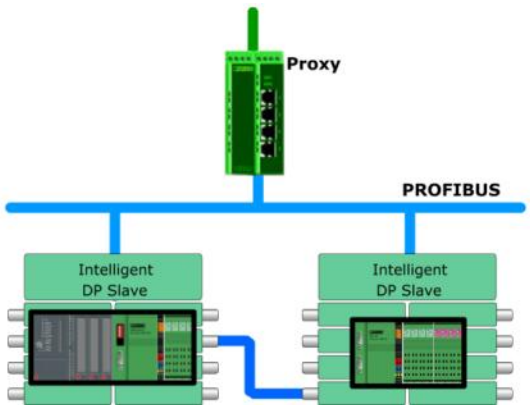
Components of a PCD file

- Description of the component as a library element
  - Component Identification, Component Name (e.g. Fill)
- Description of 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)
- Storage location of the component project

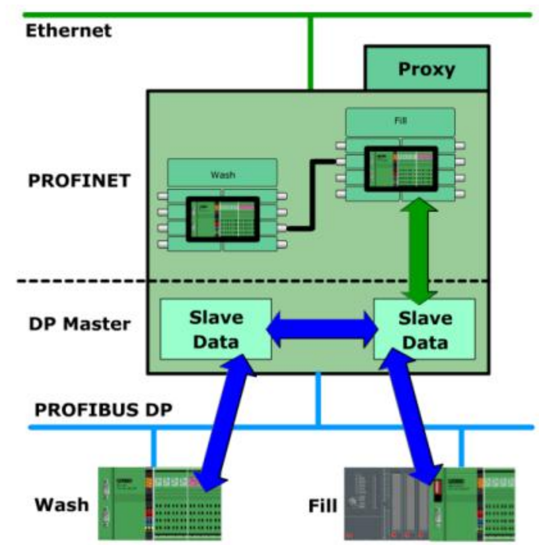
A PCD file is usually created after the creation of the application software by using the vendor-specific tool, e.g. *PC WORX* by Phoenix Contact. The PCD can be loaded into the engineering tool, e.g. Siemens *iMap*. Alternatively, you can obtain the vendor-independent PROFINET Component Editor from PNO.

Connection using proxies

If there is an existing PROFIBUS system, you have two possibilities to integrate it into the PROFINET system by using proxy and by integrating fieldbus applications.



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

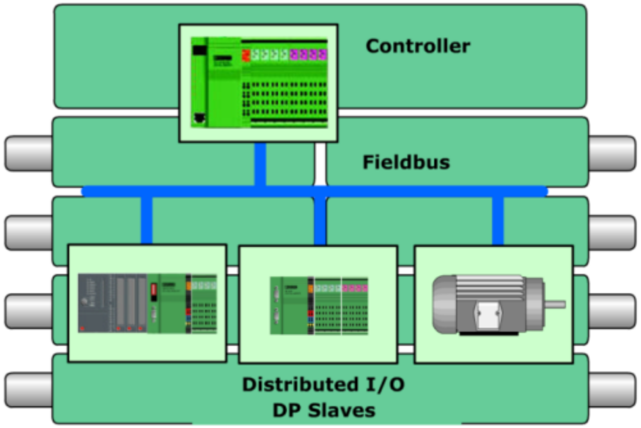


The proxy is a PROFIBUS master on one side, and on the other side, it is an Ethernet node. Proxies can be implemented as e.g. PLCs, PC-based controllers, or pure gateways. Proxies make communications between devices on different bus systems transparently possible.

Connection by integration of fieldbus apps

An entire fieldbus application can be modeled as a PROFINET component in the framework of the component model. 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 enables users to protect their investment in existing systems and wiring. Moreover, existing know-how in the user programs is preserved. This makes the seamless transition to new system segments with PROFINET possible.

3 PROFINET live

3.1 Duesseldorf Telelaboratory

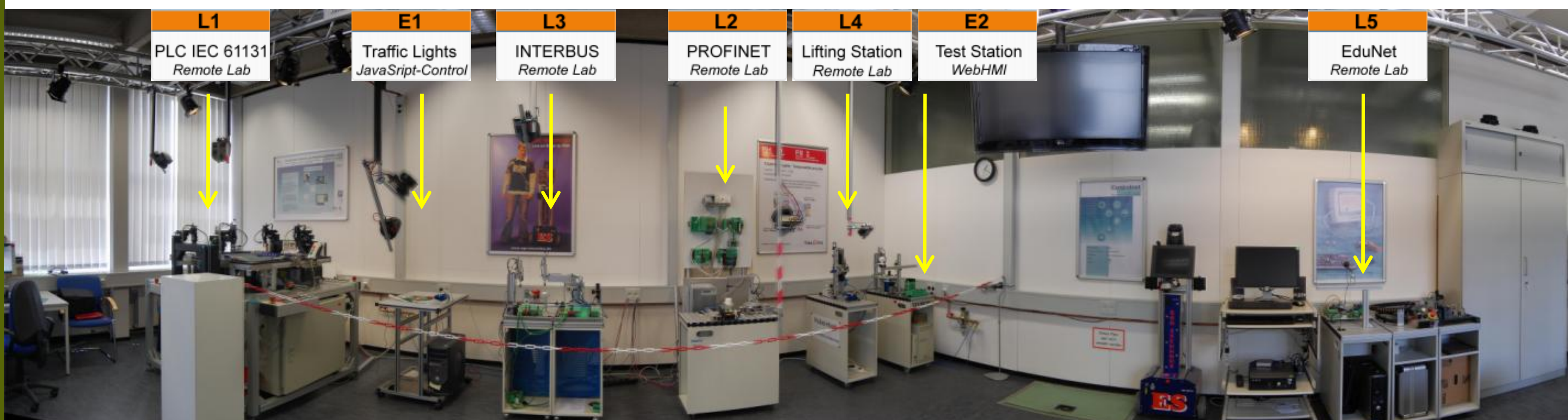
3.2 PROFINET remote lab

Duesseldorf Telelaboratory

Düsseldorfer Telelabor

Duesseldorf Telelaboratory (DT) was founded in January 2004 as a non-profit R&D institution.

In general DT (**www.telelabor.de**) is a platform for development and testing of new concepts of remote engineering for complex technical systems in the industry.



At present the DT offers five remote labs (L1 ... L5) as well as two open remote test environments (E1, E2) with many remote-controlled experiments.

E-learning & doing in PROFINET remote lab

Learning task

Configuration and programming of a PROFINET IO system in order to control a handling station.

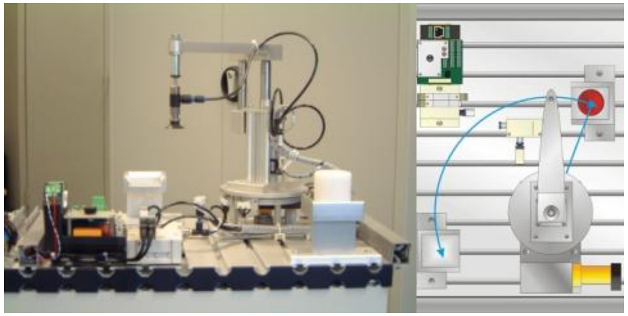
Test it using
“PROFINET Remote Lab”

http://www.telelabor.de

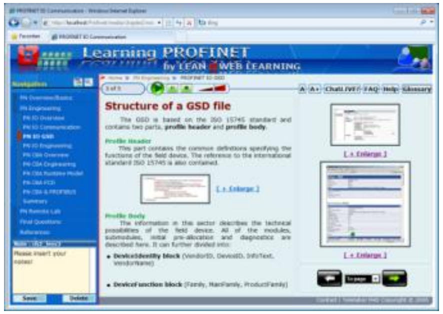
Learning procedure

Configuration of a PROFINET IO system by PC WORX and integration of a INTERBUS system by using a proxy. Programming a simple PLC program in order to control the handling of a workpiece from one storage place to another.

Learning equipment



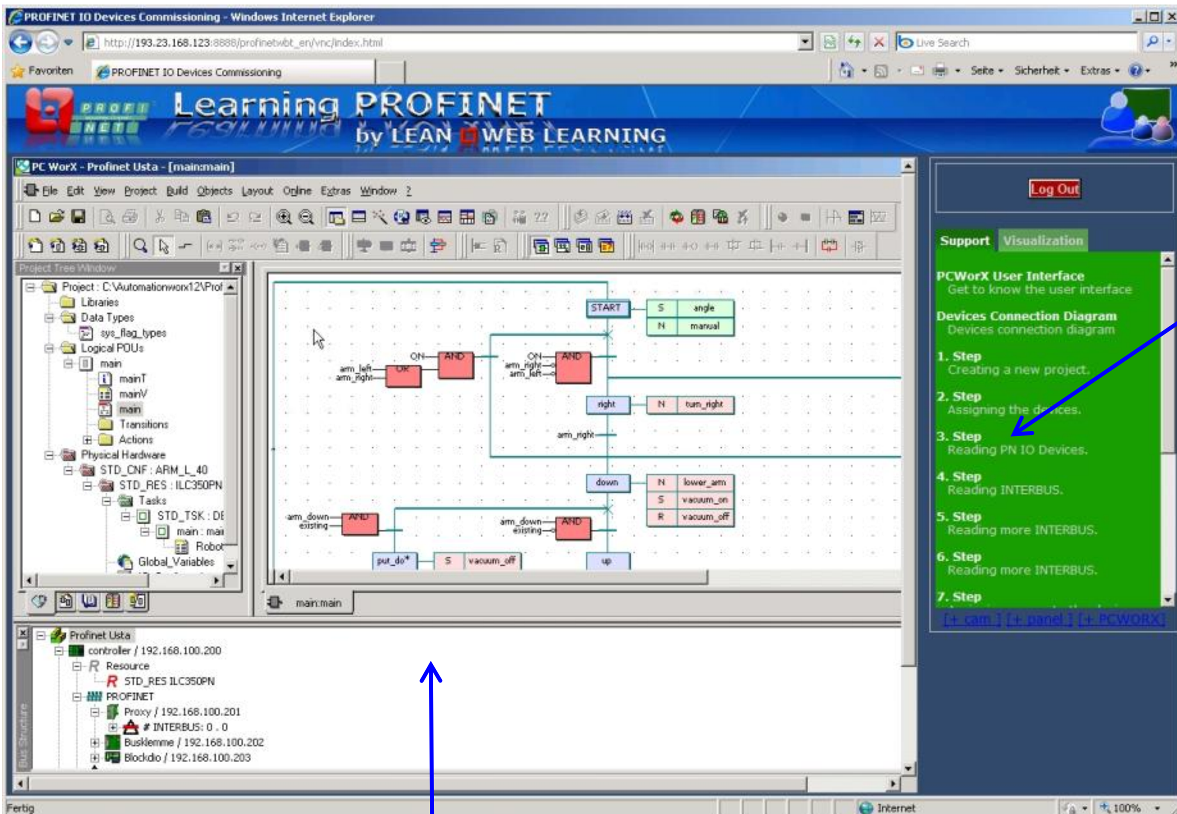
Handling station



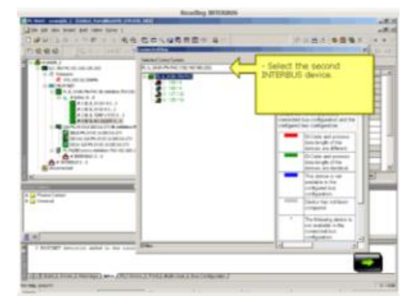
Web-based training system (basics and engineering)

Webpage Remote Lab

Learning pages in the PROFINET remote lab

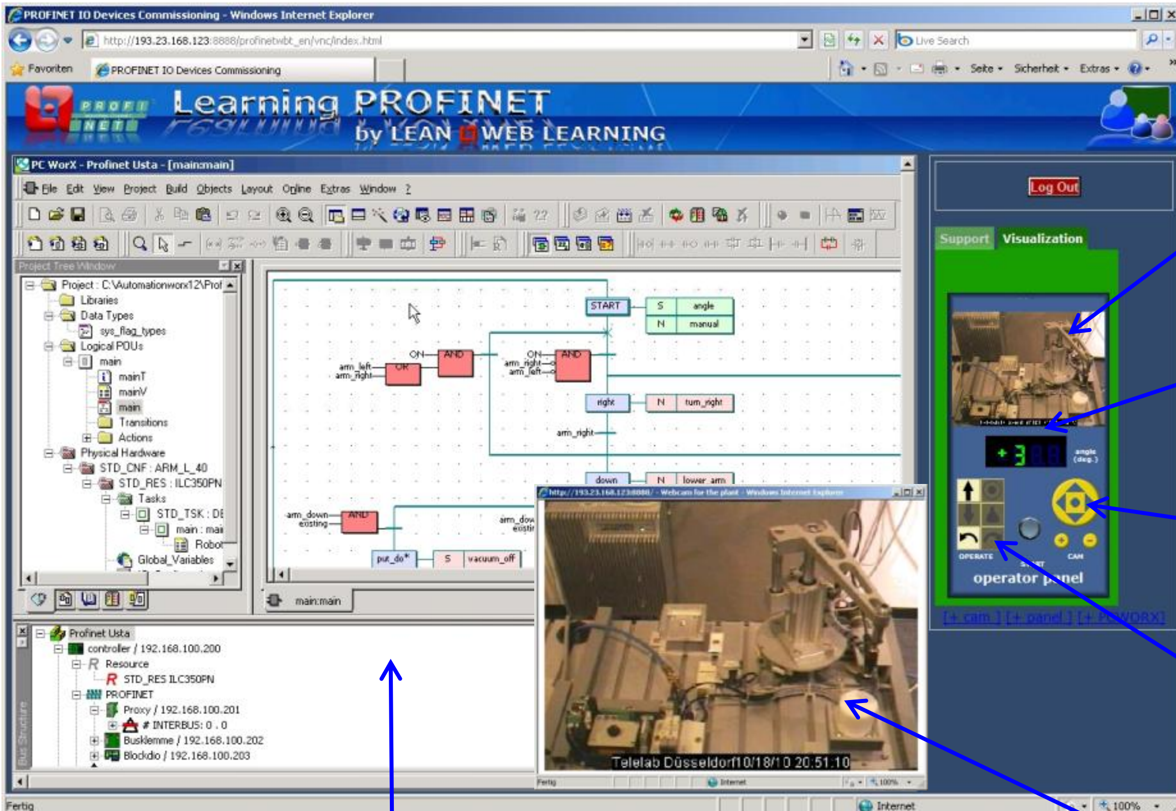


Different flash movies for the help of PROFI-NET configuration by PC WORX



VNC remote desktop in order to work with PC WORX (configuration of PROFINET IO, PLC programming)

Learning pages in the PROFINET remote lab



Webcam for the handling station

Visualization of the rotation angle of the table

Operator panel of the Webcam

Operator panel of the handling station

Scaled up Webcam window

VNC remote desktop in order to work with PC WORX (configuration of PROFINET IO, PLC programming)

Summary

PROFINET IO uses the provider/consumer model and has three device roles, IO Controller, IO Supervisor and IO Device. The connections between them can be Non-Real Time (NRT), Real Time (RT) and Isochronous Real Time (IRT) depending on their requirements for data transmission. The technical functions of PROFINET devices are described by GSD files.

PROFINET CBA is designed for distributed automation. PROFINET CBA uses the component concept that is based on DCOM. With the proxy concept, you can integrate classical fieldbus systems into PROFINET.

In the Duesseldorf Telelaboratory (`www.profinet-lab.de`) you can test and experiment with a real PROFINET IO system for automation of a handling station.

References

Popp, M.; Weber, K.: The rapid way to PROFINET. - PROFIBUS Nutzerorganisation e.V., Karlsruhe, Germany, 2004

Popp, M.: Das PROFINET IO Buch. - Hütig GmbH & Co. KG, Heidelberg, Germany, 2005

Bormann, A.; Hilgenkamp, I.: Industrial Networks. - Phoenix Contact GmbH & Co. KG, Blomberg, Germany, 2006

PROFIBUS International: PROFINET architecture description and specification. - Version 2.01, `http://www.PROFIBUS.com`

Siemens AG: PROFINET System Description Siemens. - `http://support.automation.siemens.com/WW/view/en/25571355`

Pigan, R.; Metter, M.: Automating with PROFINET, 2008, ISBN 978-3-89578-294-7