



REAL TIME AUTOMATION



PROFINET IO

Another Innovative Distributed
Automation Solution



PROFINET IO

Another Innovative Distributed Automation Solution

Introduction

This paper presents an overview of PROFINET CBA, a high-level network for industrial automation applications. Built on standard DCOM and RPC technologies, PROFINET uses traditional Ethernet hardware and software to define a network that structures the task of configuring, accessing and controlling industrial automation devices. PROFINET is based on the object interface structure defined by the Distributed Component Object Model (DCOM) that, based the history of Microsoft inter-process communications, is a descendant of DDE, OLE and Active X.

PROFINET CBA views a system as a series of "Technological Components". These components act independently and coordinate their activities to form an integrated system. DCOM is a good choice for such a system. It is an object oriented mechanism which structures how a Client (data requester) can locate, request, and transfer data from a Server (data source). Building on the DCOM model, PROFINET CBA strives to provide a seamless integrated system from the sensor-actuator network to the controller and enterprise networks.

PROFINET CBA is much more than PROFIBUS on Ethernet. In fact, the name is very confusing. PROFINET has little in common with PROFIBUS. PROFIBUS is a data oriented mechanism for transferring I/O from device to device over a cable at a maximum speed of 12Meg baud. PROFIBUS data exchange is mainly cyclic with devices that have a prior understanding of the data structure and meaning. In PROFINET CBA, data can be located, information about its structure can be obtained, and data can be exchanged on event triggers, cyclic schedules, or other mechanisms. PROFINET operates on Ethernet at ten to 100 mega baud speeds.

PROFINET uses more traditional Information Technology (IT) concepts and software than all of the other common networks for Industrial Ethernet applications. This has both advantages and disadvantages that will be apparent as the concepts in this paper are developed.



Rocco knows:

**PROFINET and PROFIBUS
are not the same.**

A LITTLE BACKGROUND

Most people who work in an office associate the term "Ethernet" with the physical cable behind their desk. This cable connects their office PC to the printers and Servers of the local network and the infinite web sites on the Internet. This cable is only the physical part of Ethernet, the media carrying Ethernet messages to your PC. On this wire are a whole series of communication protocols such as IP, the Internet Protocol; TCP, the Transport Control Protocol; and various Microsoft protocols such as NetBEUI. This suite of protocols works well for the office environment. It allows users to share files, access printers, send email, search the Internet, and perform all the other communications used in the office environment.

The needs of the factory floor are much different, with some very special requirements. Instead of accessing files and printers, factory floor controllers must access data embedded in drive systems, operator workstations, and I/O devices. Instead of letting a user wait while a task is being performed, factory floor data communications needs are real-time or very close to real time. Terminating the fill operation on a bottle requires much more time-precise communication than accessing the next page of an Internet site.

Traditionally, Ethernet had only limited acceptance in Industrial Automation. Until recently, the expense, lack of sophisticated switches and routers, and domination of large vendors with proprietary protocols prevented the wide acceptance of Ethernet on the factory floor. Now, with prices falling, PCs with inherent Ethernet capability moving in droves onto the factory floor, and intelligent switches and routers, Ethernet is gaining acceptance. Only the lack of a widely accepted, flexible application layer targeted to Industrial Automation has prevented its complete acceptance.



PROFINET IO Benefits

PROFINET IO is a unique industrial Ethernet application layer. It offers many benefits over competing application layers including:

- ◆ **High Speed Operation – The real time communication channel provides high speed message exchange by bypassing the time required to process the TCP/IP stack.**
- ◆ **Seamless and nearly identical Siemens S7 PLC integration to PROFIBUS**
- ◆ **Support for time-critical motion control applications**
- ◆ **Short Startup**
- ◆ **Distributed Intelligence**
- ◆ **Ease of installation**
- ◆ **Minimum commissioning time and engineering support**

PROFINET Device Classification

PROFINET IO classifies devices into three types. IO-Controllers are devices that execute an automation program. Controllers, functionally similar to a PROFIBUS Class 1 Master, exchange data with IO-Devices. IO-Devices are distributed sensor/actuator devices connected to the IO-Controller over Ethernet. In PROFIBUS terms, IO-Devices are similar to PROFIBUS slaves. IO-Supervisors are HMIs, PCs or other commissioning, monitoring, or diagnostic analysis devices. These devices are similar to Class 2 PROFIBUS Masters.

IO-Controllers map IO data from PROFINET IO devices into the process image of the controller. In Siemens S7 Programmable Controllers, IO data, alarms, and status data are mapped into the process image in much the same way it is done for PROFIBUS devices. These data values are then available for use by the control program. IO-Controllers must support the following kinds of services:

- ◆ **Cyclic Data Exchange – The exchange of data between IO-Controllers and IO-Devices.**
- ◆ **Acyclic Data Exchange – The exchange of Configuration and Diagnostic data**
- ◆ **Alarms – Alarm data exchange from an IO-Device to an IO-Controller**
- ◆ **Context Management – Connection processing**

IO-Supervisors are used for commissioning and diagnostic data collection. IO-Supervisors can read and write internal diagnostic data associated with the PROFINET IO stack or diagnostic data provided by the application program of a device. IO-Supervisors can also read and write configuration data using special, non-cyclic record data object services. These types of devices may only be used during the commissioning process or they may be used as an HMI to display diagnostic data to the end user.

A PROFINET IO system requires at least one IO-Controller and one IO-Device. Systems can be configured in various configurations; multiple IO-Controllers for a single IO-Device; single IO-Controllers for multiple IO-devices; and multiple IO-Controllers with multiple IO-Devices.

PROFINET IO Network Representation

The network representation of a device is the view of the device from the network. In Modbus and Modbus TCP, devices are represented as a series of registers (16-bit integers) and coils (bits). In EtherNet/IP and DeviceNet devices are represented as objects. Other networks have other network configurations. In Lonworks, data is represented by a series of “data



tags” that the device provides to the outside world. In every case above, the device presents some view of itself to the outside world. That representation is the network representation.

The PROFINET IO network representation is very similar to PROFIBUS. It consists of a device with slots, sub-slots, and channels. Sub-slots are subcomponents of a slot. Each sub-slot is assigned a number of I/O points or channels. A channel is the PROFINET IO term which refers to one physical discrete input, discrete output, analog input, or analog output. A device can have almost any number of slots, sub-slots, and channels.

Except for Slot zero, every other slot and sub-slot contains status, diagnostic, and alarm data for the channels of that slot. For points transferred to an IO-Controller, provider status and diagnostic information is automatically transferred on every IO scan cycle. For points transferred to an IO-device, consumer status is returned to the IO-Controller.

Even though PROFINET devices and GSD files refer to “Slot zero”, there is no Slot zero. The first IO slot of a device is slot one. Instead of referring to an actual slot, Slot zero refers to the device itself. No IO data is contained in Slot zero. In place of IO data, Slot zero manages all the generic device data like the vendor name, product catalog number, and software and hardware version information, as well as other similar information.

Modules are specific functional components that can be associated with a slot. Modules can be virtual or real. A module must be plugged into a slot before the IO device goes online. The module gives the slot a specific identity. For example, a 16 discrete input module gives the slot a 16 discrete input identity. In the same way that modules provide identity to slots, sub-modules provide sub-slot identity. The 16 discrete input module can be composed of one 16 discrete input sub-module, two 8 input sub-modules or four 4 input sub-modules.

IO-Controllers associate to a device and all the slots and sub-slots. The current version of PROFINET IO only supports devices associated to one IO-Controller at a time. Future versions of PROFINET IO may lift this restriction and associate to IO-Controllers by slot.

In an IO-Controller like a Siemens 317 Programmable Controller, the IO points or channels for the sub-slots assigned to that Programmable Controller are collected and that data group forms the IO data image that is transferred between the IO Device and the IO Controller. For example, if you have a PROFINET IO device with four input slots (16 inputs each) and two output slots (16 outputs each), the IO image transferred between the controller and device is 4 bytes in the direction of the Programmable Controller and 2 bytes in the direction of the IO Device.

In an IO Controller, all PROFINET IO inputs are assigned to the Input data table and all outputs are assigned to the Output Data Table, just as is done for PROFIBUS.

Configuring PROFINET Devices

PROFINET IO devices are configured using a configuration tool that acts as the IO-Supervisor. The IO-Supervisor uses a GSD file similar to the GSD files common to PROFIBUS. Unlike PROFIBUS GSD files, PROFINET IO GSD files are XML based and carry much more information than the PROFIBUS GSD. Because they are XML based, PROFINET GSD files are called GSDML files.

Configuration is transferred from the IO-Supervisor using the Record Data Object (RDO) services. These services allow an external device to read and write data maintained by either the PROFINET IO stack or the application. Record Data is non-real time data. It includes data like configuration, diagnostic and status data. Record data is always transferred non-cyclically in a connection oriented, queued transmission mode. Record data can consist of:

- ◆ **Diagnostic Data – the diagnostic data associated with a channel (I/O point)**
- ◆ **Logbook Entries – The record of alarms and diagnostics maintained by the PROFINET IO device**
- ◆ **Identification Entries – ID data which describes the device name, family name and other identifying data**
- ◆ **IO Data Objects – Channel Data**

The IP Address of the PROFINET IO device is stored in persistent memory in the device. It can be modified by an IO Supervisor using RDO services. A PROFINET IO device can also be configured to obtain an IP Address automatically from a Bootp or DHCP server.



Alarm vs. Diagnostic Data

Alarms and Diagnostic data in a PROFINET IO system are related and easily confused by the PROFINET IO novice. Diagnostic data is data associated with a channel or I/O point. For example, an analog input can have an over-limit diagnostic or a discrete input can have a short circuit diagnostic. Diagnostic data is always transferred non-cyclically using Record Data communications over the non real time channel. An IO-Supervisor must specifically request the diagnostic data from the device using RDO (Record Data Object) services.

Alarm data is very different. While one type of alarm is an alert that diagnostic data exists on a specific channel, there are many more reasons why a device would signal an alarm. Alarms are signaled when a module or sub-module is plugged or pulled, when the wrong module or sub-module is plugged, when a process limit is reached, when the device state changes, or for many other reasons (See Table 1). Unlike Diagnostic data, Alarm data is transmitted on the Real Time (RT) channel and only to the programmable controller linked to that module. For example, a device with two modules attached to two different programmable controllers issues module alarm data only to the programmable controller exchanging data with the module enabling the alarm.

Table 1 — Alarm Data Types

Diagnostic Appears	Update Alarm	Diagnostic disappears alarm
Process Alarm	Redundancy Alarm	Multicast communications mismatch
Module/Sub-module Pull Alarm	Supervisor Control Alarm	Port Data Change Notification Alarm
Module/Sub-module Plug Alarm	Supervisor Release Alarm	Sync Data Change Notification Alarm
Status Alarm	Wrong Sub-module Alarm	Isochronous Mode Notification Alarm

A unique feature of PROFINET IO alarms is that alarms are not only issued on failures, but are also issued when the alarm condition is cleared. For example, when an alarm is issued on an input shot circuit, another alarm is issued when the short circuit disappears. Almost the same thing happens when an Alarm indicates that Diagnostic data exists for a channel (I/O point). Instead of issuing the alarm for every channel with Diagnostic data, an Alarm is issued when the first Diagnostic data entry is created and when the last Diagnostic data entry is released.

Both Alarm data and Diagnostic data are acknowledged. Diagnostic data is acknowledged simply because the IO-Supervisor receives a response message to the Diagnostic data read request. Alarm data is part of the Real Time (RT) data issued by the device and is specifically acknowledged by the Programmable Controller.

PROFINET GSDML File

The GSDML (Generic Station Description Markup Language) file is an XML (eXtensible Markup Language) file that describes the expected implementation of a PROFINET IO device. Table 3 describes the top level elements of a PROFINET IO GSDML file.

Element	Description
Profile Header	Describes the XML file itself including its version number, the
Profile Body	Details the actual device implementation.

Table 2 — Top Level GSDML Elements



The Profile Body details the device implementation in three main elements.

Element	Description
Device Identity	The device identity identifies the device in terms of the vendor ID assigned by PROFIBUS International, the text description of the name and the text string that describes the device.
Device Function	Specifies the text description of the family name for this device
Application Process	The Application Process describes the application of the device. It describes the number of slots, sub-slots, modules and, sub-modules that the device can have.

Table 3 – Profile Body

The Application Process, composed of two very important elements, details the device implementation in three main elements.

Element	Description
DeviceAccessPointList	The DAP List describes the general configuration of the device. It describes the maximum number of modules, the maximum IO size and the module in Slot zero. Slot zero module data includes its manufacturer ID, the vendor name, its description, version number and the order number.
Module List	The Module list describes all the modules and sub-modules which are valid for this application process. The actual channel data is described for each of the possible sub-modules that may be installed in the device.

Table 5 – Application Process Important Elements

For IO data, the GSDML file describes the structure of the cyclic input and output data transferred between the Programmable Controller and the PROFINET IO device. Any mismatch between the size or structure of the input and output data and the actual internal device structure generates an alarm to the controller.

PROFINET IO Constant Data

Every PROFINET IO device uses a number of constant values. Table 6 defines these values and lists their sources.

ITEM	DESCRIPTION	PERSISTENCE
Vendor ID	Unique value identifying an authorized PROFINET IO Vendor. This value is assigned by PROFIBUS International.	Static Constant
Device ID	Unique value identifying a PROFINET IO device. This value is assigned by the device manufacturer.	Static Constant
Module ID	Unique value identifying a specific module type. This value is assigned by the device manufacturer. When the PROFINET IO device plugs in a module, the module id must agree with the module id specified in the GSDML file.	Static Constant
Sub-module ID	Unique value identifying a specific sub-module type. This value is assigned by the device manufacturer. When the PROFINET IO device plugs in a sub-module, the sub-module id must agree with the sub-module id specified in the GSDML file.	Static Constant
Product Family	A manufacturer specific text string describing the product family.	Static Constant
Station Name	A text string describing the function of the station in the application. The PROFINET IO device is delivered with a default station name. An IO-Supervisor or IO-Controller can send a new station name to the	Stored in Persistent Memory in the PROFINET IO device
IP Address	The IP Address of the device. The IP Address can be changed by an IO-Controller or IO-Supervisor or by a DHCP server. Every PROFINET IO device is shipped with a default IP Address.	Stored in Persistent Memory in the PROFINET IO device

Table 6 – PROFINET IO Data Constants

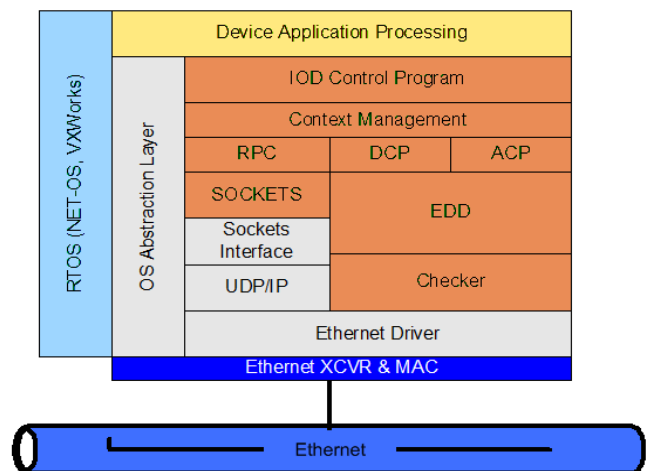
PROFINET RUNTIME STRUCTURE

The PROFINET Runtime Software or “PROFINET Core” is the software made available by PROFIBUS International to developers. This software is only one of a number of components necessary to implement a PROFINET device. See Figure 1 for a graphical representation of the components of a PROFINET device.

USER APPLICATION

The user application “plugs” the modules into the slots on power on, starts the PROFINET task, generates diagnostic data, issues and releases alarms, and maps the IO data between physical device IO and PROFINET IO.

Figure 1 — PROFINET Runtime Software Components



Yellow indicates a user application. Orange indicates the PROFINET IO. Gray indicates an interface module.

RTOS - Real Time OS

The Real Time Operating System performs task management. At the minimum, when PROFINET IO runs in one task, there is a user application task and a data exchange task.

PROFINET IO

The PROFINET IO kernel is the software provided by PROFIBUS International to developers. This software must not be changed. It consists of the following major sub-components:

- ◆ **IO Control Program** – The IOD exchanges IO data with the application
- ◆ **Context Manager** - The CM creates and manages the application and communication relationships with Programmable Controllers and IO-Supervisors
- ◆ **RPC (Remote Procedure Call)** - The subsystem that manages non-cyclic communications messages that arrive over UDP. These messages are the Read/Write services.
- ◆ **DCP** – Discovery Protocol for identifying PROFINET devices
- ◆ **ACP** – Alarm Control for managing Alarm generation and release
- ◆ **Sockets** – Sockets implementation that supports the sending/receiving of UDP messages
- ◆ **EDD** – The Ethernet Device Driver provides services to send and receive Cyclic, Non-cyclic and Non-Real Time (NRT) messages
- ◆ **Checker** – The Checker provides Frame Management and routing services. Frames are routed to the EDD or the Sockets interface for processing.

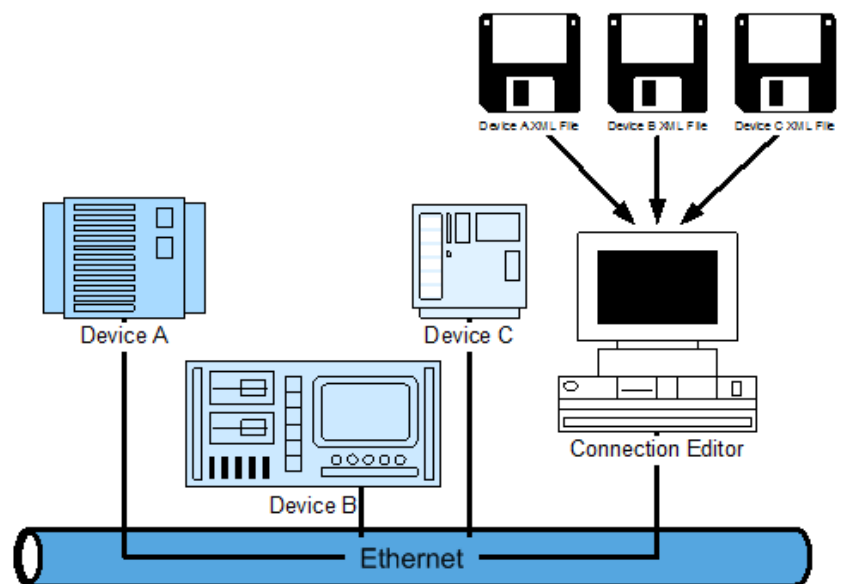
Interface Modules

The Interface modules provide the interface between the PROFINET IO kernel and the RTOS. These are mostly standard RTOS components but some must be modified to support PROFINET IO operation. The OS Abstraction layer is the only one that is not at least partially provided by the RTOS vendor. The abstraction layer maps the OS calls of the PROFINET core to the OS calls of the RTOS and is specific to the RTOS. The abstraction layer is specific to an implementation of the RTOS.

PROFINET Configuration Structure

PROFINET IO devices are typically configured by Siemens Step 7 software. Using Step 7, device descriptions are loaded from the GSDML files. IO configurations are mapped into the IO tables of the Siemens IO Controller and IP addresses are assigned to each PROFINET IO device.

Figure 2 — PROFINET Engineering Model



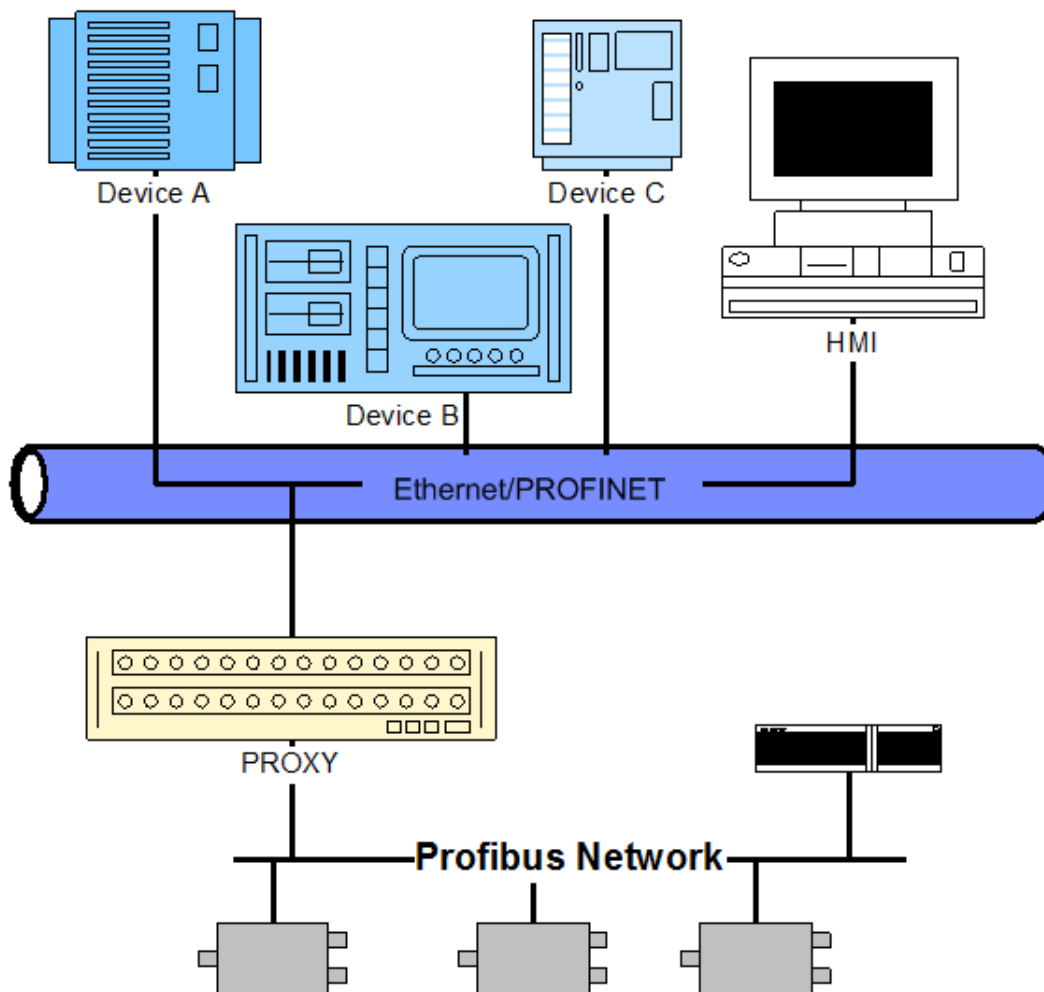
WHERE DOES PROFIBUS FIT IN?

PROFIBUS® is a worldwide standard for transmission of process data between field level devices and programmable controllers. How can I/O data and other information travel between PROFIBUS and PROFINET? At the physical level, these two communication networks are completely incompatible. PROFIBUS is, at its core, an RS485-based communication standard with a maximum baud rate of 12M. At the core of PROFINET is Ethernet, another worldwide standard with baud rates of 100M and beyond.

Are they at all compatible at the protocol level? PROFIBUS is I/O driven. Data is transferred as blocks of inputs and outputs which are interpreted by agreement between the PROFIBUS Master and the Slave. PROFINET IO is also I/O driven with data transferred in blocks of inputs and outputs. In both systems, there is a Master or IO-Controller and one or more IO devices.

Since a PROFIBUS Master device can access all the data of the PROFIBUS network the logical place for a communications gateway is as the Master device of the PROFIBUS network. This is illustrated in the Figure 3.

Figure 3—PROFIBUS Proxy Operation





PROFINET CERTIFICATION

Unlike PROFIBUS, all PROFINET devices must pass certification prior to product launch. The Certification Office of the PROFIBUS User Organization is responsible for establishing test centers where PROFINET devices can be certified.

Certification of a PROFINET device is a three step process. First, the parts of the PROFINET software core modified by the developer and the PROFINET Object Model are tested for compliance with standards. Second, the XML file supplied by the product developer is used with an Engineering tool to test interoperability. In this test, connections are made between the interfaces of the Device Under Test (DUT) and other PROFINET devices. Third, the PROFINET device is connected to a network of other PROFINET devices to determine how well the device works on a loaded network.

HOW TO GET STARTED

The PROFINET Runtime Core software can be downloaded free of charge (membership required) from PROFIBUS International as a zip file. Porting PROFINET to a specific RTOS requires the developer to port each PROFINET component. The components to port include the RPC, Context Manager, Sockets Interface, Discovery Protocol, Alarm Handler Interface and the Ethernet Device Driver. The PROFINET Integration manual provides individual instructions for the porting of each of these components.

Prior to starting this effort the developer should have a background in the Microsoft RPC, COM, DCOM and C++. An extensive background in the specific operation of your RTOS, the operation of your TCP/IP stack, and the fundamentals of your processor is also important.

PROFINET implementation is not without challenges. Porting of your PROFINET core to your embedded RTOS is a significant challenge. The PROFINET core is designed for windows. It is a major effort to port the PROFINET core to a non-windows RTOS. Some of the more challenging aspects of this integration effort include mapping the Windows memory management services, real time clock functions, and DCOM to the embedded system. The target device requires significant resources. The PROFINET core contains a large number of source files with significant code and memory requirements including non-volatile memory.

FOR MORE INFORMATION

- ◆ PROFIBUS International, *PROFINET Architecture Description and Specification, Version 1.0, August 2001*
- ◆ PROFIBUS International, *PROFINET Implementation Guide, Version 1.2, July 2002*
- ◆ Open Software Foundation (OSF), www.opensoftwarefoundation.com
- ◆ Dr. Richard Grimes, *“DCOM Programming”*, 1997 Wrox Press Ltd.



Appendix 1: PROFINET Terms

PROFINET like many other networking systems has a set of unique terminology. Table 1 contains a few of the terms commonly used when discussing PROFINET.

Term	Abbreviation	Definition
Application Relationship	<i>AR</i>	The relationship between a PROFINET I/O Controller and an I/O device. A PROFINET I/O device can support more than one Application Relationship.
Application Process	<i>AP</i>	The application process running in the PROFINET I/O device. PROFINET supports a default Application Processes and additional, profile specific application processes.
Channel		A single I/O point. A Channel can be discrete or analog.
Sub-slot		A group of one or more channels. Sub-slots can be real or virtual.
Slot		A group of one or more Sub-slots. Slots can be real or virtual.
Module		Modules are user defined components that plug into slots. Modules can be real or virtual.
Sub-module		A component of a module that is plugged into a sub-slot. A sub-module is real or virtual.
Cyclic Communications		Scheduled, repetitive communications. IO data and alarm transfers are cyclic.
Non-cyclic Communications		Unscheduled, on demand communications. Diagnostic messages from an IO Supervisor to an IO Device are Acyclic.
Generic Station Description	<i>GSDML</i>	The file containing the XML description of the PROFINET IO de-
Provider Status		The Status an IO device provides to an IO Controller with the data transferred to the Controller.
Consumer Status		The Status an IO device provides to an IO Controller for the data it consumes from IO Controller.
Real Time	<i>RT</i>	The Real Time PROFINET IO Channel. IO and Alarm Data are transferred over the RT Channel.
Non Real Time	<i>NRT</i>	The non Real Time PROFINET IO Channel. Configuration and diagnostic messages are transferred over the NRT Channel.
Communication Relationship	<i>CR</i>	A virtual communication channel within an AR
Frame ID		The two byte field in the Ethernet frame which defines the type of PROFINET IO message.
Proxy		A device which maps non PROFINET IO data to PROFINET.
Discovery Control Protocol	<i>DCP</i>	A communications protocol with PROFINET IO that allows an IO Controller or Supervisor to find every PROFINET IO device on a subnet.