SIEMENS

SIMATIC

PROFINET IO
From PROFIBUS DP to PROFINET IO

Programming Manual

The following supplement is part of this documentation:

No.	Product Information	Drawing number	Edition
1	Information on new and revised diagnostics data records	A5E01648459-01	07/2008

This manual is part of the documentation package with order number: 6ES7398-8FA10-8BA0

10/2006
A5E00298268-03

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

Danger

indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 48 48
90437 NÜRNBERG
GERMANY

Order No.: A5E00298268-03
℗ 02/2007

Preface

Purpose of this manual

This manual gives you an overview of the differences between PROFIBUS DP and PROFINET IO with regard to migration. It supports you in the installation, commissioning and ongoing operation of a PROFINET IO system.

The manual describes the procedure of programming diagnostics for IO devices.

The target readership of this manual includes application programmers and technical personnel working in the field of configuring, commissioning and servicing automation systems.

Basic knowledge required

To understand this manual, you will need to know the following:

- General knowledge in the field of automation technology
- Familiarity with computers or equipment similar to PCs (such as programming devices) running a Windows operating systems.
- Knowledge of STEP 7. Refer to the *Programming with STEP 7 V5.4* manual.
- Sound knowledge of PROFINET IO and PROFIBUS DP communication functions.
- Sound knowledge of SIMATIC distributed I/O

You should also be familiar with the PROFINET system description.

Scope

This documentation forms the basis of the documentation for all PROFINET-related products. The documentation for the various PROFINET products is based on this documentation.

Position in the IT world

Additional manuals required depending on the application:

- The *PROFINET IO Getting Started Collection* manual
- The *Programming with STEP 7 V5.4 SP1* manual
- The *PROFINET system description* manual
- The *Profinet IO diagnostics in the user program* application description

Guide

Topics covered in this manual:

- Comparison of PROFIBUS DP and PROFINET IO:
- Data records for diagnostics and status requests
- Examples of diagnostics in the user program
- Annexes

The glossary explains important terminology. The index contains important keywords which allow quick access to relevant text passages.

Recycling and disposal

The devices described in this documentation can be recycled on account of their low pollutant

content. For environmentally-sustainable recycling and disposal of your

old devices, contact a company which is certified for the disposal of electrical scrap.

Changes compared to the previous version

This manual includes a description of new technologies and enhanced functionality of the SIMATIC device family.

Readership

This manual is aimed primarily at groups engaged in the design and engineering of networked automation solutions with SIMATIC products:

- Decision-makers
- Planners
- Project engineers

Commissioning engineers and field service personnel will also benefit from this manual.

Additional support

Contact your local Siemens partner if you have any questions about the use of products described in this manual and cannot find the right answers.

- Find your contact partner at:
  http://www.siemens.com/automation/partner
- The guide to our technical documentation of SIMATIC products and systems is available at:
  http://www.siemens.de/simatic-doku
- The online catalog and ordering system are available at:
  http://mall.automation.siemens.com/

Training Centers

Siemens offers a range of courses to help you get started with the SIMATIC S7 automation system. Contact your regional Training Center, or the central Training Center in D-90327 Nuremberg, Germany.

- Phone: +49 (911) 895-3200
- Internet: http://www.sitrain.com

Technical Support

Use the Support Request web form to contact Technical Support for all A&D products on the

- Internet at: http://www.siemens.de/automation/support-request
- Phone: + 49 180 5050 222
- Fax: + 49 180 5050 223

For further information about Siemens Technical Support, visit our website at http://www.siemens.de/automation/service

Service & Support on the Internet

In addition to our pool of documentation, we offer you a comprehensive knowledge base on the Internet.

http://www.siemens.com/automation/service&support

The Service & Support pages provide:

- The Newsletter which provides the latest information on your products.
- The documents you need using our search engine at Service & Support.
- A world-wide forum where users and specialists share their experience.
- Your local Automation & Drives representative.
- Information about on-site service, repairs and spare parts. Lots more is available on our "Services" pages.

Table of contents

Tables

Guide to the PROFINET documentation

1

Overview

The diagram below shows an overview of the PROFINET documentation pool.

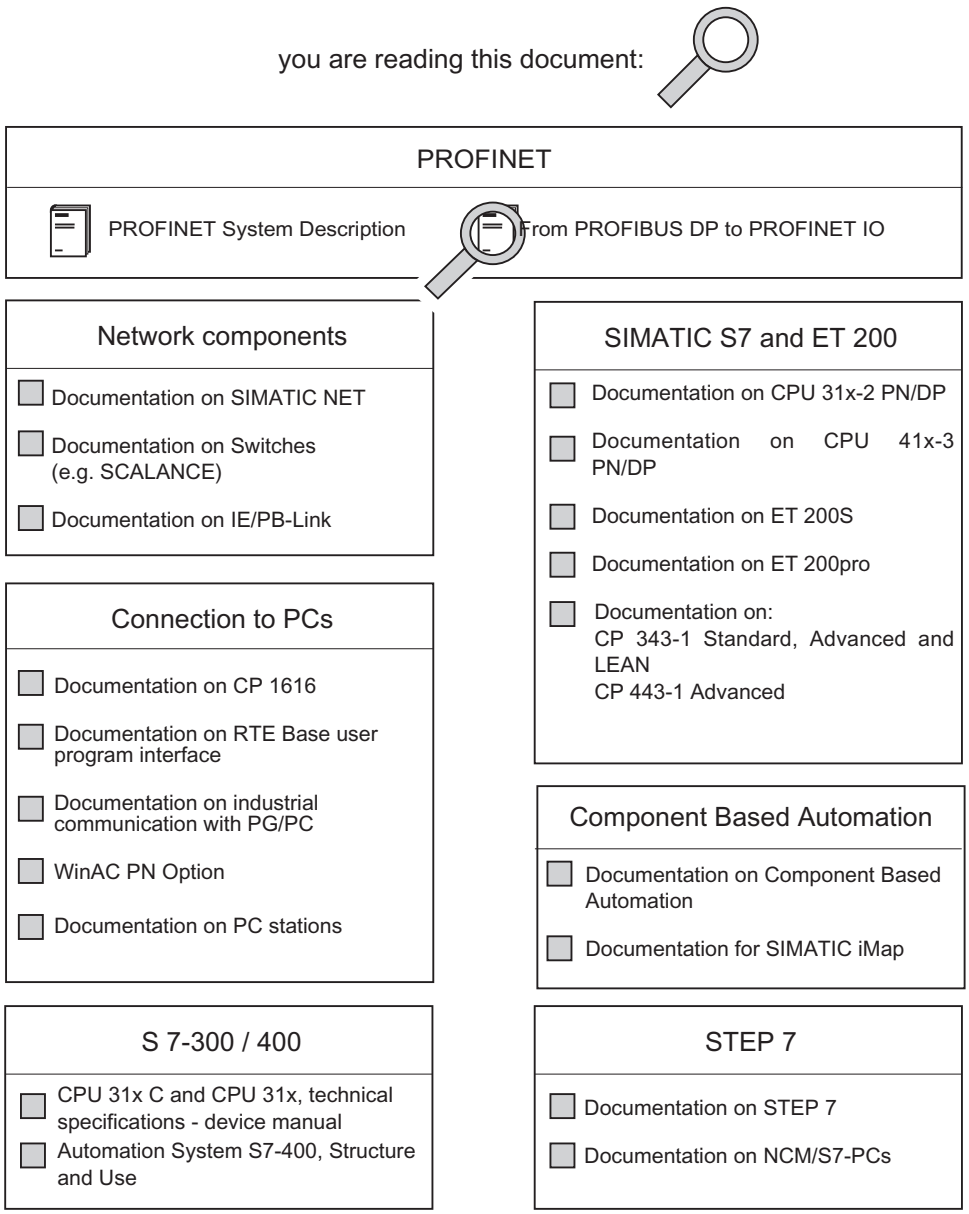

Figure 1-1 Overview of the documentation pool

Information on PROFIBUS and PROFINET

Members of the PROFIBUS Nutzerorganisation e. V. (PNO) (PROFINET user organization) include more than 1200 manufacturers and users who are focussed on the standardization of PROFIBUS and PROFINET communication systems.

Additional information about PROFIBUS DP and PROFINET IO is available on the Internet at **http://www.profibus.com**. Installation guidelines (Installation Guideline PROFINET) are available at: **http://www.profibus.com/libraries.html**

PROFINET IO and PROFIBUS DP 2

Content of this section

This chapter describes the major differences between PROFINET IO and PROFIBUS DP.

2.1 Comparison of PROFINET IO and PROFIBUS DP

Adaptations to be made in the user program

Check in particular the functionality outlined below in an application program for PROFIBUS devices which is also used by devices which communicate over PROFINET.

- If using blocks which are not supported in PROFINET IO:
  Refer to the section *Blocks in PROFINET IO and PROFIBUS DP.*
- If using system status lists which are not supported in PROFINET IO:
  Refer to the section *System Status Lists in PROFINET IO and PROFIBUS DP.*
- The physical addresses in PROFINET IO and PROFIBUS DP do not match:
  Refer to section *Diagnostics using SFB54 in OB82*.

Revise the user program if your answer is yes.

It became necessary to implement certain new blocks for PROFINET IO, as PROFINET is capable of handling larger quantity frameworks.

The new blocks and systems status lists replace the previous versions and are generally compatible. These objects can be used in PROFINET IO and PROFIBUS DP.

You can basically implement previously used blocks and systems status lists when operating only with PROFIBUS DP. However, you are strongly advised to migrate to the "new" system and standard functions.

Comparison of the transmission technology of PROFINET IO and PROFIBUS DP

Table 2-1 Comparison of the transmission technology of PROFINET IO and PROFIBUS DP

Feature	PROFINET IO	PROFIBUS DP
Cable transmission technology	Industrial Ethernet over copper or fiber-optic cable.	PROFIBUS over copper or fiber-optic cable.
Wireless transmission technology	Industrial WLAN supports wireless transmission.	Infrared transmission is supported.

Comparison of PROFINET IO and PROFIBUS DP topology

Table 2-2 Comparison of PROFINET IO and PROFIBUS DP topology

Feature	PROFINET IO	PROFIBUS DP
Topology	Standard: Star and tree topology Line and ring topologies are supported	Standard: Line Tree and ring topologies are supported
Implementation in star topology	Only one network node per switch port	PROFIBUS DP is normally looped through from one node to the next. For information about tree and ring topologies, refer to the *PROFIBUS Networks* manual.
Implementation in tree topology	The switches are interconnected.	
Implementation in line topology	PROFINET devices are interconnected using integrated switches.	
Implementation in ring topology	Both ends in a line are joined by a redundancy manager to form a ring topology.	

Addressing I/O devices/DP slaves

Table 2-3 Addressing I/O devices/DP slaves

Feature	PROFINET IO	PROFIBUS DP
Addressing	Assigning IP addresses and device names to IO devices in STEP 7. Transferring the device name to Micro Memory Card by means of STEP 7. Assignment of IP addresses to IO devices by the IO controller. Assignment of IP addresses to switches or CPs by means of Primary Setup Tool (PST). Certain switches feature an integrated web-based management tool which can be accessed with standard browsers. This tool can also be used to assign IP addresses.	PROFIBUS address coding by means of DIL switch or configuration in STEP 7.

GSD file

Table 2-4 Import of device data in STEP 7

Feature	PROFINET IO	PROFIBUS DP
Import of device data in STEP 7	GSD file in XML format	GSD file in ASCII format

The GSD file for PROFINET IO is imported as in PROFIBUS DP.

For more information on GSD files, refer to the STEP 7 Online Help and to the *PROFINET System Description*, contribution ID 19292127.

2.2 Representation in STEP 7/NCM PC

STEP 7/NCM PC versions which support migration to PROFINET IO

STEP 7 V5.3 Service Pack 1 or higher is required to integrate PROFINET devices in the SIMATIC system.

Comparison of PROFINET IO and PROFIBUS DP in STEP 7 / NCM PC

The PROFINET IO and PROFIBUS DP configuration is basically the same in STEP 7 and NCM PC, with the exception of a few names. The table below shows the differences in names.

Table 2-5 Comparison of the representation of PROFINET IO and PROFIBUS DP in STEP 7 and NCM PC

Feature	PROFINET IO	PROFIBUS DP
Subnet name	Ethernet	PROFIBUS
Subsystem name	IO system	DP master system
Name of the master device	IO controller	DP master
Name of the slave device	IO device	DP slave
Hardware catalog	PROFINET IO	PROFIBUS DP
Numbering	Device number	PROFIBUS address (corresponds with the station number)
Operating parameters, diagnostics address	Listed in the object properties of the interface module in slot 0	Listed in the object properties of the station System parameters which are not available at a module/submodule are inactive.

NCM PC

For information on the basic properties of NCM PC, refer to the *PROFINET System Description.*

Blocks in PROFINET IO and PROFIBUS DP

3

Content of this section

This section covers:

- Blocks designed for use with PROFINET
- Blocks designed for use with PROFIBUS DP
- Blocks designed for use with PROFINET IO and PROFIBUS DP

Compatibility of the new blocks

New blocks were implemented for PROFINET IO, as PROFINET is capable of handling larger quantity frameworks. The new blocks are also used for PROFIBUS.

Comparison of the system and standard functions of PROFINET IO and PROFIBUS DP

For CPUs with integrated PROFINET interface, the table below provides an overview of:

- System and standard functions for SIMATIC which you will have to upgrade for migration from PROFIBUS DP to PROFINET IO.
- New system and standard functions

Table 3-1 System and standard functions which are new or have to be replaced

Blocks	PROFINET IO	PROFIBUS DP
SFC12 (deactivation and activation of DP slaves/IO devices)	Yes CPU S7-300: firmware V2.4.0 or later S7-400: firmware V5.0 or later	Yes
SFC13 (reading diagnostics data from a DP slave)	No Replaced by: • Event-driven: SFB54 • Status-driven: SFB52	Yes
SFC 58/59 (write/read record in I/O)	No Replaced by: SFB53/52	Yes Should already have been replaced by SFB53/52 in DPV1
SFB52/53 (read/write record)	Yes	Yes
SFB54 (evaluate interrupt)	Yes	Yes
SFC102 (read predefined parameters - S7-300 CPU only)	No Replaced by: SFB81	Yes for S7-300 SFC54 for S7-400

Blocks	PROFINET IO	PROFIBUS DP
SFB81 (read predefined parameters)	Yes	Yes
SFC5 (query start address of a module)	No (replaced with: SFC 70)	Yes
SFC70 (query start address of a module)	Yes	Yes
SFC49 (query the slot at a logical address)	No Replaced by: SFC71	Yes
SFC71 (query the slot at a logical address)	Yes	Yes
SFC105 (status of dynamically-assigned ALARM_Dx system resources)	Yes (firmware V2.5 or later)	Yes (firmware V2.5 or later)
SFC106 (enable dynamically-assigned system resources)	Yes (firmware V2.5 or later)	Yes (firmware V2.5 or later)
SFC107 (generate acknowledgeable message with associated value)	Yes (firmware V2.5 or later)	Yes (firmware V2.5 or later)
SFC108 (generate non-acknowledgeable message with associated value)	Yes (firmware V2.5 or later)	Yes (firmware V2.5 or later)

The table below provides an overview of SIMATIC system and standard functions which must be emulated by other functions when migrating from PROFIBUS DP to PROFINET IO.

Table 3-2 System and standard functions in PROFIBUS DP which can be emulated in PROFINET IO functions

Blocks	PROFINET IO	PROFIBUS DP
SFC54 (read default parameters - S7-400 CPU only)	No Replaced by: SFB81	Yes, for CPU S7-400
SFC55 (write dynamic parameters)	No Emulate using SFB53	Yes
SFC56 (write predefined parameters)	No Emulate using SFB81 and SFB53	Yes
SFC57 (assign module parameters)	No Emulate using SFB81 and SFB53	Yes

SIMATIC system and standard functions not supported in PROFINET IO:

- SFC7 (trigger process interrupt at DP master)
- SFC11 (synchronize groups of DP slaves)
- SFC72 (read data from communication partner within local S7 station)
- SFC73 (write data to communication partner within local S7 station)
- SFC74 (cancel communication with partner within local S7 station)
- SFC103 (determine the bus topology in a DP master system)

Comparison of the organization blocks of PROFINET IO and PROFIBUS DP

The table below shows the changes to OB83 and OB86 in PROFINET IO compared to PROFIBUS DP.

Table 3-3 OBs in PROFINET IO and PROFIBUS DP

Blocks	PROFINET IO	PROFIBUS DP
OB83 (hot swapping of modules/submodules)	Also supported on S7-300, new error information	S7-300 does **not** support this function Slaves integrated via the GSD file report the removal/insertion of modules/submodules during operation in the form of a diagnostics interrupt and thus via OB82. S7 slaves report a station failure and call OB86 when an insertion/removal interrupt is generated.
OB86 (rack failure)	New error information	Unchanged

Detailed information

For detailed information about the blocks, refer to the *System Software for S7-300/400 System and Standard Functions* manual.

See also

CP 343-1 (Page 115)

CP 443-1 Advanced (Page 116)

System Status Lists in PROFINET IO and PROFIBUS DP

4

Content of this section

This section covers:

- The system status lists designed for PROFINET IO
- The system status lists designed for PROFIBUS DP
- The system status lists designed for PROFINET IO and PROFIBUS DP

Introduction

The CPU of SIMATIC modules can provide specific information. The CPU saves this information to the "system status list" (SSL).

The SSL describes the current status of the automation system. It provides an overview of:

- the configuration
- the current parameter settings
- the current states
- processes in the CPU and the modules assigned to it

The data in the system status list is read-only. The SSL is a virtual list which is only created on request.

You can use the SSL to obtain the following information about the PROFINET IO system:

- System data
- Module status information in the CPU
- Diagnostics data for a module
- Diagnostics buffer

Compatibility of the new SSLs

New system status lists were implemented for PROFINET IO, as PROFINET supports larger volumes of project data.

You should also use these new SSLs with PROFIBUS.

You can use an existing PROFIBUS SSL as usual if this list is also supported in PROFINET. If you try to use an SSL which is not supported in PROFINET, the program returns an error code in RET_VAL (8083: Incorrect or illegal index).

Comparison of the SSLs of PROFINET IO and PROFIBUS DP

Table 4-1 Comparison of the SSLs of PROFINET IO and PROFIBUS DP

SSL ID	PROFINET IO	PROFIBUS DP	Validity
W#16#0591	Yes (parameter adr1 changed)	Yes	Module status information for the interfaces of a module/submodule
W#16#0A91	Yes (parameter adr1 changed)	Yes	Status information of all subsystems and master systems (S7-300 without CPU 318-2 DP only)
W#16#0C91	Yes (parameter adr1/adr2 and target/actual type ID changed)	Yes	Module status information of a module/submodule in a central rack or interconnected with an integrated DP or PN interface module using the logical module address.
W#16#4C91	No	Yes	Not for S7-300 Module status information of a module interconnected with an external DP or PN interface module using the start address
W#16#0D91	Yes (parameter adr1 changed)	Yes	Module status information of all modules in the specified rack/station
W#16#0696	Yes	No	Module status information of all submodules of a module using the logical address of the module; not possible for submodule 0 (= module)
W#16#0C96	Yes	Yes	Module status information of a submodule using the logical address of this submodule
W#16#xy92	No (replaced with: SSL ID W#16#0x94)	Yes	Rack/station status information Also replace this SSL in PROFIBUS DP with the SSL with ID W#16#xy94.
W#16#0x94	Yes	Yes	Target state of the stations or central racks
W#16#x294	Yes	Yes	Actual state of the stations or central racks
W#16#0x694	Yes	Yes	All stations of an IO subsystem or all central racks in error state
W#16#0x794	Yes	No	Error/maintenance state of the stations or central racks

Additional information about SSLs

For detailed information about the individual SSLs, refer to the *System Software for S7-300/400 System and Standard Functions* manual and the STEP 7 V5.4 SP1 online help.

Records with PROFINET IO

5

Content of this section

This section covers:

- The most important differences between PROFINET IO and PROFIBUS DP in terms of diagnostics
- How the diagnostics mechanism works in PROFINET IO
- Structure of a diagnostics and configuration record in PROFINET IO.

Additional information

For further information about diagnostics, refer to the STEP 7 Online Help.

5.1 Introduction

5.1.1 Overview of diagnostics and configuration records

Consistent diagnostics concept

PROFINET IO supports you with a consistent diagnostics concept.

The next section describes the basic features of the diagnostics concept.

Diagnostics mechanism

The IO device outputs a diagnostics interrupt to the IO controller when it detects faults such as wire break. This interrupt calls a corresponding OB in the user program (diagnostics interrupt OB82), in order to generate a defined (programmed) response to the fault.

The IO controller automatically sets the parameters and configures a replacement device or module. Cyclic exchange of user data is restored in the next step.

Diagnostics records in PROFINET IO

There are two different types of diagnostics record:

1. Channel diagnostics records

   Channel diagnostics records are generated if a channel is in an error state and / or has triggered an interrupt.
   A diagnostics record of length 0 is returned if there is no fault.

2. Vendor-specific diagnostics records

   The structure and size of vendor-specific diagnostics records depend on the vendor's settings.

   For information about vendor-specific diagnostics records, refer to the appropriate device manual.

Profile and structure of the diagnostics and configuration records

A PROFINET IO device consists of one or more "logical devices". These devices in turn contain one or more APIs (Application Process Identifiers) and at least API 0. The PROFINET IO profile (PROFIdrive, for example) is coded using the API.

Each PROFINET IO device supports at least one **A**pplication **P**rocess **I**dentifier (API).

The diagnostics records (W#16#800A, for example) can differ in terms of their structure. This difference is characterized by the BlockVersion. BlockVersion 0101 of record W#16#X00A, for example, was upgraded with the API number in order to enable diagnostics for IO devices with several APIs.

By contrast to PROFIBUS DP, the API is used in PROFINET IO as a profile ID and is a parameter which identifies the profile. Examples for various application scenarios:

Table 5-1 Application profiles

Field of application	Profile	API
Drive technology	PROFIdrive	W#16#3A00 - W#16#3AFF
Safety technology	PROFIsafe	W#16#3A00 - W#16#3AFF
Conveyor systems	Intelligent pumps	W#16#5D00 - W#16#5DFF

Requirements

The diagnostics information is only generated for configured modules / submodules / channels.

List of diagnostics and configuration records in PROFINET IO

The table below lists important diagnostics records in PROFINET IO.

The specified record sizes apply to at least one faulty channel.

Table 5-2 Diagnostics records in PROFINET IO

Record number	Content and meaning	Size in bytes
W#16#800A	The record returns - channel diagnostics data and / or - extended channel diagnostics data for a submodule slot Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#800B	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a submodule slot Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#800C	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a submodule slot Note: This record is only available if maintenance is requested or demanded, and when a fault is detected; refer to section 5.5.7 The record may also contain status information for an IE/PB link.	0 - 4176
W#16#8010	The record returns - channel diagnostics data and / or - extended channel diagnostics data for a submodule slot Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176

Record number	Content and meaning	Size in bytes
W#16#8011	The record returns - channel diagnostics data and / or - extended channel diagnostics data for a submodule slot Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176
W#16#8012	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a submodule slot Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#8013	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a submodule slot Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176
W#16#C00A	The record returns - channel diagnostics data and / or - extended channel diagnostics data for a module slot Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#C00B	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a module slot Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176

Record number	Content and meaning	Size in bytes
W#16#C00C	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a module slot Note: This record is only available if maintenance is requested or demanded, and when a fault is detected; refer to section 5.5.7 The record may also contain status information for an IE/PB link.	0 - 4176
W#16#C010	The record returns - channel diagnostics data and / or - extended channel diagnostics data for a module slot Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#C011	The record returns - channel diagnostics data and / or - extended channel diagnostics data for a module slot Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176
W#16#C012	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a module slot Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#C013	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a module slot Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176

Record number	Content and meaning	Size in bytes
W#16#E00A	The record returns - channel diagnostics data and / or - extended channel diagnostics data for an AR Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#E00B	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an AR Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#E00C	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an AR Note: This record is only available if maintenance is requested or demanded, and when a fault is detected; refer to section 5.5.7 The record may also contain status information for an IE/PB link.	0 - 4176
W#16#E010	The record returns - channel diagnostics data and / or - extended channel diagnostics data for an AR Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#E011	The record returns - channel diagnostics data and / or - extended channel diagnostics data for an AR Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176

Record number	Content and meaning	Size in bytes
W#16#E012	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an AR Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#E013	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an AR Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176
W#16#F00A	The record returns - channel diagnostics data and / or - extended channel diagnostics data for an API Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#F00B	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an API Note: This record is only available if a fault is detected; see section 5.5.7	0 - 4176
W#16#F00C	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an API Note: This record is only available if maintenance is requested or demanded, and when a fault is detected; refer to section 5.5.7 The record may also contain status information for an IE/PB link.	0 - 4176

Record number	Content and meaning	Size in bytes
W#16#F010	The record returns - channel diagnostics data and / or - extended channel diagnostics data for an API Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#F011	The record returns - channel diagnostics data and / or - extended channel diagnostics data for an API Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176
W#16#F012	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an API Note: This record is only available if maintenance is requested; see section 5.5.7	0 - 4176
W#16#F013	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for an API Note: This record is only available if maintenance is demanded; see section 5.5.7	0 - 4176
W#16#F80C	The record returns - channel diagnostics data and / or - extended channel diagnostics data and / or - vendor-specific diagnostics data for a device Note: This record is only available if maintenance is requested or demanded, and when a fault is detected; refer to section 5.5.7 The record may also contain status information for an IE/PB link.	0 - 4176

The list below shows the important configuration records in PROFINET IO.

Table 5-3 Configuration data records in PROFINET IO

Record number	Content and meaning	Size in bytes
W#16#8000	Target configuration at subslot level	22 - 4176
W#16#C000	Target configuration at subslot level	22 - 4176
W#16#E000	Target configuration at AR level	22 - 4176
W#16#8001	Actual configuration at subslot level	0 - 4176
W#16#C001	Actual configuration at subslot level	0 - 4176
W#16#E001	Actual configuration at AR level	0 - 4176
W#16#E002	Deviation from the target configuration of the IO device	0 - 4176
W#16#F000	Actual configuration at API level	0 - 4176

Structure of other records

The structure of all the records is defined in the *PROFINET IO - Application Layer Service Definition - Application Layer Protocol Specification* standard. Members of the PROFIBUS User Organisation can download this standard from http://www.profibus.com.

A list of vendor IDs is included in the management information of OB82 for PROFINET IO (see section 6.5.3) and is also available at http://www.profibus.com.

Additional information

For further information about diagnostics, refer to the *PROFINET system description* system manual.

5.1.2 Overview of the additional records in PROFINET IO

Overview of the records relevant to PROFINET IO

Table 5-4 Records for reading I/O handling in PROFINET IO

Record number	Content and meaning	Size in bytes
W#16#801E	This record returns the substitute values for a submodule.	0 - 4176
W#16#8028	This record returns the current input data of the submodule.	0 - 4176
W#16#8029	This record returns the current output data of the submodule.	0 - 4176

Table 5-5 Records which return the status of the PROFINET interfaces

Record number	Content and meaning	Size in bytes
W#16#802A	This record returns the current port settings.	0 - 4176
W#16#802B	This record returns the configured port settings.	0 - 4176
W#16#802F	This record returns the configured port settings.	0 - 4176
W#16#8060	This record returns the current settings of the optical port.	0 - 4176
W#16#8061	This record returns the configured settings of the optical port.	0 - 4176
W#16#8062	This record returns the configured settings of the optical port.	0 - 4176
W#16#8070	This record returns the configured settings of the PROFINET interface.	0 - 4176
W#16#F831	This record returns the group record for the configured settings of the PROFINET interface and its ports (IRT parameter settings only).	0 - 4176
W#16#F841	This record returns the group record for the current settings of the PROFINET interface and its ports.	0 - 4176
W#16#F842	This record returns the group record for the configured settings of the PROFINET interface and its ports.	0 - 4176

Table 5-6 Records for reading/writing I&M data in PROFINET IO

Record number	Content and meaning	Size in bytes
W#16#AFF0	This record returns I&M 0 data.	0 - 4176
W#16#AFF1	This record returns I&M 1 data.	0 - 4176
W#16#AFF2	This record returns I&M 2 data.	0 - 4176
W#16#AFF3	This record returns I&M 3 data.	0 - 4176
W#16#F840	This record returns a list of submodules which send different I&M 0 data.	0 - 4176

Table 5-7 Records for reading/writing protocol parameters in PROFINET IO

Record number	Content and meaning	Size in bytes
W#16#F821	This record returns all APIs supported by a PROFINET IO device.	0 - 4176
W#16#F830	This record returns a list of internal error events, for example, the cause of the cancellation of a communication relationship.	0 - 4176

Additional information

For detailed information about the records, refer to the V2.1 of the PROFINET specification "Application Layer services for decentralized periphery and distributed automation" and "Application Layer protocol for decentralized periphery and distributed automation".

5.2 Device model with PROFINET IO

5.2.1 Device model of an IO device

Introduction

The PROFINET IO device model describes the structure of modular and compact field devices. It builds upon the basic features of PROFIBUS DP.

The definition of submodules and APIs has been added to the device model in order to increase the flexibility of an IO device.

Modules / submodules / channels

A PROFINET IO device has a modular structure similar to a PROFIBUS DP slave.

The modules are inserted into slots and the submodules are inserted into subslots. The modules / submodules have channels which are used to read and output process signals.

The diagram below shows the setup.

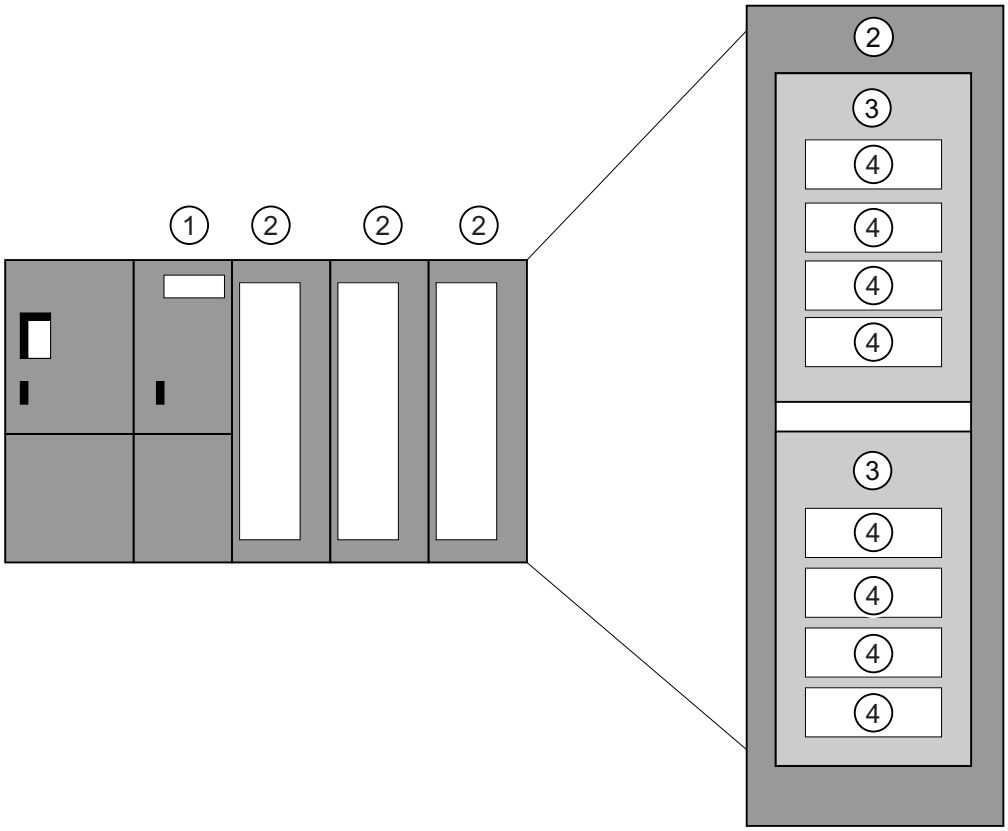

Figure 5-1 Structure of a PROFINET device

Digit	Description
①	Slot with interface circuit
②	Slot with module
③	Subslot with submodule
④	Channel

It is possible to divide a slot into several subslots into which the submodules are inserted.

5.2.2 Diagnostics levels with PROFINET IO

Concept

The IO device sends all error messages that occur to the IO controller. The scope and volume of diagnostics information varies according to the level of diagnostics data evaluation.

Diagnostics levels

You can evaluate diagnostics data at different levels.

The number and type of channels is selected at the diagnostics levels.

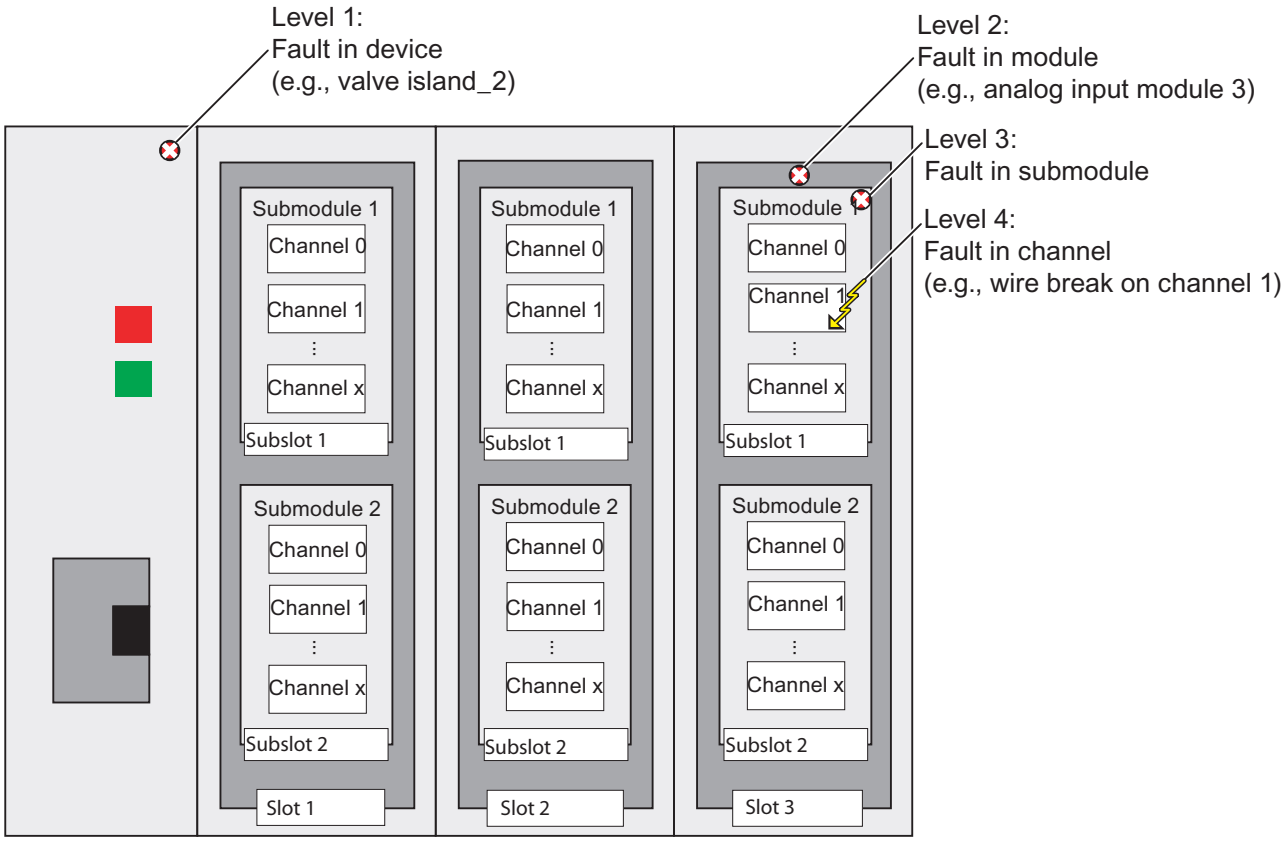

Figure 5-2 Diagnostics levels in PROFINET IO

Table 5-8 Diagnostics levels

Level	Fault location
1	Fault in device, valve block 2
2	Fault in module, analog module 3
3	Fault in submodule
4	Channel fault, wire break at channel 1

Addressing level and records

Diagnostics and configuration data is evaluated at the following addressing levels:

- AR (Application Relation)
- API (Application Process Identifier)
- Slot
- Subslot

A group of diagnostics and configuration records are available at each addressing level. The individual groups of records are distinguished by the first letter of the record number.

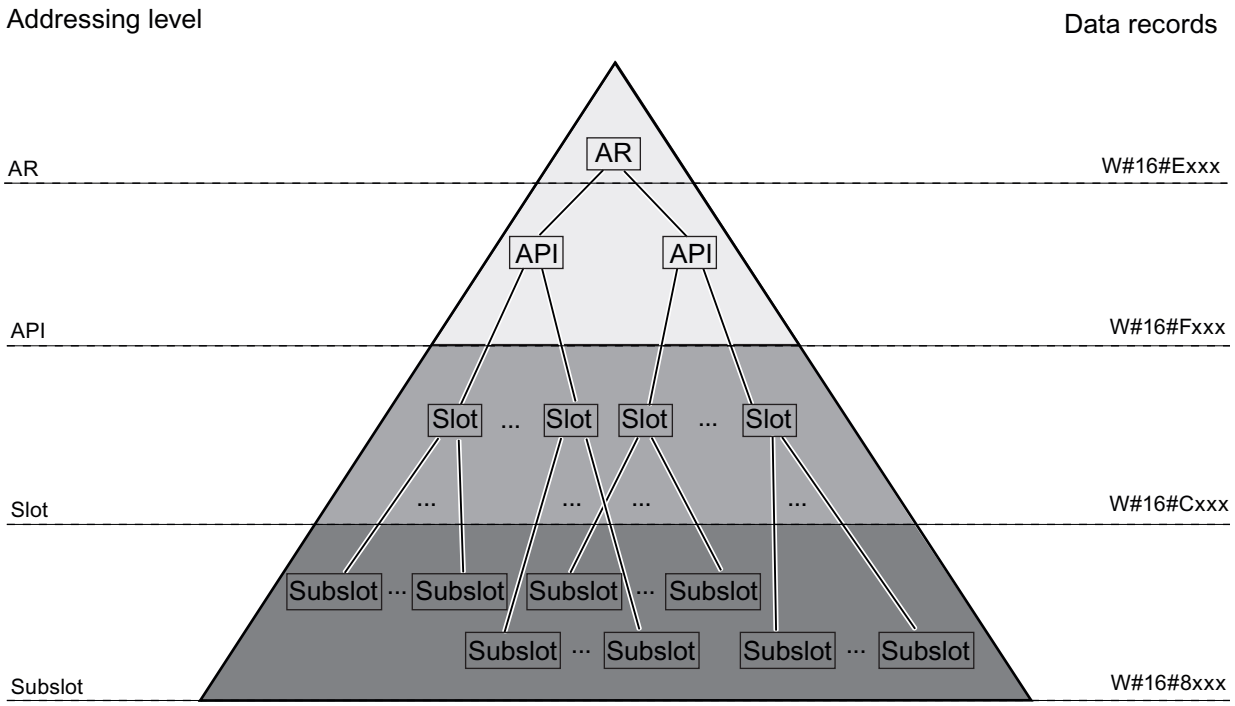

Figure 5-3 Diagnostics levels

The information for each IO device (addressing level AR), module (addressing level slot) or submodule (addressing level subslot) is always transferred in separate diagnostics or configuration records. The record returns diagnostics or configuration data for one or more subslots, slots and APIs, depending on the addressing level.

5.3 Structure of Diagnostics Data Records

5.3.1 Function and selection of a diagnostics data record

Introduction

The basic **structure** is the same for each of the following diagnostics records:

- W#16#800A, W#16#800B, W#16#800C, W#16#8010, W#16#8011, W#16#8012, W#16#8013,
- W#16#C00A, W#16#C00B, W#16#C00C, W#16#C010, W#16#C011, W#16#C012, W#16#C013,
- W#16#E00A, W#16#E00B, W#16#E00C, W#16#E010, W#16#E011, W#16#E012, W#16#E013,
- W#16#F00A, W#16#F00B, W#16#F00C, W#16#F010, W#16#F011, W#16#F012, W#16#F013.

The **content**and **size** of a diagnostics record may vary depending on the type of diagnostics (see "User structure identifier").

Record identifier

You can select a suitable record for specific diagnostics in an application program with reference to the record name. The structure is described below.

This relates to the **first digit** and **last two digits** of the record number:

- **First digit**:

  The first digit of the name of a diagnostics record (for example, W#16#**8**00A) refers to the **addressing level** (AR, API, slot, subslot). You can request diagnostics information at one of these addressing levels.

- **Last two digits**:

  The last two digits of the name of a diagnostics record (W#16#C0**12**) in combination with the **U**ser **S**tructure **I**dentifier (**USI**) identify the type of diagnostics record, such as:

  – Channel diagnostics
  – Extended channel diagnostics
  – Vendor-specific diagnostics
  – MaintenanceRequest
  – MaintenanceDemanded

Addressing level

There are suitable groups of diagnostics records available for each addressing level - AR, API, slot, subslot - at which diagnostics data is requested.

The first letter of the record number identifies the group (W#16#**E**0XX, W#16#**F**0XX, W#16#**C**0XX or W#16#**8**0XX).

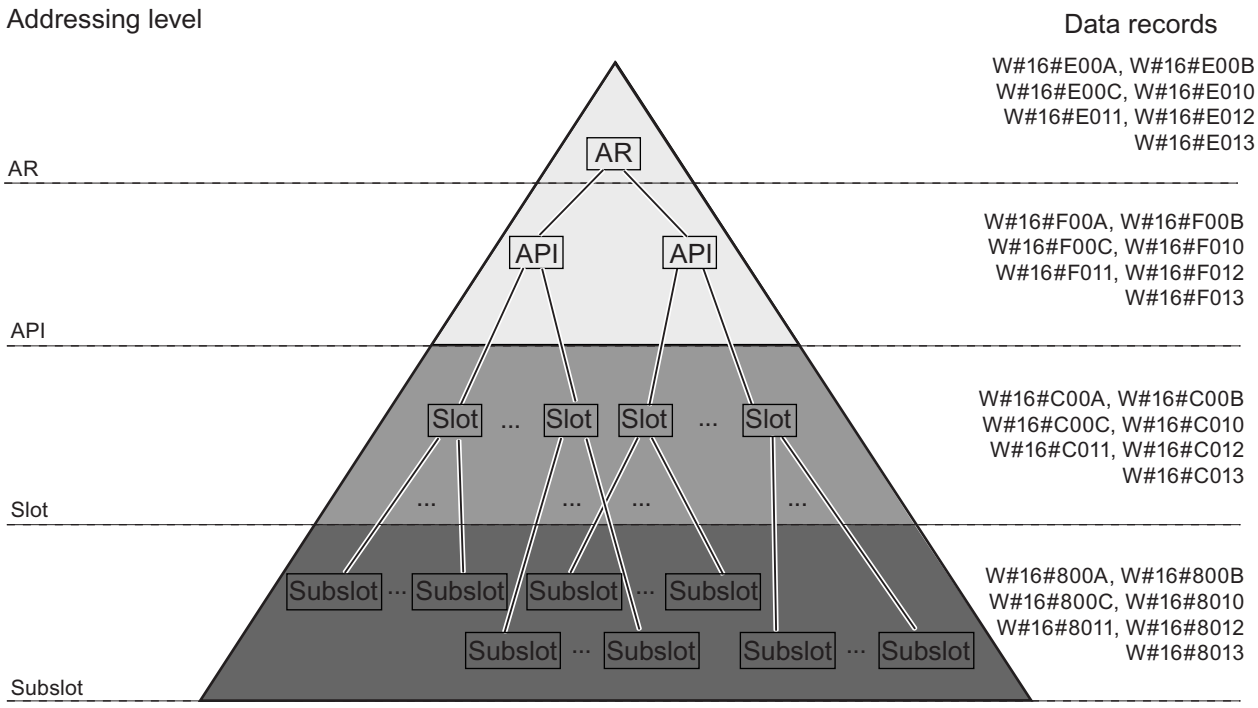

Figure 5-4 Addressing levels for diagnostics records

User Structure Identifier (USI)

The USI identifies the type of diagnostics data:

- Channel diagnostics
- Extended channel diagnostics
- Vendor-specific channel diagnostics

The additional **U**ser **S**tructure **I**dentifier (USI) feature can be used to distinguish between different diagnostics records.

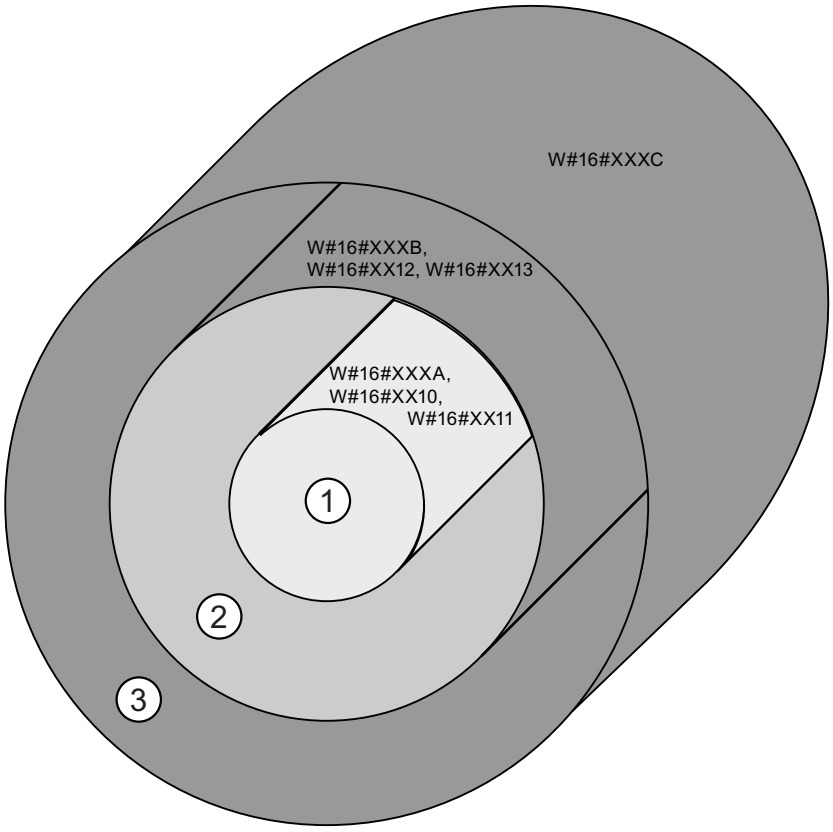

Figure 5-5 Diagnostics record and User Structure Identifier (USI)

Digit	Meaning
①	- USI = W#16#8000 -> channel diagnostics - USI = W#16#8002 -> extended channel diagnostics Note: The record may contain channel diagnostics data and extended channel diagnostics data with MaintenanceRequest, MaintenanceDemanded and error messages.
②	- USI = W#16#0000-W#16#7FFF -> vendor-specific diagnostics - USI = W#16#8000 -> channel diagnostics - USI = W#16#8002 -> extended channel diagnostics Note: The record may contain channel diagnostics data, extended channel diagnostics data and vendor-specific diagnostics data with MaintenanceRequest, MaintenanceDemanded and error messages.
③=①+②	- USI = W#16#0000-W#16#7FFF -> vendor-specific diagnostics - USI = W#16#8000 to W#16#80FF -> range for standardized USI values, for example, W#16#8000 (channel diagnostics) or W#16#8002 (extended channel diagnostics) - USI = W#16#9000 to W#16#9FFF -> profile-specific Note: The records may contain MaintenanceRequest, MaintenanceDemanded and error messages.

Example

The example below shows how to select a suitable record.

You only evaluate the **channel diagnostics data** of a **slot** on the ET 200S. Use table 5-4 to select the suitable record as follows:

1. **Channel diagnostics data** is contained in all diagnostics records with the **USI=W#16#8000 / W#16#0x8002** (see diagram above). This concerns all diagnostics records with record number W#16#X0**0A** (framed row in table 5-6).
2. The diagnostics function is called at **slot** level, which suggests all diagnostics records with the numbers W#16#**C**0XX (framed column in table 5-6) for selection.

You select diagnostics record **W#16#C00A** which contains all available channel diagnostics data of the slot (module).

Addressing levels for Diagnosis information with Faults

Addressing level	Subslot	Slot	API	AR
①	W#16#800A	W#16#C00A	W#16#F00A	W#16#E00A
②	W#16#800B	W#16#C00B	W#16#F00B	W#16#E00B
③	W#16#800C	W#16#C00C	W#16#F00C	W#16#E00C

Addressing levels for Diagnosis information with Maintenance demand

Addressing level	Subslot	Slot	API	AR
①	W#16#8011	W#16#C011	W#16#F011	W#16#E011
②	W#16#8013	W#16#C013	W#16#F013	W#16#E013
③				

Addressing levels for Diagnosis information with Maintenance required

Addressing level	Subslot	Slot	API	AR
①	W#16#8010	W#16#C010	W#16#F010	W#16#E010
②	W#16#8012	W#16#C012	W#16#F012	W#16#E012
③				

Figure 5-6 Addressing levels and records

See above for the key to digits ① to ③.

5.3.2 Structure of diagnostics data records

Block diagram of the structure

The diagram shows the structure of the diagnostics records with their DBs:

- W#16#800A, W#16#800B, W#16#800C, W#16#8010, W#16#8011, W#16#8012, W#16#8013
- W#16#C00A, W#16#C00B, W#16#E00C, W#16#C010, W#16#C011, W#16#C012, W#16#C013
- W#16#E00A, W#16#E00B, W#16#E00C, W#16#E010, W#16#E011, W#16#E012, W#16#E013
- W#16#F00A, W#16#F00B and W#16#F00C, W#16#F010, W#16#F011, W#16#F012, W#16#F013

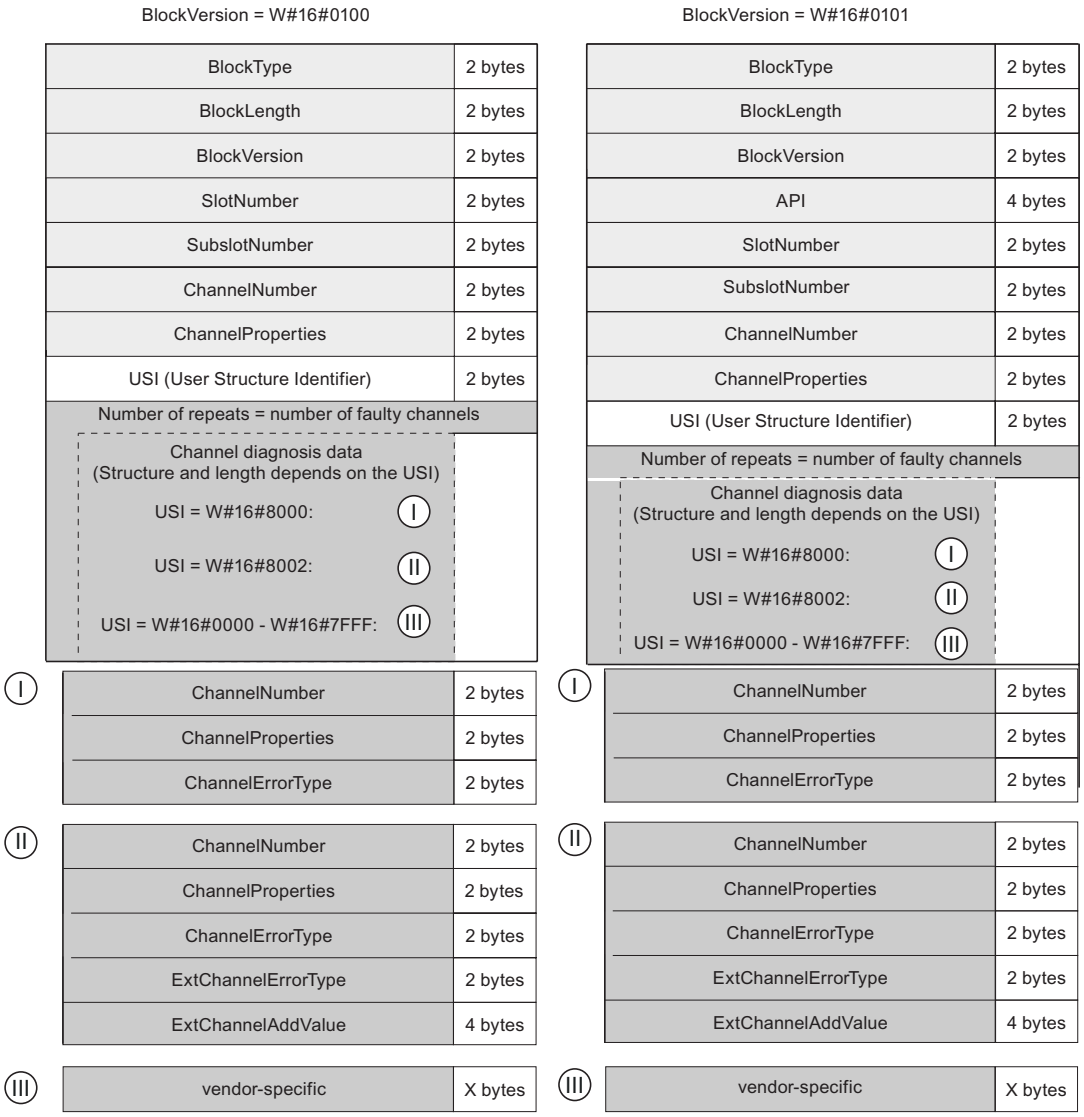

Figure 5-7 Diagnostics records

The diagnostics record with BlockVersion W#16#0100 is located on the left of the diagram; the diagnostics record with BlockVersion W#16#0101 is located on the right. It also contains the Application Process Identifier (API). The API data field shows the coding of all available profiles (PROFIdrive, for example).

The structure of the ChannelDiagnosisData block is based on the User Structure Identifier. All three possible structures are identified by the selection boxes ⓘ–ⓘⓘⓘ.

API

Application **P**rocess **I**dentifier. Term used in PROFINET IO Standard IEC 61158; its value specifies the application which is used to process IO data.

The IEC assigns the APIs specific profiles which are defined in context by the PROFINET User Organisation. The standard API is 0.

Number of faulty channels

General rule: A diagnostics record (ChannelDiagnosisData) is generated for each subslot with faulty channels. A record of length 0 will be sent if no fault is detected.

Several instances of the ChannelDiagnosisData DB are present if several channel faults are detected.

You can determine the number of faulty channels by reading the data value in the BlockLength data field (number of successive bytes). Comparison with the values in the table header identifies the relevant column, while the USI with the BlockVersion gives the row containing the number of faulty channels.

If the BlockLength = 28 for a diagnostics record with USI= W#16#8000 and BlockVersion= W#16#0101, for example, then the table (column 6, row 3) shows that **2 channels** are faulty.

Table 5-9 ChannelDiagnosis and number of channels

USI	BlockVersion	BlockLength							
		18 Byte	22 Byte	24 Byte	28 Byte	30 Byte	34 Byte	36 Byte	40 Byte
W#16#8000	W#16#0001	1 channel	-	2 channels	-	3 channels	-	4 channels	-
	W#16#0101	-	1 channel	-	**2 channels, example** (see also section 6.3)	-	3 channels	-	4 channels
W#16#8002	W#16#0001	-	-	1 channel	-	-	-	2 channels	-
	W#16#0101	-	-	-	1 channel	-	-	-	2 channels

Data blocks, detailed information

For detailed information about the various DBs of the diagnostics records, refer to section 5.5.

5.3.3 Evaluating diagnostics data

Task definition

You want to evaluate diagnostics and status data of an IO device in the application program.

The example below demonstrates the evaluation of diagnostics data record W#16#E00C.

Note the following information:

- Chapter 6.3: The description of the procedure is based on the example of diagnostics data record W#16#E00C.
- Chapter 5.3.2: Structure of diagnostics data records

General procedure

1. Read diagnostics data record W#16#E00C by calling SFB52.
2. Evaluate the LEN parameter of SFB52 -> Result: LEN = 58.
3. Read the values of the following parameters of the diagnostics data record:
   - BlockLength in bytes 2 and 3 -> Result: BlockLength = W#16#001C; converted = 28 bytes
   - BlockVersion in bytes 4 and 5 -> Result: BlockVersion = W#16#0101
   - USI for BlockVersion W#16#0101 in bytes 18 and 19 -> Result: USI = W#16#8000 -> Returns 6 bytes of channel diagnostics data per faulty channel.

Result:

Reading the value of BlockLength and knowledge of the structure of diagnostics record W#16#E00C for BlockVersion W#16#0101 with USI W#16#8000 gives the following result:

The record with a total length of 32 bytes contains two channel diagnostics records.

	BlockLength = W#16#001C = 16 + 6 + 6 = 28		
BlockType + BlockLength	**BlockVersion ... USI**	**Diagnostics data for channel 1**	**Diagnostics data for channel 0**
Total length of this record = 4 bytes + 16 bytes + 6 bytes + 6 bytes = 32 bytes			

Additional diagnostics data is available for evaluation because LEN > 32 bytes.

1. Again, read the values of the following parameters of the second diagnostics record:
   - BlockLength in byte 34 and 35 -> Result: BlockLength = W#16#0016; converted = 22 bytes
   - BlockVersion in byte 36 and 37 -> Result: BlockVersion = W#16#0101
   - USI in bytes 50 and 51 for BlockVersion W#16#0101 -> Result: USI = W#16#8000 -> Returns 6 bytes of channel diagnostics data per faulty channel.

Result:

Reading the value of BlockLength and knowledge of the record structure of diagnostics record W#16#E00C for BlockVersion 0101 with USI W#16#8000 returns the following result:

The record with a total length of 26 bytes contains two channel diagnostics records.

	BlockLength = 16 bytes + 6 bytes = 22 bytes	
BlockType + BlockLength	**BlockVersion ... USI**	**Diagnostics data for channel 0**
Total length of this record = 4 bytes + 16 bytes + 6 bytes = 26 bytes		

Overall result

The first record has a length of 32 bytes, the second has a length of 26 bytes. The length total of both records is 58 bytes. This value is consistent with parameter LEN = 58 bytes and indicates that all the data has been evaluated and that no other data exists.

5.4 Structure of the configuration data records

5.4.1 Structure of the configuration data records W#16#8000, W#16#8001, W#16#C000, W#16#C001, W#16#E000, W#16#E001, W#16#E002, W#16#F000

Introduction

The basic **structure** of configuration records W#16#8000, W#16#8001, W#16#C000, W#16#C001, W#16#E000, W#16#E001, and W#16#F000 is the same.

The **content** and **size** of a configuration record differs according to the type of configuration.

Record identifier

You can select a suitable record for specific configuration information in a user program with reference to the record names. Their structure is described below.

This relates to the **first digit** and **last digit** of the name of a configuration record:

- **First digit - addressing level**:

  The first digit of the name of a configuration record (for example, W#16#**8**001) identifies the **addressing level** (AR, API, slot, subslot). Configuration data can be requested at this addressing level.

- **Last digit - target / actual configuration**:

  If this is **0** as in W#16#800**0**:

  then it is a configuration record that you can use to request the **target configuration**.

  If it is **1** as in W#16#800**1**:

  then it is a configuration record that you can use to request the **actual configuration**.

Note

Configuration record W#16#F000

Only configuration record W#16#F00**0** is used to request the **actual configuration**. It is therefore an exception in the identification scheme described above.

Addressing level

The model of a PROFINET IO device describes its modular structure, which is based on the general principles of PROFIBUS DP (refer to PROFINET System Manual, pages 2 to 6).

You can use suitable groups of configuration records to request configuration data at a specific addressing level, that is, AR, API, slot or subslot. The first digit of the configuration record identifies the group.

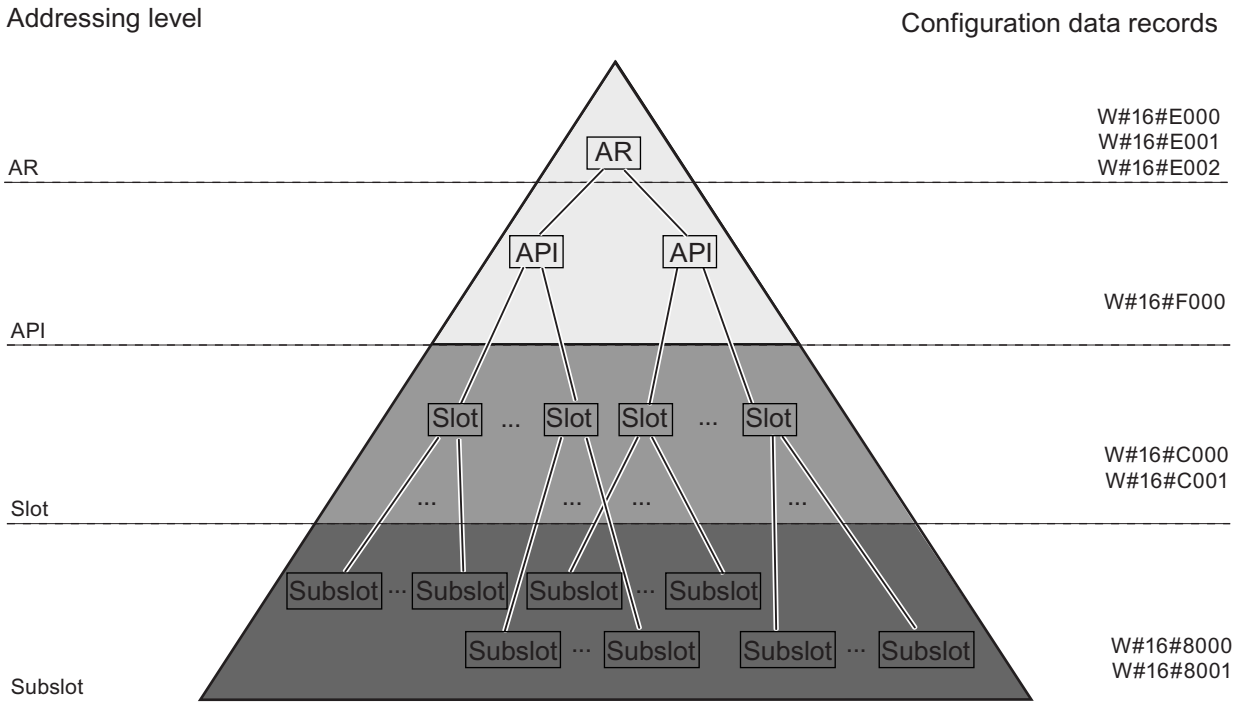

Figure 5-8 Addressing levels and associated configuration records

Example

The following example explains how to select a suitable record.

You want to read the **actual configuration** from a **slot** at an ET 200S.

To select the right configuration record from the table of "Configuration records for target and actual configuration":

1. Read the **actual configuration** using the configuration record with a "1" as the last digit of the record number, that is, W#16#8001, W#16#C001, W#16#E001 and W#16#F000 (third row in the table "Configuration records for target and actual configuration").
2. The configuration is addressed at the **slot** level (third column of the table "Configuration records for target and actual configuration").
3. This returns record **W#16#C001**which you can use to read the actual configuration of any slot.

	Addressing level			
Configuration case	Subslot	Slot	API	AR
Desired configuration	W#16#8000	W#16#C000		W#16#E000
Actual configuration	W#16#8001	W#16#C001	W#16#F000	W#16#E001
Desired – actual difference				W#16#E002

Figure 5-9 Configuration records for the target and actual configuration

Block diagram of the structure

The diagram below shows the structure of the configuration records:

- W#16#8000, W#16#8001
- W#16#C000, W#16#C001
- W#16#E000, W#16#E001
- W#16#F000

Including their data blocks

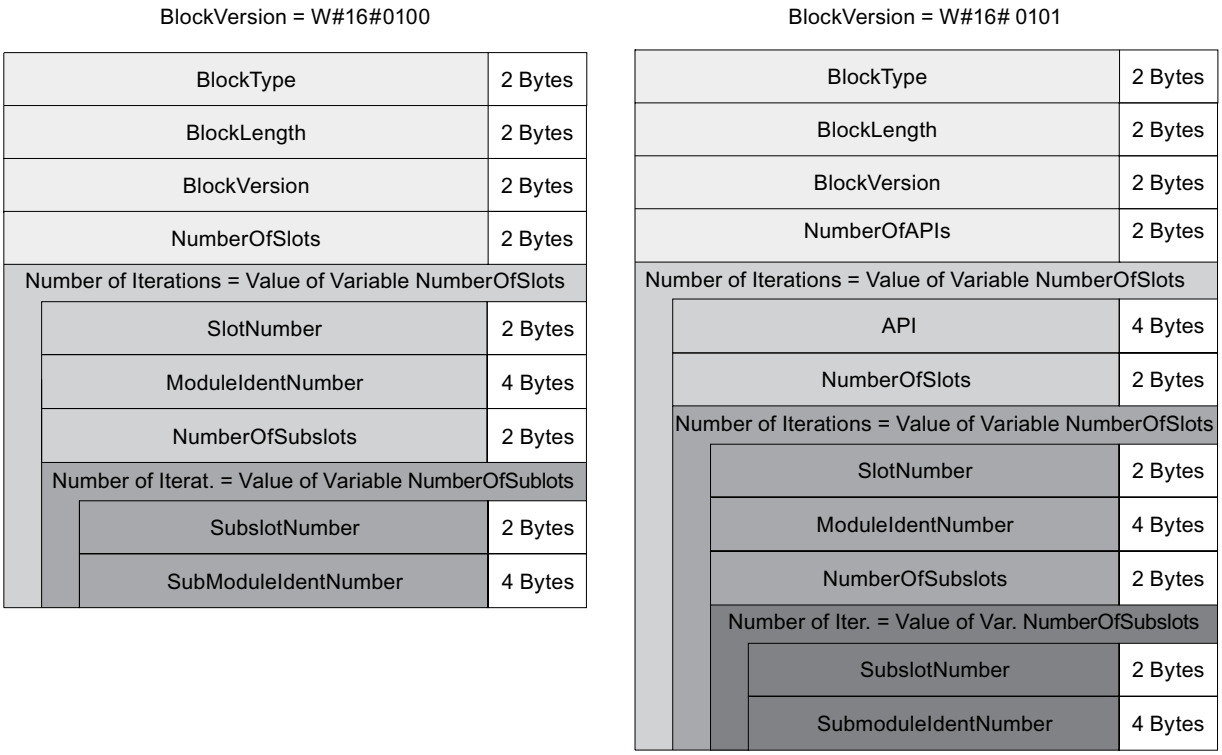

Figure 5-10 PROFINET IO record W#16#8000, W#16#8001, W#16#C000, W#16#C001, W#16#E000, W#16#E001, W#16#F000

The configuration record with BlockVersion W#16#0100 is located on the left of the diagram, and the configuration record with BlockVersion W#16#0101 is located on the right.

Configuration records with BlockVersion W#16#0101 support multiple APIs and also include the Application Process Identifier (API). The API data field shows the coding of all available profiles (PROFIdrive, for example).

Data blocks, detailed information

For detailed information about the various DBs of the diagnostics records, refer to section 5.5.

5.4.2 Structure of the configuration data record W#16#E002

Schematic Structure

The following graphic shows the structure of the configuration data record W#16#E002 and the composition from the data blocks shown.

The data blocks are described in the chapter 5.5 Blocks of the diagnostics and configuration records.

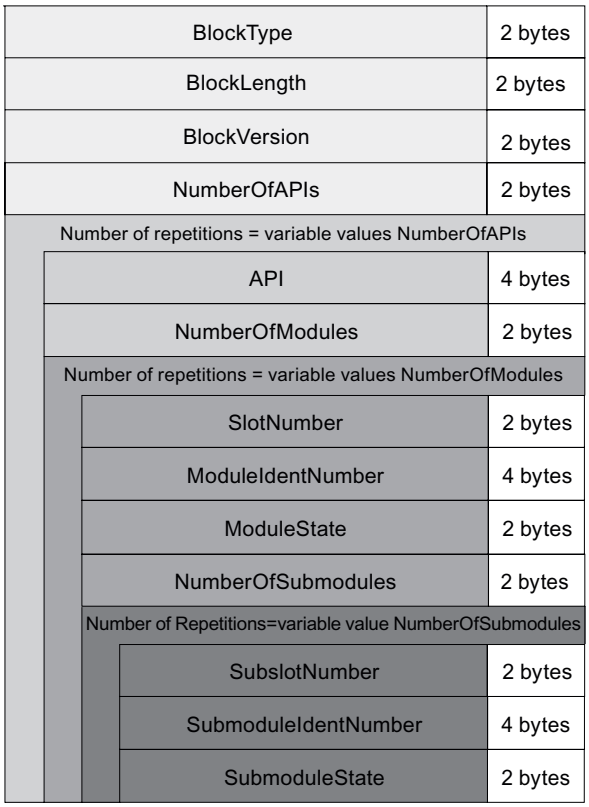

Figure 5-11 PROFINET IO record W#16#E002

5.5 Blocks of the diagnostics and configuration records

5.5.1 API

Table 5-10 Address space of API

API (hexadecimal value)	Meaning
W#16#0000	Default value
W#16#00000001 to W#16#FFFFFFFF	Address space for defined profiles

5.5.2 BlockLength

The BlockLength data field contains the coding of the **number of successive bytes** of the diagnostics or configuration record. This code returns the length of the diagnostics or configuration record, without the number of bytes for the BlockType and BlockLength data fields, which each have a length of 2 bytes.

5.5.3 BlockType

Table 5-11 BlockType coding

BlockType	Meaning
W#16#0001	Interrupt transfer channel 1
W#16#0002	Interrupt transfer channel 2
W#16#0010	Diagnostics record
W#16#0012	Configuration record for target configuration
W#16#0013	Configuration record for actual configuration
W#16#8104	Configuration record Target/actual comparison

5.5.4 BlockVersion

The *BlockVersion* data block in turn consists of the *BlockVersionHigh* and *BlockVersionLow* DBs, which are each one byte long.

Table 5-12 Coding of BlockVersion

BlockVersion	Value (hexadecimal)	Meaning
BlockVersionHigh	B#16#01	Identifies the first value of the version number, W#16#01xx
BlockVersionLow	B#16#00 or B#16#01	Version number W#16#010**0** or W#16#010**1**

5.5.5 ChannelErrorType

Table 5-13 ChannelErrorType coding

Value (hexadecimal)	Meaning	Error message
W#16#0000	Reserved	Unknown error
W#16#0001	Short-circuit	Short-circuit
W#16#0002	Undervoltage	Undervoltage
W#16#0003	Overvoltage	Overvoltage
W#16#0004	Overload	Overload
W#16#0005	Overtemperature	Overtemperature
W#16#0006	Wire break	Wire break
W#16#0007	Violation of high limit	Violation of high limit
W#16#0008	Violation of low limit	Violation of low limit
W#16#0009	Error	Error
W#16#000A to W#16#000F	Reserved	Unknown error
W#16#0010	Vendor-specific Incorrect parameter assignment	Incorrect parameter assignment
W#16#0011	Vendor-specific Power supply failure	Power supply failure
W#16#0012	Vendor-specific Fuse blown/tripped	Fuse blown/tripped
W#16#0013	Vendor-specific	Vendor-specific
W#16#0014	Vendor-specific Ground fault	Ground fault
W#16#0015	Vendor-specific Reference point not found	Reference point not found
W#16#0016	Vendor-specific Sampling error	Sampling error
W#16#0017	Vendor-specific Violation of threshold limits	Violation of threshold limits

Value (hexadecimal)	Meaning	Error message
W#16#0018	Vendor-specific Output switched off	Output switched off
W#16#0019	Vendor-specific Safety-related fault	Safety-related fault
W#16#001A	Vendor-specific External fault	External fault
W#16#001B to W#16#001F	Vendor-specific	Vendor-specific
W#16#0020 to W#16#00FF	Reserved for standard profiles for all devices	Standard profiles for all devices (for example, PROFIsafe)
W#16#0100 to W#16#7FFF	Vendor-specific	Vendor-specific
W#16#8000	Data transfer not possible	Data transfer not possible
W#16#8001	Wrong neighborhood	Wrong neighborhood
W#16#8002	Redundancy loss	Redundancy loss
W#16#8003	Loss of synchronization (on the bus side)	Loss of synchronization (on the bus side)
W#16#8004	Loss of isochronous operation (at the device)	Loss of isochronous operation (at the device)
W#16#8005	Direct communication error	Direct communication error
W#16#8006	Reserved	Reserved
W#16#8007	Fiber-optics Error	Optical transfer not possible
W#16#8008	Error Network component	Problems with network function
W#16#8009	Timebase error	No timer or problems with timebase accuracy
W#16#800A to W#16#8FFF	Reserved	Unknown error
W#16#9000 to W#16#9FFF	Reserved for technological profiles (for example, PROFIdrive)	Profile-specific
W#16#A000 to W#16#FFFF	Reserved	Unknown error

5.5.6 ChannelNumber

Table 5-14 ChannelNumber coding

Value (hexadecimal)	Meaning
W#16#0000 to W#16#7FFF	Vendor-specific
W#16#8000	Submodule
W#16#8001 to W#16#FFFF	Reserved

5.5.7 ChannelProperties

Structure of ChannelProperties

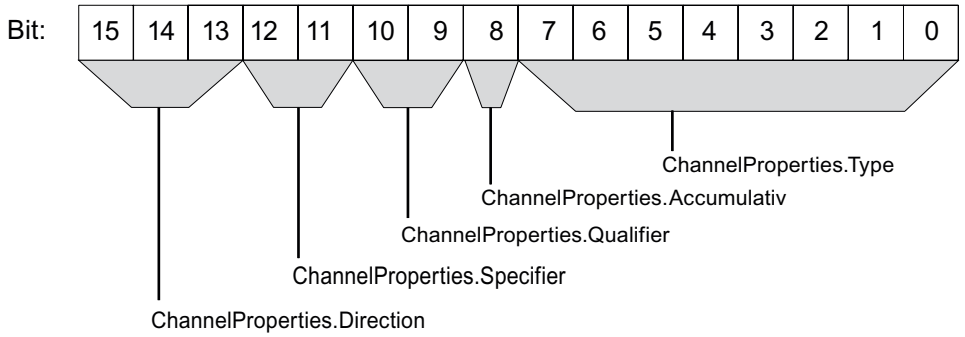

Figure 5-12 Structure of ChannelProperties

5.5.7.1 ChannelProperties.Type (Bit 0 - 7)

Table 5-15 Coding of ChannelProperties.Type

Value (hexadecimal)	Meaning
B#16#00	If ChannelNumber is W#16#8000 (submodule).
B#16#01	1 bit
B#16#02	2 bits
B#16#03	4 bits
B#16#04	8 bits
B#16#05	16 bits
B#16#06	32 bits
B#16#07	64 bits
B#16#08 - B#16#FF	Reserved

5.5.7.2 ChannelProperties.Accumulative (Bit 8)

Table 5-16 ChannelProperties.Accumulative coding

Value (hexadecimal)	Name	Meaning
0	-	No channel error group message (concerns only one channel)
1	Accumulative	Channel error group message (concerns several channels)

5.5.7.3 Combination of ChannelPropertes.Qualifier (Bit 9/10) and ChannelProperties.Specifier (Bit 11/12)

Table 5-17 Combination of the values of MaintenanceRequired / MaintenanceDemanded and Specifier

MaintenanceRequired (bit 9)	MaintenanceDemanded (bit 10)	Specifier (Bit 12/11)	Meaning	Possible at
0	0	00	All sublevel* diagnostics - MaintenanceRequired, MaintenanceDemanded and Qualified Diagnosis - are no longer active	Evaluation of diagnostics interrupts by calling SFB54 in OB82
		01	Diagnostics active	Evaluation of diagnostics interrupts using SFB54 in OB82 or by calling SFB52 to read the record
		10	Diagnostics no longer active	Evaluation of diagnostics interrupts by calling SFB54 in OB82
		11	Status message - only available in combination with vendor-specific error	Evaluation of diagnostics interrupts by calling SFB54 in OB82
0	1	00	Reserved	—
		01	MaintenanceDemanded active	Evaluation of diagnostics interrupts using SFB54 in OB82 or by calling SFB52 to read the record
		10	MaintenanceDemanded no longer active	Evaluation of diagnostics interrupts by calling SFB54 in OB82
		11	MaintenanceDemanded no longer active - all others remain active	
1	0	00	Reserved	—
		01	MaintenanceRequired active	Evaluation of diagnostics interrupts using SFB54 in OB82 or by calling SFB52 to read the record
		10	MaintenanceRequired no longer active	Evaluation of diagnostics interrupts by calling SFB54 in OB82
		11	MaintenanceRequired no longer active - all others remain active	

MaintenanceRequired (bit 9)	MaintenanceDemanded (bit 10)	Specifier (Bit 12/11)	Meaning	Possible at
1	1	00	Reserved	—
		01	Graded diagnostics active	Evaluation of diagnostics interrupts using SFB54 in OB82 or by calling SFB52 to read the record
		10	Graded diagnostics no longer active	Evaluation of diagnostics interrupts by calling SFB54 in OB82
		11	Graded diagnostics no longer active - all others remain active	

Sublevel in this context refers to the clearance of all characteristic data of the ExtChannelErrorType and ChannelErrorType blocks from outgoing events.

5.5.7.4 ChannelProperties.Specifier (Bit 11 - 12)

Table 5-18 ChannelProperties.Specifier coding

Value (hexadecimal)	Meaning	Possible at
00	Reserved	--
01	Diagnostics pending	Evaluation of diagnostics interrupts using SFB54 in OB82 or by calling SFB52 to read the record
10	Outgoing event and no other events	Evaluation of diagnostics interrupts by calling SFB54 in OB82
11	Outgoing event but others remain	Evaluation of diagnostics interrupts by calling SFB54 in OB82

5.5.7.5 ChannelProperties.Direction (Bit 13 - 15)

Table 5-19 ChannelProperties.Direction coding

Value	Meaning
000	Vendor-specific
001	Input
002	Output
003	Input/output
004 - 007	Reserved

5.5.8 ExtChannelAddValue

This field is of the Unsigned32 data type.

The content of this data field is 0 if information for extended channel diagnostics is not available.

5.5.9 ExtChannelErrorType

Table 5-20 ExtChannelErrorType coding

Value (hexadecimal)	Meaning
W#16#0000 to W#16#FFFF	Coding depends on the ChannelErrorType See the PROFINET IO Application Layer Service Definition & Application Layer Protocol Specification or IEC 61158

Table 5-21 Coding of ExtChannelErrorType for ChannelErrorType W#16#0000 - W#16#7FFF

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Group message
W#16#8001 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-22 Coding of ExtChannelErrorType for ChannelErrorType "No data transfer possible"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Port status error - cable not connected, for example
W#16#8001	Error due to incorrect interface setting – full duplex and half duplex
W#16#8002	Error due to runtime delay – the configured cable length does not match the real cable length
W#16#8003 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-23 Coding of ExtChannelErrorType for ChannelErrorType "wrong neighborhood"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Wrong neighbor - device
W#16#8001	Wrong neighbor - port
W#16#8002	Neighbor does not support RealTime Class 3 or is not configured
W#16#8003	Error due to incorrect interface setting – full duplex and half duplex
W#16#8004	incorrect or missing configuration of media redundancy
W#16#8005	No neighbor available
W#16#8006	Neighbor does not support bumpless media redundancy
W#16#8007 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-24 Coding of ExtChannelErrorType for ChannelErrorType "Redundancy loss"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Media Redundancy Manager reports error
W#16#8001	Ring open – media redundancy no longer available
W#16#8002	Ring open – bumpless media redundancy no longer available
W#16#8003	Several media redundancy managers in the ring
W#16#8004 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-25 Coding of ExtChannelErrorType for ChannelErrorType "Loss of isochrone mode" and "Timebase error"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	No synchronization received
W#16#8001	RealTime Class 3 – incorrect synchronization configuration
W#16#8002	RealTime Class 3 – incorrect configuration
W#16#8003	Jitter out of limits
W#16#8004 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-26 Coding of ExtChannelErrorType for ChannelErrorType "Isochrone state error"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Transfer timeout at outputs
W#16#8001	Transfer timeout at outputs
W#16#8002 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-27 Coding of ExtChannelErrorType for ChannelErrorType "Multicast CR error"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Data recipient in communication - wrong or missing sender
W#16#8001	Data recipient in communication – unknown sender
W#16#8002 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-28 Coding of ExtChannelErrorType for ChannelErrorType "Optical transfer not possible"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Undershoot of specified receiving level
W#16#8001 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

Table 5-29 Coding of ExtChannelErrorType for ChannelErrorType "Network function error"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 to W#16#7FFF	Vendor-specific
W#16#8000	Network overload - discarding message frames
W#16#8001 to W#16#8FFF	Reserved
W#16#9000 to W#16#9FFF	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved

5.5.10 Module identification number

Table 5-30 ModuleIdentNumber coding

Value (hexadecimal)	Meaning
DW#16#00000000	Reserved
DW#16#00000001 to DW#16#FFFFFFFF	Vendor-specific

5.5.11 Module state

Table 5-31 ModuleState coding

Value (hexadecimal)	Meaning	Description
W#16#0000	Module not found	Module not inserted
W#16#0001	Incorrect module	Incorrect ModuleIdentNumber
W#16#0002	Correct module	Module is OK, but at least one submodule is disabled, incorrect or missing - or a submodule is running diagnostics
W#16#0003	Substitute	Module is not the one requested - but is compatible. The IO device is capable of adapting itself to the module
W#16#0004 to W#16#FFFF	Reserved	

5.5.12 SlotNumber

Table 5-32 SlotNumber coding

Value (hexadecimal)	Meaning
W#16#0000 to W#16#7FFF	The first slot number is zero. The last slot number is W#16#7FFF.
W#16#8000 to W#16#FFFF	Reserved

5.5.13 SubmoduleIdentNumber

Table 5-33 SubmoduleIdentnumber coding

Value (hexadecimal)	Meaning
DW#16#00000000 to DW#16#FFFFFFFF	Vendor-specific

5.5.14 SubmoduleState

Structure of SubmoduleState

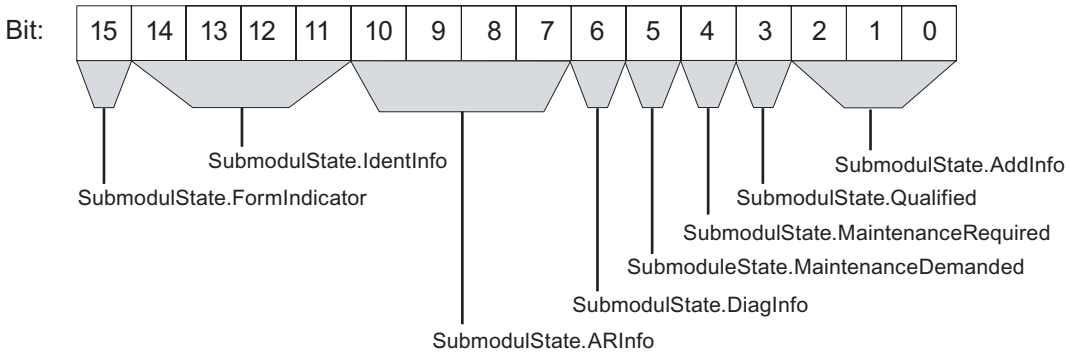

Figure 5-13 Structure of SubmoduleState

5.5.14.1 SubmoduleState.AddInfo (Bit 0 - 2)

Table 5-34 SubmoduleState.AddInfo coding

Value	Meaning	Description
000	No meaning	
001	Acceptance not allowed	Submodule is not suitable for acceptance from IO supervisor AR
002	Reserved	

5.5.14.2 SubmoduleState.MaintenanceRequired (Bit 4)

SubmoduleState.MaintenanceRequired (Bit 4)

Table 5-35 SubmoduleState.MaintenanceRequired coding

Value (hexadecimal)	Meaning	Description
0	No MaintenanceRequired active	MaintenanceRequired not available for this submodule.
1	MaintenanceRequired available	MaintenanceRequired is available for this submodule.

5.5.14.3 SubmoduleState.MaintenanceDemanded (Bit 5)

SubmoduleState. MaintenanceDemanded (bit 5)

Table 5-36 SubmoduleState.MaintenanceDemanded coding

Value (hexadecimal)	Meaning	Description
0	No MaintenanceDemanded active	MaintenanceDemanded not available for this submodule.
1	MaintenanceDemanded available	MaintenanceDemanded available for this submodule.

5.5.14.4 SubmoduleState.DiagInfo (Bit 6)

SubmoduleState.DiagInfo (Bit 6)

Table 5-37 SubmoduleState.DiagInfo coding

Value (hexadecimal)	Meaning	Description
0	No diagnostics data available	No diagnostics data available or saved for this submodule.
1	Diagnostics data available	Diagnostics data available for this submodule: The data can be read using the corresponding records.

5.5.14.5 SubmoduleState.ARInfo (Bit 7 - 10)

Table 5-38 SubmoduleState.ARInfo coding

Value	Meaning	Description
0000	User	Submodule is available to the user
0001	ApplicationReadyPending	Submodule is not available to the user, for example, on account of parameter errors
0002	Locked	Submodule is not available to the user. Example: simultaneous request of several overriding functions
0003	Locked by IO controller	The controller is not the owner of the submodule (submodule is not available to the user)
0004	Locked by IO supervisor	The controller is not the owner of the submodule (submodule is not available to the user)
0005 to 000F	Reserved	Reserved

5.5.14.6 SubmoduleState.IdentInfo (Bit 11 - 14)

Table 5-39 SubmoduleState.IdentInfo coding

Value	Meaning
0000	OK
0001	Substitute
0002	Incorrect
0003	No submodule
0004 to 000F	Reserved

5.5.14.7 SubmoduleState.FormIndicator (Bit 15)

SubmoduleState.FormatIndicator (Bit 15)

Table 5-40 SubmoduleState coding. FormatIndicator

Value (hexadecimal)	Meaning	Description
1	SubmoduleState consists of SubmoduleState.IdentInfo, .ARInfo and .AddInfo	Supported by the IO controller, IO device and IO supervisor.
0	Reserved	Reserved

5.5.15 SubslotNumber

Table 5-41 SubslotNumber coding

Value (hexadecimal)	Meaning
W#16#0000	Determined by the module; does not address the submodule.
W#16#0001 to W#16#7FFF	The first subslot number for the submodule is one. The last subslot number for the submodule is W#16#7FFF.
W#16#8000 to W#16#8FFF	Used for 16 interface modules with up to 255 ports 0x8IPP with I counting interfaces and P counting ports; PP := 1 to 255; I := 0 to 15 if PP=00; describes the actual interface module for example, 8001: I=0 and PP=01, port 1 of interface 0
W#16#9000 to W#16#FFFF	Reserved

5.5.16 USI

Table 5-42 USI (UserStructureIdentifier) coding

Value (hexadecimal)	Meaning	Description
W#16#0000 to W#16#7FFF	Vendor-specific	When combined with the interrupt type, diagnostics incoming/outgoing is vendor-specific Diagnostics in AlarmNotification and diagnostics data. Vendor-specific usage in combination with other interrupt types.
W#16#8000	ChannelDiagnosis	Used only in combination with ChannelDiagnosis in AlarmNotification and diagnostics data.
W#16#8001	Multiple	Only used in combination with data which correspond to the structure of “(BlockHeader, Data*)*”. BlockType always corresponds to the AlarmType used.
W#16#8002	ExtChannelDiagnosisData	Used only in combination with ChannelDiagnosisWithAddInfo in AlarmNotification and diagnosis data.
W#16#8003	Qualified	Graded extended channel diagnostics
W#16#8004 to W#16#80FF	Reserved	
W#16#8100	Maintenance	Maintenance
W#16#8101 to W#16#8FFF	Reserved	
W#16#9000 - W#16#9FFF	Reserved for profiles	Reserved for profiles
W#16#A000 to W#16#FFFF	Reserved	

Examples of diagnostics data records

6

6.1 Example of diagnostics data record W#16#800A

Example of diagnostics record W#16#800A

Record W#16#800A is read from subslot 1 / slot 2. Wire break at one of the two submodule channels (outputs).

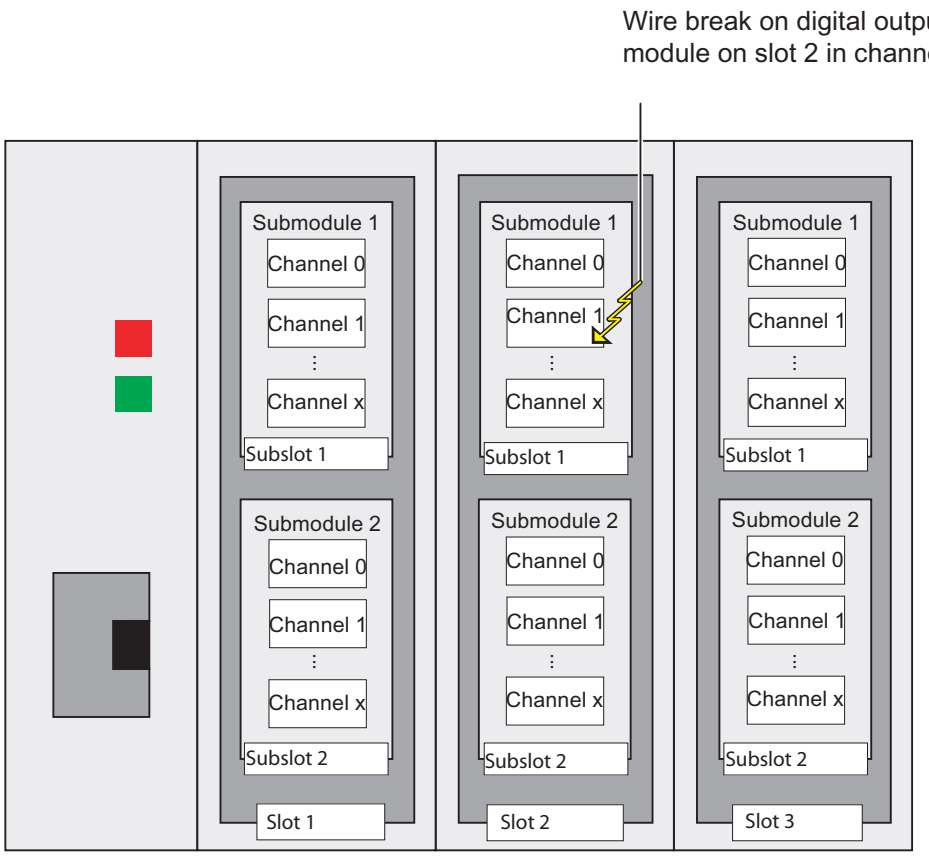

Figure 6-1 Wire break at channel 1 / subslot 1 / slot 2

The content of the diagnostics record is then as follows:

Table 6-1 Example of diagnostics record W#16#800A for a faulty channel

DB name		Content	Comment
Diagnostics record W#16#800A returns **one** record for a subslot (subslot level was addressed with this record); only one channel diagnostics record is available as only one channel is faulty.			
BlockHeader		BlockHeader consists of BlockType, BlockLength, BlockVersion	
	BlockType	W#16#0010	DiagnosisBlock, that is, this is a diagnostics record
	BlockLength	W#16#0016	Decimal 22, that is, 22 bytes are appended to the BlockLength data block.
	BlockVersion	W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
API		DW#16#00000000	API is 0; that is, it has no profile
SlotNumber		W#16#0002	Module in slot 2
SubslotNumber		W#16#0001	First subslot
ChannelNumber		W#16#8000	Diagnostics at submodule level
ChannelProperties		W#16#0800	In the binary system: 0000 1000 0000 0000 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
	ChannelProperties.Type (bit 0 to 7)	W#16#00	Set to 0 if ChannelNumber = W#16#8000
	ChannelProperties.Accumulative (bit 8)	0	No channel error group message
	MaintenanceRequired (bit 9)	0100	Diagnostics pending
	MaintenanceDemanded (bit 10)		
	ChannelProperties.Specifier (bit 11 to 12)		
	ChannelProperties.Direction (bit 13 to 15)	000	Vendor-specific
USI		W#16#8000	Three data blocks are appended to this DB for each faulty channel: ChannelNumber, ChannelProperties and ChannelErrorType

DB name			Content	Comment
The data blocks shown below are generated for each faulty channel; channel 1 in this example				
	ChannelNumber		W#16#0001	Channel 1
	ChannelProperties		W#16#4801	In the binary system: 0100 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9)	0100	Diagnostics pending
		MaintenanceDemanded (bit 10)		
		ChannelProperties.Specifier (bit 11 to 12)		
		ChannelProperties.Direction (bit 13 to 15)	010	Output
	ChannelErrorType		W#16#0006	Wire break

Record W#16#800A is read from slot 2 / subslot 1. Wire break at one of the two submodule channels (outputs).

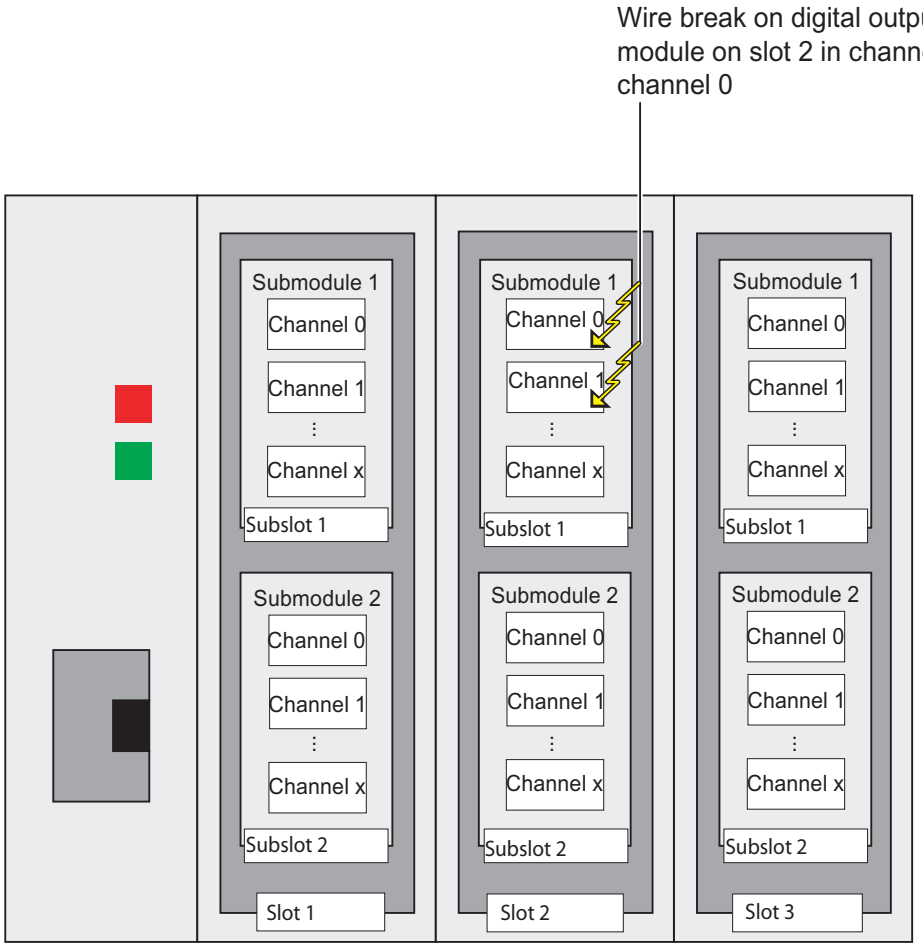

Figure 6-2 Wire break at the digital outputs channel 0 & channel 1 / subslot 1 / slot 2

The content of the diagnostics record is then as follows:

Table 6-2 Example of diagnostics record W#16#800A for two faulty channels

DB name		Content	Comment
Diagnostics record W#16#800A returns just one record for the subslot (subslot level was addressed with this record); two records are returned because there are two faulty channels.			
BlockHeader		BlockHeader consists of BlockType, BlockLength, BlockVersion	
	BlockType	W#16#0010	DiagnosisBlock, that is, this is a diagnostics record
	BlockLength	W#16#001C	Decimal 28, that is, 28 bytes are appended to the BlockLength data block.
	BlockVersion	W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
API		DW#16#00000000	API is 0; that is, it has no profile
SlotNumber		W#16#0002	Module in slot 2
SubslotNumber		W#16#0001	First subslot

DB name			Content	Comment
ChannelNumber			W#16#8000	Diagnostics at submodule level
ChannelProperties			W#16#0800	In the binary system: 0000 1000 0000 0000 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
	ChannelProperties.Type (bit 0 to 7)		B#16#00	Set to 0 if ChannelNumber = W#16#8000
	ChannelProperties.Accumulative (bit 8)		0	No channel error group message
	MaintenanceRequired (bit 9) **MaintenanceDemanded (bit 10)** **ChannelProperties.Specifier (bit 11 to 12)**		0100	Diagnostics pending
	ChannelProperties.Direction (bit 13 to 15)		000	Vendor-specific
USI			W#16#8000	Three data blocks are appended to this DB for each faulty channel: ChannelNumber, ChannelProperties and ChannelErrorType
Followed by channel diagnostics for faulty channel 1				
	ChannelNumber		W#16#0001	Channel 1
	ChannelProperties		W#16#4801	In the binary system: 0100 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9) **MaintenanceDemanded (bit 10)** **ChannelProperties.Specifier (bit 11 to 12)**	0100	Diagnostics pending
		ChannelProperties.Direction (bit 13 to 15)	010	Output
	ChannelErrorType		W#16#00 06	Wire break

DB name			Content	Comment
Followed by channel diagnostics for faulty channel 0				
	ChannelNumber		W#16#0000	Channel 0
	ChannelProperties		DW#16#4801	In the binary system: 0100 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9)	0100	Diagnostics pending
		MaintenanceDemanded (bit 10)		
		ChannelProperties.Specifier (bit 11 to 12)		
		ChannelProperties.Direction (bit 13 to 15)	010	Output
	ChannelErrorType		W#16#0006	Wire break

6.2 Example of the diagnostics data record W#16#800C

Example of diagnostics record W#16#800C

Diagnostics record W#16#800C is read from subslot 1 / slot 3. Short-circuit at one of the two submodule channels (inputs).

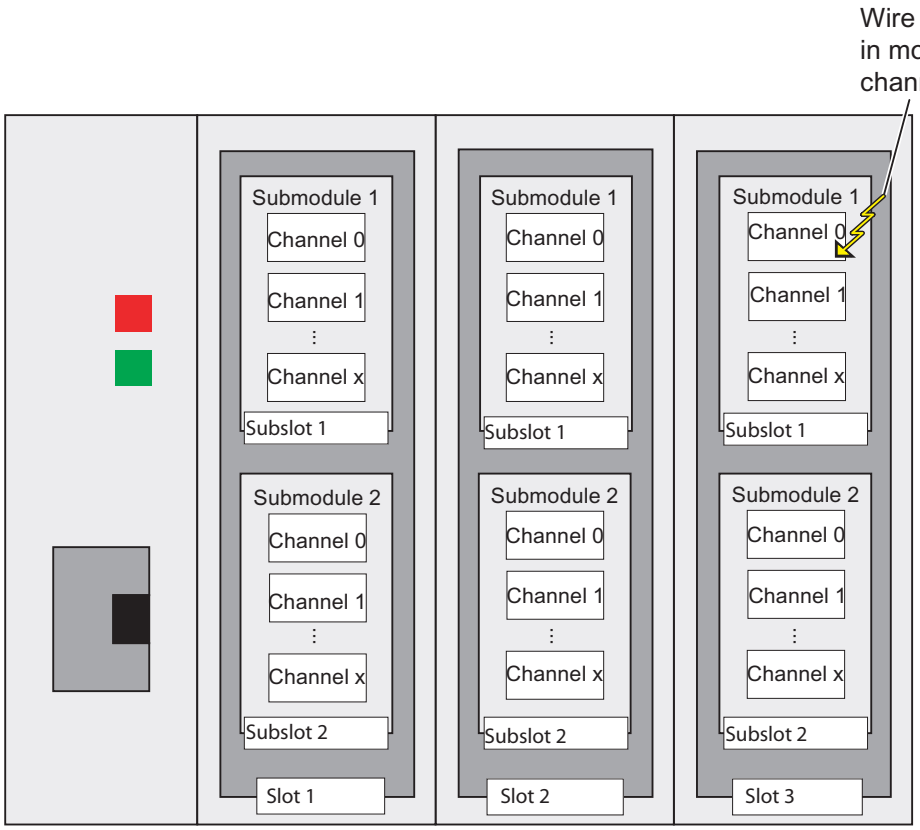

Figure 6-3 Short-circuit at the digital input channel 0 / subslot 1 / slot 3

The content of the diagnostics record is then as follows:

Table 6-3 Example of diagnostics record W#16#800C for one faulty channel

DB name	Content	Comment
Diagnostics record W#16#800C returns one record for a subslot (subslot level was addressed with this record); it returns one channel diagnostics record because only one channel is faulty.		
BlockHeader	BlockHeader consists of BlockType, BlockLength, BlockVersion	
BlockType	W#16#0010	DiagnosisBlock, that is, this is a diagnostics record
BlockLength	W#16#0016	Decimal 22, that is, 22 bytes are appended to the BlockLength data block.
BlockVersion	W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
API	DW#16#00000000	API is 0; that is, it has no profile
SlotNumber	W#16#0003	Module in slot 3
SubslotNumber	W#16#0001	First subslot
ChannelNumber	W#16#8000	Diagnostics at submodule level
ChannelProperties	W#16#0800	In the binary system: 0000 1000 0000 0000 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
ChannelProperties.Type (bit 0 to 7)	B#16#00	Set to 0 if ChannelNumber = W#16#8000
ChannelProperties.Accumulative (bit 8)	0	No channel error group message
MaintenanceRequired (bit 9)	0100	Diagnostics pending
MaintenanceDemanded (bit 10)		
ChannelProperties.Specifier (bit 11 to 12)		
ChannelProperties.Direction (bit 13 to 15)	000	Vendor-specific
USI	W#16#8000	Three data blocks are appended to this DB for each faulty channel: ChannelNumber, ChannelProperties and ChannelErrorType

DB name			Content	Comment
Followed by channel diagnostics for faulty channel 0				
	ChannelNumber		W#16#0000	Channel 0
	ChannelProperties		W#16#2801	In the binary system: 0010 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9)	0100	Diagnostics pending
		MaintenanceDemanded (bit 10)		
		ChannelProperties.Specifier (bit 11 to 12)		
		ChannelProperties.Direction (bit 13 to 15)	001	Input
	ChannelErrorType		W#16#0001	Short-circuit

6.3 Example of diagnostics data record W#16#E00C

Example of diagnostics record W#16#E00C

Diagnostics record W#16#E00C is read from a device (AR). Submodule 1 (outputs) of slot 2 returns two wire break diagnostics records, and submodule 1 (inputs) of slot 3 reports a short-circuit.

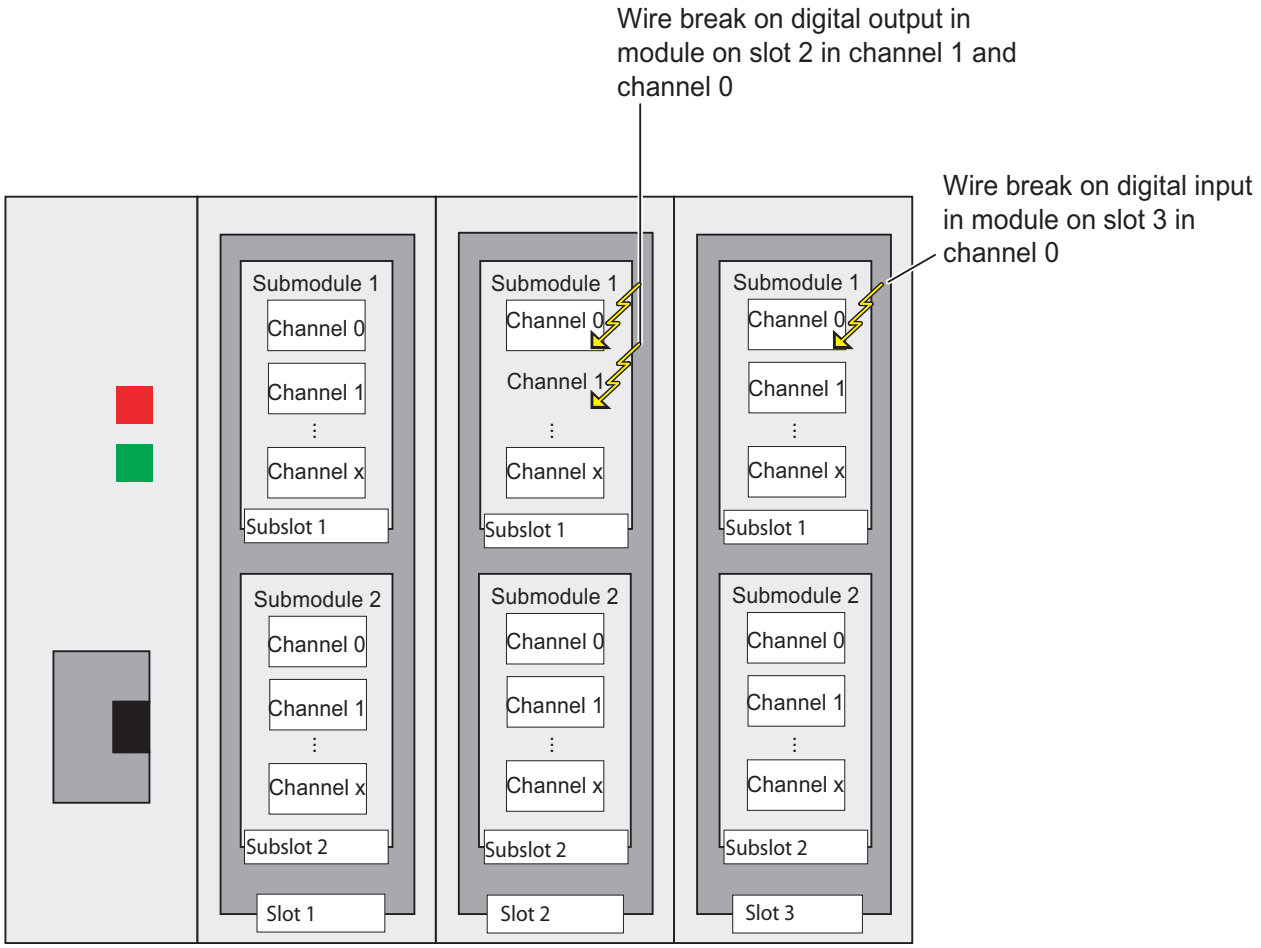

Figure 6-4 Short-circuit at digital input channel 0 / subslot 1 / slot 3; wire break at digital output channel 0 & channel 1 / subslot 1 / slot 2

This gives the following diagnostics record:

Table 6-4 Example of diagnostics record W#16#E00C with two diagnostics records

DB name		Content	Comment
This is followed by a record for slot 2 which contains the faulty channels 1 and 0			
BlockHeader		BlockHeader consists of BlockType, BlockLength, BlockVersion	
	BlockType	W#16#0010	DiagnosisBlock, that is, this is a diagnostics record
	BlockLength	W#16#001C	Decimal 28, that is, 28 bytes are appended to the BlockLength data block.
	BlockVersion	W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
API		DW#16#00000000	API is 0; that is, it has no profile
SlotNumber		W#16#0002	Module in slot 2
SubslotNumber		W#16#0001	First subslot
ChannelNumber		W#16#8000	Diagnostics at submodule level
ChannelProperties		W#16#0800	In the binary system: 0000 1000 0000 0000 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
	ChannelProperties.Type (bit 0 to 7)	B#16#00	Set to 0 if ChannelNumber = W#16#8000
	ChannelProperties.Accumulative (bit 8)	0	No channel error group message
	MaintenanceRequired (bit 9)	0100	Diagnostics pending
	MaintenanceDemanded (bit 10)		
	ChannelProperties.Specifier (bit 11 to 12)		
	ChannelProperties.Direction (bit 13 to 15)	000	Vendor-specific
USI		W#16#8000	Three data blocks are appended to this DB for each faulty channel: ChannelNumber, ChannelProperties and ChannelErrorType

DB name			Content	Comment
Followed by channel diagnostics for faulty channel 1				
	ChannelNumber		W#16#0001	Channel 1
	ChannelProperties		W#16#4801	In the binary system: 0100 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9) **MaintenanceDemanded (bit 10)** **ChannelProperties.Specifier** **(bit 11 to 12)**	0100	Diagnostics pending
		ChannelProperties.Direction **(bit 13 to 15)**	010	Output
	ChannelErrorType		W#16#0006	Wire break
Followed by channel diagnostics for faulty channel 0				
	ChannelNumber		W#16#0000	Channel 0
	ChannelProperties		W#16#4801	In the binary system: 0100 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9) **MaintenanceDemanded (bit 10)** **ChannelProperties.Specifier** **(bit 11 to 12)**	0100	Diagnostics pending
		ChannelProperties.Direction **(bit 13 to 15)**	010	Output
	ChannelErrorType		W#16#0006	Wire break

DB name			Content	Comment
This is followed by a record for slot 3 which contains the faulty channel 0				
BlockHeader			BlockHeader consists of BlockType, BlockLength, BlockVersion	
	BlockType		W#16#0010	DiagnosisBlock, that is, this is a diagnostics record
	BlockLength		W#16#0016	Decimal 22, that is, 22 bytes are appended to the BlockLength data block.
	BlockVersion		W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
API			DW#16#00000000	API is 0; that is, it has no profile
SlotNumber			W#16#0003	Module in slot 3
SubslotNumber			W#16#0001	First subslot
ChannelNumber			W#16#8000	Diagnostics at submodule level
ChannelProperties			W#16#0800	In the binary system: 0000 1000 0000 0000 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
	ChannelProperties.Type (bit 0 to 7)		B#16#00	Set to 0 if ChannelNumber = W#16#8000
	ChannelProperties.Accumulative (bit 8)		0	No channel error group message
	MaintenanceRequired (bit 9) **MaintenanceDemanded (bit 10)** **ChannelProperties.Specifier (bit 11 to 12)**		0100	Diagnostics pending
	ChannelProperties.Direction (bit 13 to 15)		000	Vendor-specific
USI			W#16#8000	This DB is appended three data blocks: ChannelNumber, ChannelProperties and ChannelErrorType
Followed by channel diagnostics for faulty channel 0				
	ChannelNumber		W#16#0000	Channel 0
	ChannelProperties		W#16#2801	In the binary system: 0010 1000 0000 0001 ChannelProperties consists of (bit 0 to 7) ChannelProperties.Type (bit 8) ChannelProperties.Accumulative (bit 9) MaintenanceRequired (bit 10) MaintenanceDemanded (bit 11 to 12) ChannelProperties.Specifier (bit 13 to 15) ChannelProperties.Direction
		ChannelPropertiesType (bit 0 to 7)	B#16#01	1 bit
		ChannelProperties.Accumulative (bit 8)	0	No channel error group message
		MaintenanceRequired (bit 9) **MaintenanceDemanded (bit 10)** **ChannelProperties.Specifier (bit 11 to 12)**	0100	Diagnostics pending
		ChannelProperties.Direction (bit 13 to 15)	001	Input
	ChannelErrorType		W#16#0001	Short-circuit

6.4 Example of the configuration data record W#16#E000

Example of a configuration record W#16#E000

Configuration record W#16#E000 contains the target configuration of an IO device.

This example shows the configuration of 5 slots on an IO device.

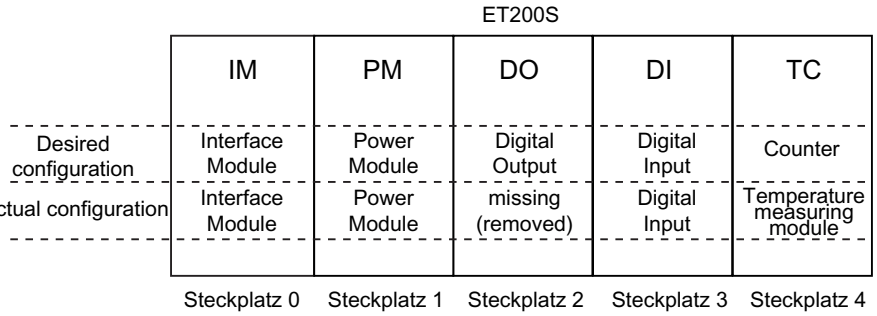

Figure 6-5 Configuration error in configuration records W#16#E000, W#16#E001 and W#16#E002

This gives the following configuration record:

Table 6-5 Example of a configuration record W#16#E000

DB name	Content	Comment
One configuration record is generated for each AR		
BlockHeader	BlockHeader consists of BlockType, BlockLength, BlockVersion	
BlockType	W#16#0012	ExpectedIdentificationDataBlock, that is, this is the diagnostics record for the expected configuration
BlockLength	W#16#0050	Decimal 80, that is, 80 bytes are appended to the BlockLength data block.
BlockVersion	W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
NumberOfAPIs	W#16#0001	An API is available
Data blocks generated for each API:		
API	DW#16#00000000	API=0, that is, no profile available
NumberOfSlots	W#16#0005	Five slots are configured for this device
Data blocks generated for each configured slot: The next five data blocks return information about slot 0		
SlotNumber	W#16#0000	Slot number = 0
ModuleIdentNumber	DW#16#00000322	Vendor-specific
NumberOfSubslots	W#16#0001	One configured submodule
Data blocks generated for each configured subslot:		
SubslotNumber	W#16#0001	Slot number = 1
SubmoduleIdentNumber	DW#16#00000000	Vendor-specific
The next five data blocks return information about slot 1		
SlotNumber	W#16#0001	Slot number = 1
ModuleIdentNumber	DW#16#00000684	Vendor-specific
NumberOfSubslots	W#16#0001	One configured submodule
Data blocks generated for each configured subslot:		
SubslotNumber	W#16#0001	Slot number = 1
SubmoduleIdentNumber	DW#16#00000000	Vendor-specific

DB name				Content	Comment
	The next five data blocks return information about slot 2				
		SlotNumber		W#16#0002	Slot number = 2
		ModuleIdentNumber		DW#16#000088a1	Vendor-specific
		NumberOfSubslots		W#16#0001	One configured submodule
		Data blocks generated for each configured subslot:			
			SubslotNumber	W#16#0001	Slot number = 1
			SubmoduleIdentNumber	DW#16#00000000	Vendor-specific
	The next five data blocks return information about slot 3				
		SlotNumber		W#16#0003	Slot number = 3
		ModuleIdentNumber		DW#16#00001094	Vendor-specific
		NumberOfSubslots		W#16#0001	One configured submodule
		Data blocks generated for each configured subslot:			
			SubslotNumber	W#16#0001	Slot number = 1
			SubmoduleIdentNumber	DW#16#00000000	Vendor-specific
	The next five data blocks return information about slot 4				
		SlotNumber		W#16#0004	Slot number = 4
		ModuleIdentNumber		DW#16#0000d6d8	Vendor-specific
		NumberOfSubslots		W#16#0001	One configured submodule
		Data blocks generated for each configured subslot:			
			SubslotNumber	W#16#0001	Slot number = 1
			SubmoduleIdentNumber	DW#16#00000000	Vendor-specific

6.5 Example of the configuration data record W#16#E001

Example of a configuration record W#16#E001

Configuration record W#16#E001 contains the actual configuration of an IO device. This example shows four slots, as the module has been removed from slot 2 (target configuration corresponds to five slots; see section 6.4).

Table 6-6 Example of a configuration record W#16#E001

DB name	Content	Comment
One configuration record is generated for each AR		
BlockHeader	BlockHeader consists of BlockType, BlockLength, BlockVersion	
BlockType	W#16#0013	RealIdIdentificationData, that is, this record returns the actual configuration
BlockLength	W#16#0042	Decimal 66, that is, 66 bytes are appended to the BlockLength data block.
BlockVersion	W#16#0101	BlockVersion 0101 is assigned to this diagnostics record
NumberOfAPIs	W#16#0001	An API is available
Data blocks generated for each API:		
API	DW#16#00000000	API=0, that is, no profile available
NumberOfSlots	W#16#0004	Four modules are physically available
Data blocks generated for each existing module: The next five data blocks return information about slot 0		
SlotNumber	W#16#0000	Slot number = 0
ModuleIdentNumber	DW#16#00000322	Vendor-specific
NumberOfSubslots	W#16#0001	One configured submodule
Data blocks generated for each configured subslot:		
SubslotNumber	W#16#0001	Subslot number = 1
SubmoduleIdentNumber	DW#16#00000000	Vendor-specific
The next five data blocks return information about slot 1		
SlotNumber	W#16#0001	Slot number = 1
ModuleIdentNumber	DW#16#00000684	Vendor-specific
NumberOfSubslots	W#16#0001	One configured submodule
Data blocks generated for each configured subslot:		
SubslotNumber	W#16#0001	Subslot number = 1
SubmoduleIdentNumber	DW#16#00000000	Vendor-specific
The next five data blocks return information about slot 3		
SlotNumber	W#16#0003	Slot number = 3
ModuleIdentNumber	DW#16#00001094	Vendor-specific
NumberOfSubslots	W#16#0001	One configured submodule
Data blocks generated for each configured subslot:		
SubslotNumber	W#16#0001	Subslot number = 1
SubmoduleIdentNumber	DW#16#00000000	Vendor-specific

DB name				Content	Comment
	The next five data blocks return information about slot 4				
		SlotNumber		W#16#0004	Slot number = 4
		ModuleIdentNumber		DW#16#000017FF	Vendor-specific
		NumberOfSubslots		W#16#0001	One configured submodule
		Data blocks generated for each configured subslot:			
			SubslotNumber	W#16#0001	Subslot number = 1
			SubmoduleIdentNumber	DW#16#00000000	Vendor-specific

6.6 Example of the configuration data record W#16#E002

Example of a configuration record W#16#E002

Configuration record W#16#E002 returns the difference between the target and actual configuration (see sections 6.1 and 6.2) of an IO device. The module is missing from slot 2, and an incorrect module has been inserted in slot 4.

Table 6-7 Example of a diagnostics record W#16#E002

DB name				Content	Comment
This configuration record returns the difference between the target and actual configuration					
BlockHeader				BlockHeader consists of BlockType, BlockLength, BlockVersion	
	BlockType			W#16#8104	ModuleDiffBlock, that is, this record returns the difference between the configured and diagnosed modules
	BlockLength			W#16#0026	Decimal 38, that is, 38 bytes are appended to the BlockLength data block.
	BlockVersion			W#16#0100	BlockVersion 0100 is assigned to this diagnostics record
NumberOfAPIs				W#16#0001	An API is available
Data blocks generated for each API:					
	API			DW#16#00000000	API=0, that is, no profile available
	Number of modules			W#16#0002	Two modules differ from the target configuration
	Data for incorrect module				
		SlotNumber		W#16#0002	Module in slot 2
		ModuleIdentNumber		DW#16#000088a1	ModuleIdentNumber of the missing module
		ModuleState		W#16#0000	Module missing
		NumberOfSubslots		W#16#0000	No submodule available as the module is not inserted
	Data for incorrect module				
		SlotNumber		W#16#0004	Module in slot 4
		ModuleIdentNumber		DW#16#000017ff	ModuleIdentNumber of the incorrect module

DB name				Content	Comment
		ModuleState		W#16#0001	Incorrect module
		NumberOfSubslots		W#16#0001	A submodule is available
		Data blocks generated for each configured subslot:			
			SubslotNumber	W#16#0001	Submodule at subslot 1
			SubmoduleIdentNumber	DW#16#00000000	Vendor-specific
			SubmoduleState	W#16#9000 In the binary system: 1001 0000 0000 0000 Bit 15 = 1 Bit 11 to 14 = 0010 Bit 7 to 10 = 0000 Bit 6 = 0 Bit 5= 0 Bit 4 = 0 Bit 3 = 0 Bit 0 to 2 = 000	 Format indicator is 1 incorrect module AR has a submodule No diagnostics data available No MaintenanceDemanded No MaintenanceRequest No graded extended channel diagnostics No meaning

Diagnostics for PROFINET IO

7

7.1 Contents - Diagnostics for PROFINET IO

Content of this section

This section covers:

- The most important differences between PROFINET IO and PROFIBUS DP in terms of diagnostics
- Operating principle of the diagnostics mechanism in PROFINET IO
- How to obtain detailed diagnostics information by calling system function blocks (SFBs) and system functions (SFCs) in the user program.

Additional information

For further information on diagnostics functions, refer to the STEP 7 Online Help system.

7.2 Diagnostics mechanism in PROFINET IO

Consistent diagnostics concept

PROFINET IO supports you with a consistent diagnostics concept. PROFINET provides IO diagnostics information about the error and interrupt states of a system in three different ways, as in PROFIBUS DP:

- Diagnostics using the STEP 7 configuration and engineering tool
- Diagnostics using status LEDs
- Diagnostics in the STEP 7 application program (AP)

The following sections describe the differences between PROFINET IO and PROFIBUS DP for all three diagnostics methods.

Diagnostics mechanism

The IO device outputs a diagnostics interrupt to the CPU when it detects a channel fault such as short-circuit at an analog input. The operating system responds by calling a diagnostics OB (for example, OB82). The local variables of this OB contain the logical base address of the faulty device, and include diagnostics data with a length of four bytes.

The IO controller automatically sets the corresponding parameters and configuration data of the new IO device or module inserted to replace a defective device/module. The new IO device must be assigned the configured name, for example, by reading it from the MMC of the former IO device.

Cyclic exchange of user data is restored in the next step.

7.3 Diagnostics using the STEP 7 configuration and engineering tool

Diagnostics in STEP 7

The diagnostics options provided in STEP 7 for PROFIBUS DP components are also available for PROFINET IO.

"Report system error"

The "Report system error" diagnostics function in STEP 7 can be used in PROFINET IO as in PROFIBUS DP.

"Report system error" is a convenient way to display diagnostics messages returned by the CPU of an IO device or IO controller.

STEP 7 automatically generates the necessary blocks and message texts. The user only needs to download the generated blocks to the CPU and transfer the texts to the connected HMI devices.

Information on configuring "Report system error"

For detailed information on the concept and configuration of "Report system error", refer to the Online Help of STEP 7 V5.3 SP2 or later.

Information on other diagnostics options

For detailed information on other diagnostics options, such as displaying the module status ("Module status"), hardware diagnostics ("Hardware diagnostics"), and identifying nodes ("Node flash test"), refer to the Online Help of STEP 7 V5.3 or later.

Tip: Locating faults at a failed IO device

You can no longer use the IO controller to access a failed IO device.

However, you can still the failed IO device in STEP 7 if a physical connection exists.

Select:

1. *Target system -> Display accessible nodes*.
2. The failed device.
3. Using the *Target system -> Download module status* command.
4. The *Diagnostics* tab.

When you have completed these steps, the program shows the IO device slot from which the faults occurred.

7.4 Diagnostics using Status LEDs

Introduction

Faults at the internal and external module errors are indicated by the front panel LED displays. For information on these LED displays and their evaluation, refer to the manual of the corresponding module of the SIMATIC family. Internal and external errors are combined on the S7-300 to create a group error.

7.4.1 Status LEDs with PROFINET IO and PROFIBUS DP

Comparison of diagnostics LEDs

The table below shows which LEDs are available for initial diagnostics and the meaning of these LEDs.

LED	Meaning in PROFINET IO	Meaning in PROFIBUS DP
BUSF	Lights up red: • Bus fault (no cable connection to a subnet/switch) • Wrong transmission speed • Full duplex transmission not activated	Lights up red: Bus fault at the interface (for example, bus short-circuit)
	Flashing: The PROFINET device is the controller: • Failure of a connected I/O device • At least one of the assigned IO devices cannot be addressed • Incorrect configuration The PROFINET device is an IO device: • Response monitoring timeout. • Communication on the PROFINET bus is down. • Incorrect IP address • Incorrect configuration • Incorrect parameter assignment • Incorrect or missing device name • IO controller not found / switched off, but there is an Ethernet connection.	Flashing: The module is the DP master: • Failure of a connected station • At least one of the configured slaves cannot be addressed. • Incorrect configuration The module is a DP slave: • Response monitoring timeout. • PROFIBUS DP communication is down. • Incorrect PROFIBUS address. • Incorrect configuration
RX	Lights up yellow: Receiving data at the interface. The LED flickers when small data volumes are transferred.	Not available

LED	Meaning in PROFINET IO	Meaning in PROFIBUS DP
TX	Lights up yellow: Sending data at the interface. The LED flickers when small data volumes are transferred.	Not available
LINK	Lights up green: Another device (usually a switch) is connected and the physical connection is up.	Not available
FO	Lights up yellow: Check the corresponding transmission link.	Not available

Note

The RX and TX LEDs can be combined into one LED, for example, as on the CPU 317-2 DP/PN or CP 343-1

Tip: Identification of the PROFINET device in the control cabinet

PROFINET IO devices must be assigned a device name when they are started for the first time. You can cause the LINK LED of a PROFINET IO device to flash when it needs to be named using the *PLC -> Ethernet > Assign Device Name* command in STEP 7 / HW Config. This function can be used to clearly identify a PROFINET IO device you want to address in the control cabinet.

7.5 Identification and Maintenance

Definition and properties

Identification and maintenance data (I&M) are saved to module memory in order to provide support when

- Checking the plant configuration
- Locating hardware changes in a plant
- Troubleshooting a plant

Identification data (I data) represents module information (some of which may be printed on the module housing) such as the order and serial number. I data is read-only vendor-specific module data.

Maintenance data (M data) returns system-specific information such as the installation location and date. M data is generated in the course of configuration and is written to the module memory.

The modules can be uniquely identified in online mode by means of the I&M data.

Devices which support I&M data

- **PROFIBUS DP**

  All PROFIBUS DPV1 slaves support I&M data. A definition of I&M data for PROFIBUS DPV0 is not available.

- **PROFINET IO**

  The PROFINET IO controllers and devices of the SIMATIC device family support I&M data.

New I&M functionality

The information and reporting functions are integral elements of S7 components, and are known by the term "module identification".

Access to the new I&M functionality is now standardized by the Profibus User Organisation. You can also process the I&M data of non-S7 components, as STEP 7 also supports this access.

Reading and writing I&M data in *STEP 7*

STEP 7 returns the I&M data at the "Module status" tab and in "Properties" dialog boxes of the module concerned (see the *STEP 7* Online Help.)

M data can be entered for the modules in HW Config (for example, in a dialog box in the course of configuration).

I&M data is accessed in accordance with the IEC 61158-6 standard.

The interface module from which I&M data is to be read must be available online in the H-System.

List of I&M data

The syntax of the I&M data conforms to the PROFINET Specification V2.1, "Application Layer services for decentralized periphery and distributed automation" and "Application Layer protocol for decentralized periphery and distributed automation".

Table 7-1 List of I&M data

I&M data	Explanation
MANUFACTURER_ID	The vendor name is saved to this parameter.
ORDER_ID	The order number of the module is saved to this parameter.
SERIAL_NUMBER	The serial number of the module is saved to this parameter. This structure allows the unique identification of the module.
HARDWARE_REVISION	The product version of the module is saved to this parameter. The product version is incremented when the module is modified.
SOFTWARE_REVISION	Returns information on the firmware version of the module
REVISION_COUNTER	Reserved
PROFILE_ID	Generic Device
PROFILE_SPECIFIC_TYPE	On interface modules
IM_VERSION	Returns the version of the I&M data.
IM_SUPPORTED	Returns information about the available I&M data.
TAG_FUNCTION	Enter a system-wide unique identifier for the module at this parameter.
TAG_LOCATION	Enter the installation location of the module at this parameter.
IM_DATE	Enter the module installation date and time at this parameter.
IM_DESCRIPTOR	Enter a module comment at this parameter.

Diagnosis in the STEP 7 user program

8

8.1 General information

Introduction

PROFINET IO supports diagnostics in the user program using system functions (SFCs), system function blocks (SFBs), and system status lists (SSLs), as in PROFIBUS DP.

The only differences are in the individual blocks selected for detailed error diagnostics. The table provides an overview of the features and status information in PROFINET IO and PROFIBUS DP.

Comparison of the diagnostics functions of PROFINET IO and PROFIBUS DP

Table 8-1 Comparison of the diagnostics functions of PROFINET IO and PROFIBUS DP

Feature	PROFINET IO	PROFIBUS DP
Content of the diagnostics data	Faulty components only	Depending on the implementation: Faulty components only or the full status information
The diagnostics status is available as...	Standardized channel errors in diagnostics records	Diagnostics message frames
Read the diagnostics status	Use SFC51 to read the SSLs in the user program and to localize errors. Use SFB52 to read diagnostics records and evaluate these in the user program. Details see below	Use SFC13 to read diagnostics message frames and evaluate these in the user program. **or** Use SFC51 to read the SSLs in the user program and to localize errors.
Read diagnostics data for error / interrupt in the error OB	Read with SFB54 and evaluate in the user program.	
Additional interrupt information in SFB54	Error information from the interrupt triggering location. That is, for example: The interrupt-triggering node reports **only** faulty channels.	Complete status of the interrupt-triggering node. That is, for example: The interrupt-triggering station reports the status of **all** channels.
Maximum number of record numbers	65535	255

Options for evaluating diagnostics data in the S7 user program

PROFINET IO specifies a global, vendor-independent structure for records which contain diagnostics information. diagnostics information is only generated for faulty channels. The following section describes two ways to evaluate the diagnostics data for a PROFINET device.

Table 8-2 Diagnostics using records

Number of diagnostics records	Type and scope of the diagnostics	Examples are available in ...
SFB52 (detailed example in the section "Diagnostics using SFB52 in OB1")	Error information for the faulty module	Section 8.2
OB82 and SFB54 (detailed example in the section "Diagnostics using SFB54 in OB82")	OB82 is called when interrupts have been generated. SFB54 contains detailed information on the error cause and location.	Section 8.3

Diagnostics using SFC51 "RDSYSST" and SFB54 "RALARM" and "Report system error"

Example applications and details of diagnostics operations in the user program are available on the Internet Application Portal of Automation and Drives, Service & Support. You can download this document directly at:

http://support.automation.siemens.com/WW/view/en/24000238

8.2 Diagnosis with the SFB 52 in the OB1

Description

Read the record assigned the number defined at the INDEX variable by calling SFB52 "RDREC" (read record).

Example INDEX = W#16#800A, for example, when you read the diagnostics data for diagnostics record W#16#800A.

You define the maximum number of bytes to read by setting the MLEN variable, so you should select a RECORD target range of at least the same length as defined in MLEN.

Output parameter VALID = TRUE indicates the successful transfer of the record to the target area RECORD. Output parameter LEN contains the length in [bytes] of the read data.

Output parameter ERROR reports any errors detected during record transfer. ERROR = TRUE and the error information is written to output parameter STATUS when an error is detected.

Operating principle

SFB52 "RDREC" operates in asynchronous mode, that is, processing involves several SFB calls. Start the record transfer by calling SFB52 with REQ=1.

Output parameter BUSY and bytes 2 and 3 of output parameter STATUS return the request status.

Output parameter is BUSY = false when the record has been successfully transferred. Parameter ERROR = false.

Task

An IO device, for example, ET 200 S, outputs a a diagnostics interrupt to the corresponding IO controller on account of a wire break at the output channel of the slot 2 module.

You then need to analyze the cause of the fault in the user program with reference to this diagnostics interrupt, and thus read the channel diagnostics data for the faulty (sub)module using diagnostics record W#16#800A.

This is done by reading SFB52 with index W#16#800A in the IO controller.

The next example describes how to program SFB52 and the meaning of the individual diagnostics data.

Example of how to read a diagnostics record

The diagnostics data from a digital output module of an ET 200S is described in the table "Meaning of the diagnostics data".

The 317-2 PN/DP IO controller with the IO device ET 200S is configured in PROFINET IO. The PROFINET IO configuration is the same as the PROFIBUS DP configuration.

The configuration is shown in the diagram.

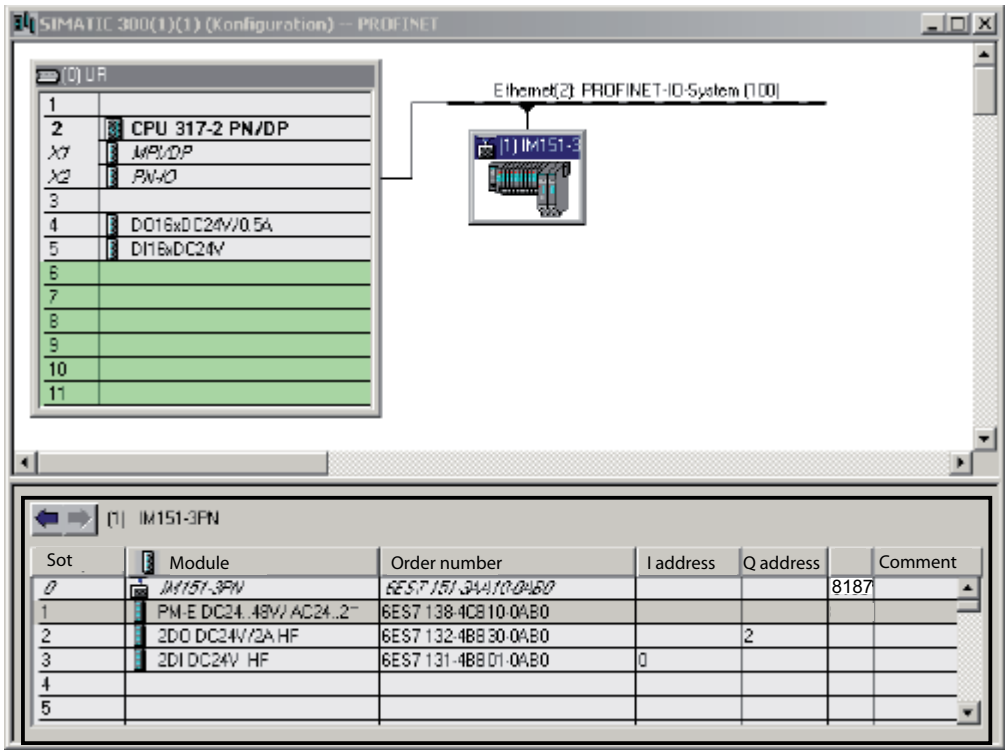

Figure 8-1 Configuration for Application Example Diagnostics

The table shows the configured modules in the PROFINET IO system: The IM 151-3 PN head module of the ET 200S is assigned diagnostics address 8187=W#16#1FFB (black-framed lower area of the diagram).

You can read a data record using SFB52 by addressing the digital output module of ET 200S device at its logical output address 2. This address must be defined at the parameter ID of SFB52. This is an output address, so bit 15 of the actual value specified here should also be set. In this case, the value DW#16#8002 must be defined as actual value at the parameter ID of SFB52.

Call SFB52 with CALL SFB52, DB52; in this example OB1. SFB52 can be called in any OB.

Meaning of the parameters

The table provides detailed information on the parameters in SFB52 to be assigned with variables and values.

Table 8-3 Parameters of SFB52 "RDREC"

Parameter	Declaration	Data type	Comment
REQ	INPUT	BOOL	REQ := 1: Transfer data
ID	INPUT	DWORD	Logical address of the PROFINET IO component (module or submodule). Bit 15 must be set when using an output module (example of address 5: ID:=DW#16#8005). You can define both an input and an output address if hybrid modules are used.
INDEX	INPUT	INT	Record number
MLEN	INPUT	INT	Maximum length in bytes of the record information to be read
VALID	OUTPUT	BOOL	New record has been received and is valid.
BUSY	OUTPUT	BOOL	BUSY = 1: The read operation has not yet ended.
ERROR	OUTPUT	BOOL	ERROR = 1: Read error
STATUS	OUTPUT	DWORD	Call identifier (bytes 2 and 3) or error code
LEN	OUTPUT	INT	Length of record information loaded
RECORD	IN_OUT	ANY	Target area for the read record

Note

Negative values

Negative values in the INDEX, MLEN and LEN parameters of SFB52 are interpreted as 16-bit unsigned integers.

The following table shows which variables and values are assigned to the parameters:

Table 8-4 STL code for reading diagnostics data

Command / parameter	Variable	Meaning
UN	M10.5	If the read operation has ended (BUSY flag = 0)...
UN	M10.6	and if no command is initiated to read the record (REQ = 0) ...
S	M10.6	then transferring the record (REQ = 1)
L	W#16#800A	Use diagnostics record W#16#800A
T	MW6	Download W#16#800A to flag word 6
CALL	SFB52, DB52	Call SFB52 with instance DB52
REQ :=	M10.6	Trigger memory bit
ID :=	DW#16#8002	DW#16#8002 is set as the logical output address (bit 15=1)
INDEX :=	MW6	Diagnostics record W#16#800A is loaded in MW6 to read the diagnostics data.
MLEN :=	50	Set a maximum length of 50 bytes for the record information to read
VALID :=	M10.4	Set the validity flag of the record in memory bit 10.4
BUSY :=	M10.5	Indicates whether the command is still running (BUSY = 1)
ERROR :=	M10.7	Set the error flag returned by a read error at memory bit M10.7.
STATUS :=	MD12	MD12 contains the error code
LEN :=	MW8	Memory word 8 contains the length of the record information that was read
RECORD :=	P#M 120.0 BYTE 100	ANY pointer to MB120 with a length of 100 bytes
U	M10.6	
R	M10.6	Reset memory bit 10.6

Structure of other records

An overview of essential diagnostics records is given in section 5: "Structure of the diagnostics records".

A full list of records is given in the "PROFINET IO - Application Layer Service Definition - Application Layer Protocol Specification" standard.

Members of the Profibus User Organisation (PNO) can download this standard at http://www.profibus.com.

Profile and structure of the diagnostics records

A PROFINET IO device consists of one or more "logical devices" which, in turn, may contain one or more APIs (**A**pplication **P**rocess **I**dentifier). Each PROFINET IO device supports at least one API.

The diagnostics records (0x800A, for example) can differ in terms of their structure. This difference is characterized by the BlockVersion. The API number was introduced, for example, in BlockVersion 0101 of records 0xX00A to allow diagnostics to be run on IO devices with several APIs.

The content and meaning of the profile ID has not changed compared to PROFIBUS DP.

The next two sections describe the structure of the diagnostics record with both versions: the diagnostics record with BlockVersion W#16#0100 and then with BlockVersion W#16#0101.

Evaluating the diagnostics record with BlockVersion W#16#0100

You can use the records shown in section 5 to work out the meaning of the various flag bytes used in this example.

Table 8-5 Meaning of the diagnostics data

Byte	Address	Content	Description
0	MB120	B#16#00	**BlockType**
1	MB121	B#16#10	W#16#0010: Diagnostics record type
2	MB122	B#16#00	**BlockLength**
3	MB123	B#16#12	W#16#0012 = 18: 18 bytes follow Record length = 22 bytes (18 + 2 bytes BlockType + 2 bytes BlockLength)
4	MB124	B#16#01	**BlockVersion**
5	MB125	B#16#00	W#16#0100: BlockVersion W#16#0100
6	MB126	B#16#00	**SlotNumber**
7	MB127	B#16#02	W#16#0002: Slot number of the interrupt-triggering component 2
8	MB128	B#16#00	**SubslotNumber**
9	MB129	B#16#01	W#16#0001: 1: Submodule slot number
10	MB130	B#16#80	**ChannelNumber**
11	MB131	B#16#00	W#16#8000: ID of the interrupt source: Submodule
12	MB132	B#16#08	**ChannelProperties**
13	MB133	B#16#00	W#16#0800 = 0000 1000 0000 0000: Bit 0 to 7: B#16#00: If ChannelNumber W#16#8000 Bit 8 = 0: No channel error group message Bit 9 to 10 = 00: Diagnostics event Bit 11 to 12 = 01: Diagnostics pending Bit 13 to 15 = 000: Vendor-specific
14	MB134	B#16#80	**USI**
15	MB135	B#16#00	USI = W#16#8000: Channel diagnostics record
16	MB136	B#16#00	**ChannelNumber**

Byte	Address	Content	Description
17	MB137	B#16#00	W#16#0000: Channel number of the interrupt-triggering component: 0
18	MB138	B#16#48	**ChannelProperties**
19	MB139	B#16#01	Bit 0 to 7: B#16#01 = 0000 0001: Data format: 1 bit Bit 8 to 15 = B#16#48: 01001000: Bit 8 = 0: No channel error group message Bit 9 to 10 = 00: Diagnostics event Bit 11 to 12: 01: Diagnostics pending Bit 13 to 15: 010 = 02: Output channel
20	MB140	B#16#00	**ChannelErrorType**
21	MB141	B#16#06	W#16#0006: Error type: Wire break

Evaluating the diagnostics record with BlockVersion W#16#0101

You can work out the meaning of the various flag bytes used in this example from the structure of the diagnostics records illustrated in section 5 and the table below.

Table 8-6 Meaning of the diagnostics data

Byte	Address	Content	Description
0	MB120	B#16#00	**BlockType**
1	MB121	B#16#10	W#16#0010: Diagnostics record type
2	MB122	B#16#00	**BlockLength**
3	MB123	B#16#16	W#16#0016 = 22: 22 bytes follow Record length = 26 bytes (22 + 2 bytes BlockType + 2 bytes BlockLength)
4	MB124	B#16#01	**BlockVersion**
5	MB125	B#16#01	W#16#0101: BlockVersion 0101
6	MB126	B#16#00	**API**
7	MB127	B#16#00	DW#16#00000000: without profile
8	MB128	B#16#00	
9	MB129	B#16#00	
10	MB130	B#16#00	**SlotNumber**
11	MB131	B#16#01	W#16#0001: Slot number of the interrupt-triggering component: 1
12	MB132	B#16#00	**SubslotNumber**
13	MB133	B#16#01	W#16#0001: 1: Submodule slot number
14	MB134	B#16#80	**ChannelNumber**
15	MB135	B#16#00	W#16#8000: ID of the interrupt source: Submodule
16	MB136	B#16#08	**ChannelProperties**

Byte	Address	Content	Description
17	MB137	B#16#00	W#16#0800 = 0000 1000 0000 0000: Bit 0 to 7: B#16#00: If ChannelNumber W#16#8000 Bit 8 = 0: No channel error group message Bit 9 to 10 = 00: Diagnostics event Bit 11 to 12 = 01: Diagnostics pending Bit 13 to 15 = 000: Vendor-specific
18	MB138	B#16#80	**USI**
19	MB139	B#16#00	USI = W#16#8000: Channel diagnostics record
20	MB140	B#16#00	**ChannelNumber**
21	MB141	B#16#00	W#16#0000: Channel number of the interrupt-triggering component: 0
22	MB142	B#16#48	**ChannelProperties**
23	MB143	B#16#01	Bit 0 to 7: B#16#01 = 0000 0001: Data format: 1 bit Bit 8 to 15: 01001000: Bit 8 = 0: No channel error group message Bit 9 to 10 = 00: Diagnostics event Bit 11 to 12: 01: Diagnostics pending Bit 13 to 15: 010 = 02: Output channel
24	MB144	B#16#00	**ChannelErrorType**
25	MB145	B#16#06	W#16#0006: Error type: Wire break

8.3 Diagnosis with the SFB54 in the OB82

Introduction

Signal and function modules which support diagnostics detect internal and external errors and generate a diagnostics interrupt to which can respond by calling an interrupt OB. The OB number and start information relating to the error event provide initial information on the cause and location of the error.

Detailed error event information can then be obtained by calling SFB54 (read additional interrupt information) in this error OB.

Note

STEP 7 Online Help

For detailed information on SFB54 and the data described in the tables below, refer to the STEP 7 Online Help.

Interrupt processing

The diagram below shows the individual steps for running diagnostics in the user program.

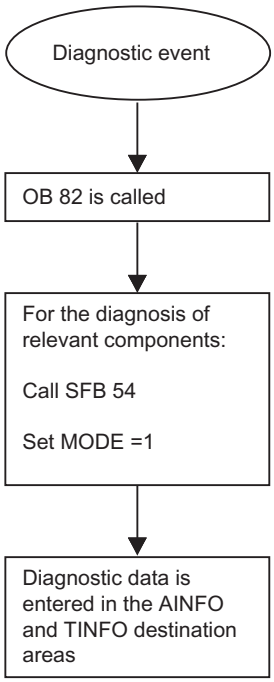

Figure 8-2 PROFINET IO diagnostics OB82 with SFB54

SFB54 function

SFB54 "RALARM" reads the interrupt data from all modules that support diagnostics, regardless of whether they are inserted in the central rack or are implemented in PROFIBUS DP or PROFINET IO.

This information is written to the output parameters STATUS, ID, LEN, TINFO, and AINFO. The OB start and management information can be found in the TINFO target area. The header and additional interrupt information (for example, in case of short-circuit) are located in AINFO target area.

Diagnostics by calling SFB54 in OB82

When a module that supports diagnostics detects an error, it outputs a diagnostics interrupt request to the CPU (for both incoming and outgoing events). This requires the diagnostics interrupt to have been enabled for the relevant module. The operating system calls OB82 in response to the diagnostics request.

The local variables of OB82 contain the logical base address and four bytes of diagnostics data about the faulty module. The CPU state changes to STOP if OB82 is not programmed.

Note

Disabling the interrupt OB

You can disable, enable or delay the diagnostics interrupt OB using SFCs 39 to 42.

You can also program SFB54 to save the diagnostics data to the AINFO and TINFO target areas.

The next section illustrates how to program diagnostics functions in SFB54.

Task

An IO device, for example, ET 200S, outputs a diagnostics interrupt to the corresponding controller on account of a wire break at the input channel of the slot 16 module. You want to call SFB54 in order to evaluate this diagnostics interrupt.

The following example describes how to program SFB54 and the meaning of the individual diagnostics data.

Programming an OB82 with SFB54

Requirements:

1. You have created and named a new STEP 7 project.
2. You have added a SIMATIC 300 station with a CPU 317-2 PN/DP.
3. You have configured an ET 200S on the PROFINET subnet of the CPU 317-2 PN/DP.

Carry out the following steps:

1. Generate an instance data block (IDB) to which you can save the interrupt data. Assign this IDB the name "IDB_SFB54".
2. Inserting OB82
   - Open your project in STEP 7
   - Select *Insert->S7 block->Organization block*, and then enter the block name "OB82" in the dialog box.
3. Start the LAD/STL/FBD editor
   - Double-click the "OB82" symbol to open the LAD/STL/FBD editor.
4. Declaring variables in SFB54
   - Enter "CALL SFB54, DB54" in the dialog box to call the SFB.
   - Then assign the values taken from the table below to the parameters of SFB54.

Table 8-7 Parameters of SFB54

Parameter	Declaration	Data type	Comment
MODE	IN	INT	MODE = • 0: Indicates the interrupt-triggering component ID and sets output parameter NEW = TRUE. • 1: Sets all the output parameters, irrespective of which component triggered the interrupt. • 2: Checks whether the component indicated in input parameter F_ID triggered the interrupt: - If not, NEW = FALSE - If yes, NEW = TRUE and all other output parameters will be set.
F_ID	IN	DWORD	Logical start address of the module from which interrupts should be received
MLEN	IN	INT	Maximum length in bytes of interrupt information to be received
NEW	OUT	BOOL	New = 1: A new interrupt was received

Parameter	Declaration	Data type	Comment
STATUS	OUT	DWORD	Error code of the SFB or IO controller
ID	OUT	DWORD	Logical start address of the component (module/submodule) from which an interrupt was received Bit 15 contains the I/O identifier: • 0: for an input address • 1: for an output address
LEN	OUT	INT	Length of the interrupt data received in bytes
TINFO	IN_OUT	ANY	Target area for OB start and management data
AINFO	IN_OUT	ANY	Target area for header and additional interrupt data Reserve a length of at least "MLEN" bytes for this parameter.

A detailed description of the individual parameters can be found in the STEP 7 Help.

Table 8-8 Parameter assignments in SFB54

Parameter	Variable	Meaning
CALL	SFB54, DB54	Calls SFB54 with instance DB54
MODE :=	1	All output parameters will be set independent of the interrupt-triggering component
F_ID :=		Does not have to be set as the F_ID is irrelevant in mode 1
MLEN :=	1500	Set a maximum length of 1500 bytes for the record information to be read
NEW :=	M1000.1	Memory bit 1000.1 = 1 if a new record was received
STATUS :=	MD10	MD10 contains the error code
ID :=	MD16	Memory doubleword 16 contains the logical start address of the module or submodule from which an interrupt was received.
LEN :=	MW24	Memory word 24 returns the length of the record information read
TINFO :=	P#M 500.0 BYTE 32	ANY pointer to MB500 with a length of 32 bytes
AINFO :=	P#M 1500.0 BYTE 1431	ANY pointer to MB1500 with a length of 1431 bytes

The information returned in the TINFO and AINFO variables are entered with a length of 32 or 1431 bytes, starting at flag bytes 500 or 1500. The data memory of the TINFO and AINFO target areas is not fully assigned, depending on which OB calls SFB54.

For detailed information, refer to the table in the context-sensitive help about blocks in the STEP 7 Online Help. To call up Help, press F1 or click on the "Help" button in the relevant dialog box.

TINFO variable table

The OB start and administration information is stored in tag table TINFO. You can read these data by calling SFB54 in OB82 as described earlier.

Table 8-9 Diagnostics data in TINFO

Byte	Address	Variable	Value	Description
0	MB500	OB_82_EV_CLASS	B#16#39	Start information OB82
1	MB501	OB_82_FLT_ID	B#16#42	
2	MB502	OB_82_PRIORITY	B#16#1A	
3	MB503	OB_82_OB_NUMBER	82 (decimal)	
4	MB504	OB_82_RESERVED_1	B#16#C5	
5	MB505	OB_82_IO_FLAG	B#16#54	
6	MW506	OB_82_MDL_ADDR	B#16#1FF6	
8	MB508	OB_82_DIAG_1	B#16#0D	
9	MB509	OB_82_MDL_TYPE	B#16#33	
10	MB510	OB_82_DIAG_2	B#16#00	
11	MB511	OB_82_DIAG_3	B#16#00	
12	MB512	Time stamp of the start information OB82	B#16#05	
13	MB513		B#16#03	
14	MB514		B#16#07	
15	MB515		B#16#11	
16	MB516		B#16#06	
17	MB517		B#16#06	
18	MB518		B#16#82	
19	MB519		B#16#22	
20	MW520	Geo address	B#16#8806	
22	MB522	Type of distributed device	B#16#08	Management information
23	MB523	Interrupt info type	B#16#00	
24	MB524	Flag PNIO controller	B#16#00	
25	MB525	EXT_DIAG_FLAG	B#16#01	
26	MW526	ID no. PNIO device	B#16#0301	
28	MW528	Vendor ID	B#16#002A	
30	MW530	Instance ID no.	B#16#0001	

Data in the TINFO target area

Table 8-10 Start information of OB82 (byte 0 to 19)

Byte	Address	Variable	Data type	Value	Description
0	MB500	OB82_EV_CLASS	BYTE	B#16#39	Event class and identifiers: Incoming event
1	MB501	OB82_FLT_ID	BYTE	B#16#42	Error code (B#16#42)
2	MB502	OB82_PRIORITY	BYTE	B#16#1A	Priority class; can be programmed in STEP 7 (HW Config). Organization blocks are processed in order of priority.
3	MB503	OB82_OB_Number	BYTE	82	OB number 82
4	MB504	OB82_RESERVED_1	BYTE	B#16#C5	Reserved
5	MB505	OB82_IO_FLAG	BYTE	B#16#54	Input module: B#16#54
6 and 7	MW 506	OB82_MD_ADR	WORD	W#16#1FF6	Logical base address of the faulty module: W#16#1FF6
8	MB508	OB_82_DIAG_1		B#16#0D	B#16#0D corresponds to 0000 1101 OB_82_DIAG_1 consists of the bits: Bit 0 = 1 OB_82_MDL_DEFECT Bit 1 = 0:OB_82_INT_FAULT Bit 2 = 1: OB_82_EXT_FAULT Bit 3 = 1: OB_82_ PNT_INFO Bit 4 = 0: OB_82_EXT_VOLTAGE Bit 5 = 0: OB_82_FLD_CONNCTR Bit 6 = 0: OB_82_NO_CONFIG Bit 7 = 0: OB_82_CONFIG_ERR
		OB_82_MDL_DEFECT	BOOL	TRUE	"Module fault"
		OB_82_INT_FAULT	BOOL	FALSE	No internal errors
		OB_82_EXT_FAULT	BOOL	TRUE	External error
		OB_82_PNT_INFO	BOOL	TRUE	Channel error
		OB_82_EXT_VOLTAGE	BOOL	FALSE	No "External auxiliary voltage missing" fault
		OB_82_FLD_CONNCTR	BOOL	FALSE	No "Front connector missing" fault
		OB_82_NO_CONFIG	BOOL	FALSE	No "Module not configured" error
		OB_82_CONFIG_ERR	BOOL	FALSE	No "Incorrect parameters in module" error
9	MB509	OB_82_MDL_TYPE	BYTE	B#16#33	B#16#33 corresponds to 0011 0011 Bit 0 to 3: 0011: Module class Bit 4 = 1: Channel information available Bit 5 = 1: User information available Bit 6 = 0: No diagnostics interrupt from proxy Bit 7 = 0: No maintenance requested

Byte	Address	Variable	Data type	Value	Description
10	MB510	OB_82_DIAG_2		B#16#00	B#16#00 corresponds to 0000 0000 OB_82_DIAG_2 consists of the bits: Bit 0 = 0: OB_82_SUB_MDL_ERR Bit 1 = 0: OB_82_COMM_FAULT Bit 2 = 0: OB_82_MDL_STOP Bit 3 = 0: OB_82_WTCH_DOG_FLT Bit 4 = 0: OB_82_INT_PS_FLT Bit 5 = 0: OB_82_PRIM_BAT_FLT Bit 6 = 0: OB_82_BCKUP_BATT_FLT Bit 7 = 0: No MaintenanceDemanded
		OB_82_SUB_MDL_ERR	BOOL	FALSE	No "Incorrect / missing user module" error
		OB_82_COMM_FAULT	BOOL	FALSE	No "Communication error"
		OB_82_MDL_STOP	BOOL	FALSE	Operating state: RUN (0: RUN; 1: STOP)
		OB_82_WTCH_DOG_FLT	BOOL	FALSE	No "Timeout error"
		OB_82_INT_PS_FLT	BOOL	FALSE	No "Internal module power supply failure"
		OB_82_PRIM_BAT_FLT	BOOL	FALSE	No "Backup battery low" fault
		OB_82_BCKUP_BATT_FLT	BOOL	FALSE	No "Backup failure" fault
		OB_82_RESERVED_2	BOOL	FALSE	Reserved
11	MB511	OB_82_DIAG_3		B#16#00	B#16#00 corresponds to 0000 0000 OB_82_DIAG_3 consists of the bits: Bit 0 = 0: OB82_RACK_FLT Bit 1 = 0: OB82_PROC_FLT Bit 2 = 0: OB82_EPROM_FLT Bit 3 = 0: OB82_RAM_FLT Bit 4 = 0: OB82_ADU_FLT Bit 5 = 0: OB82_FUSE_FLT Bit 6 = 0: OB82_HW_INTR_FLT Bit 7 = 0: OB_82_RESERVED_3
		OB82_RACK_FLT	BOOL	FALSE	No "Expansion rack failure"
		OB82_PROC_FLT	BOOL	FALSE	No "Processor failure"
		OB82_EPROM_FLT	BOOL	FALSE	No "EPROM fault"
		OB82_RAM_FLT	BOOL	FALSE	No "RAM fault"
		OB82_ADU_FLT	BOOL	FALSE	No ADC/DAC fault"
		OB82_FUSE_FLT	BOOL	FALSE	No "Fuse fault"
		OB82_HW_INTR_FLT	BOOL	FALSE	No "Process interrupt lost" error
		OB82_RESERVED_3	BOOL	FALSE	Reserved

Byte	Address	Variable	Data type	Value	Description
12 - 19	MB512 to MB519	OB82_DATE_TIME	DATE_ AND_ TIME	B#16#05	Date and time (BDC code) when the OB is called: Byte 12 to 14: Date: 07.03.2005 Year: 2005
				B#16#03	Month: 03
				B#16#07	Day: 07
				B#16#11	Byte 15 to 19: Time: 11:06:06 Hour: 11
				B#16#06	Minute: 06
				B#16#06	Second: 06
				B#16#82	Hundredth: 822 (B#16#822)
				B#16#2|**2**	Weekday: 2: Monday (B#16#X**2**)

Table 8-11 Structure of the geo address (byte 20 / 21)

Byte	Address	Variable	Data type	Value	Description
20 / 21	MW 520	Geo address of the interrupt source	WORD	W#16#8806	Record structure of the geo address in PROFINET IO (general): Bit: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 1 IO system ID Station number W#16#8806: Binary: 1000100000000110 Bit: 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 IO System ID Station number IO system ID = 1 returns the last two places of the PROFINET IO system ID (range of values = 0 to 15). Add add 100 (decimal) to this value in order to obtain the full PROFINET IO system ID; IO system ID in the example = 1+100 = 101 The station number equals 6

Table 8-12 OB81 management information (byte 22 to 25)

Byte	Address	Variable	Data type	Value	Description
22	MB522	Type of distributed device	BYTE	B#16#08	B#16#08 corresponds to 0000 1000 Type of distributed device: Bit 0 to 3: 1000: PROFINET IO; from 1001: reserved Bit 4 to 7: 0000: Profile type reserved
23	MB523	Interrupt info type	BYTE	B#16#00	Bit: 0 to 3: interrupt info type: 0000: Transparent, in PROFINET IO this is always the case (interrupt generated by a configured distributed module) Bit: 4 to 7 Structure version: 0000: Initial
24	MB524	Flags	BYTE	B#16#00	Flags of the PROFINET IO controller interface circuit Bit: 0 = 0: Interrupt from an integrated interface circuit Bit: 1 - 7: reserved
25	MB525	EXT_DIAG_FLAG	BYTE	B#16#01	B#16#01 corresponds to 0000 0001 Flags of the PROFINET IO controller interface circuit Bit 0 = 1: IO device fault Bit: 1- 7= 0000000: reserved

This concludes the management data for the TINFO target area in PROFIBUS and with a centralized configuration.

Table 8-13 Management data in PROFINET IO (byte 26 to 31)

Byte	Address	Data type	Value	Description
26 / 27	MB526 MB527	WORD	B#16#0301	PROFINET IO device ID number as a unique identifier
28 / 29	MB528 MB529	WORD	B#16#002A	Vendor ID
30 / 31	MB530 MB531	WORD	B#16#0001	ID number of the instance

Variable table AINFO (without MaintenanceRequest)

Tag table AINFO contains the BlockHeader and ID of the interrupt source, including the additional interrupt information returned when you call SFB54 in OB82.

This information is followed by diagnostics with or without maintenance data, depending on the availability of maintenance status data.

The following diagnostics record does not contain any maintenance information.

Table 8-14 Diagnostics data in AINFO

Byte	Address	Variable	Data type	Value
0 and 1	MW1500	BlockType	WORD	W#16#0002
2 and 3	MW1502	BlockLength	WORD	W#16#001E
4 and 5	MW1504	Version 0100	WORD	W#16#0100
6 and 7	MW1506	Interrupt type	WORD	W#16#0001
8 to 11	MD1508	API (Application Process Identifier) 0	DWORD	DW#16#00000000
12 and 13	MW1512	Slot	WORD	W#16#0010
14 and 15	MW 1514	Subslot	WORD	W#16#0001
16 to 19	MD1516	Module-ID	DWORD	DW#16#00008AD8
20 to 23	MD1520	Submodule-ID	DWORD	DW#16#00000000
24 and 25	MW 1524	Interrupt specifier	WORD	W#16#A854
26 and 27	MW 1526	Format-ID	WORD	W#16#8000
28 and 29	MW 1528	Channel number	WORD	W#16#0000
30 and 31	MW 1530	Info and data format	WORD	W#16#2805
32 and 33	MW 1532	Fault type	WORD	W#16#0006

Variable table AINFO (with MaintenanceRequest)

The following diagnostics record contains maintenance information. An additional DB containing additional interrupt information is generated This additional interrupt information has the Format-ID W#16#8100 and is only generated if maintenance is requested for the relevant submodule.

Table 8-15 Diagnostics data in AINFO

Byte	Address	Variable	Data type	Value
0 and 1	MW1500	BlockType	WORD	W#16#0002
2 and 3	MW1502	BlockLength	WORD	W#16#0032
4 and 5	MW1504	Version 0100	WORD	W#16#0100
6 and 7	MW1506	Interrupt type	WORD	W#16#000E
8 to 11	MD1508	API (Application Process Identifier) 0	DWORD	DW#16#00000000
12 and 13	MW1512	Slot	WORD	W#16#0000
14 and 15	MW 1514	Subslot	WORD	W#16#8001
16 to 19	MD1516	Module-ID	DWORD	DW#16#00000363
20 to 23	MD1520	Submodule-ID	DWORD	DW#16#00000001

Byte	Address	Variable	Data type	Value
24 and 25	MW 1524	Interrupt specifier	WORD	W#16#0004
26 and 27	MW 1526	Format-ID	WORD	W#16#8100
28 and 29	MW 1528	BlockType	WORD	W#16#0F00
30 and 31	MW 1530	BlockLength	WORD	W#16#0008
32 and 33	MW 1532	BlockVersion	WORD	W#16#0100
34 and 35	MW 1534	Reserved	WORD	W#16#0000
36 to 39	MW 1536	MaintenanceStatus Here: MaintenanceRequired	DWORD	W#16#00000001
40 and 41	MW 1540	Format-ID	WORD	W#16#8002
42 and 43	MW 1542	Channel number	WORD	W#16#8000
44 and 45	MW 1544	Info and data format	WORD	W#16#0A00
46 and 47	MW 1546	Fault type	WORD	W#16#8007
48 and 49	MW 1548	Additional error value	WORD	W#16#8000
50 to 53	MW 1550	Additional error information	WORD	W#16#00000010

Table 8-16 Data of the AINFO target area for interrupts without maintenance requests

Byte	Address	Variable	Value	Description
0	MB1500	reserved	B#16#00	Reserved
1	MB1501	BlockType	B#16#02 (least significant byte of BlockType)	Interrupt transfer channel 2
2 and 3	MW1502	BlockLength	W#16#001E	BlockLength: 30 successive bytes
4 and 5	MW1504	Block version	W#16#0100	Version: W#16#0100
6 and 7	MW1506	Interrupt type	W#16#0001	Interrupt type: Incoming diagnostics interrupt
8 to 11	MD1508	API	DW#16#00000000	API: 0 no profile
12 and 13	MW1512	Slot	W#16#0010	Slot number: 16
14 and 15	MW 1514	Subslot	W#16#0001	Submodule slot number of the interrupt-triggering component: 1
16 to 19	MD1516	Module-ID	DW#16#00008AD8	Module-ID; unique information about the interrupt source DW#16#00000001 to DW#32#FFFFFFFF: Vendor-specific
20 to 23	MD1520	Submodule-ID	DW#16#00000000	Submodule-ID; unique information about the interrupt source DW#16#00000000: assigned only to subslot 0

Byte	Address	Variable	Value	Description
24 and 25	MW 1524	Interrupt specifier	W#16#A854	Interrupt specifier / diagnostics status W#16#A854 corresponds to 1010100001010100 Bit 0 to 10: 00001010100: Sequence number is 84 Bit 11 = 1: Channel diagnostics available Bit 12 = 0: No vendor-specific status information available Bit 13 = 1: at least one channel diagnostics record available Bit 14 = 0: reserved Bit 15 = 1: at least one module configured within this AR is reporting diagnostics
26 and 27	MW 1526	Format-ID	W#16#8000	A channel diagnostics record follows bytes 26 and 27
28 and 29	MW 1528	Channel number	W#16#0000	Channel number: 0
30 and 31	MW 1530	Info and data format	W#16#2805 Corresponds in the binary system to: 0010100000000101	W#16#2805 corresponds to 00101000 00000101 Byte 31 (bit 0 to 7 in section 5.5.7): ChannelProperties.Type= B#16#05 Channel type (data format): Word
				Byte 30=B#16#28 corresponds to 00101000 ChannelProperties.Accumulative (bit 8) = 0: No channel error group message ChannelProperties.Qualifier (bits 9/10) = 00: Diagnostics ChannelProperties.Specifier (bits 11/12) = 01: Diagnostics pending ChannelProperties.Direction (bits 13 to 15)=001: Input
32 and 33	MW 1532	Fault type	W#16#0006	Fault type Wire break

Table 8-17 Data of the AINFO target area for interrupts with maintenance requests

Byte	Address	Variable	Value	Description
0 and 1	MW1500	BlockType	W#16#0002	Interrupt transfer channel 2
2 and 3	MW1502	BlockLength	W#16#0032	BlockLength: 50 successive bytes
4 and 5	MW1504	Block version	W#16#0100	Version: W#16#0100
6 and 7	MW1506	Interrupt type	W#16#000E	Interrupt type: Change to port status
8 to 11	MD1508	API	DW#16#00000000	API: 0 no profile
12 and 13	MW1512	Slot	W#16#0000	Slot number: 0000
14 and 15	MW 1514	Subslot	W#16#8001	Submodule slot number of the interrupt-triggering component: Port 1

Byte	Address	Variable	Value	Description
16 to 19	MD1516	Module-ID	DW#16#00000363	Module-ID; unique information about the interrupt source
20 to 23	MD1520	Submodule-ID	DW#16#00000001	Submodule-ID; unique information about the interrupt source DW#16#00000001: assigned only to subslot 1
24 and 25	MW 1524	Interrupt specifier	W#16#0004	Interrupt specifier / diagnostics status W#16#0004 corresponds to 0000000000000100 Bit 0 to 10: 00000000100: Sequence number is 4 Bit 11 = 0: No channel diagnostics available Bit 12 = 0: No vendor-specific status information available Bit 13 = 0: (at least) 0 channel diagnostics record available Bit 14 = 0: reserved Bit 15 = 0: No module configured within this AR is signaling a diagnostics state
26 and 27	MW 1526	Format-ID	W#16#8100	A maintenance diagnostics record follows bytes 26 and 27 (W#16#8100: USI for maintenance)
28 and 29	MW 1528	BlockType	W#16#0F00	BlockType: Maintenance diagnostics record
30 and 31	MW 1530	BlockLength	W#16#0008	BlockLength: 8 bytes
32 and 33	MW 1532	BlockVersion	W#16#0100	BlockVersion: W#16#0100
34 and 35	MW 1534	Reserved	W#16#0000	Reserved
36 to 39	MD1536	MaintenanceStatus Here: MaintenanceRequired	W#16#00000001	Maintenance diagnostics: MaintenanceRequest
40 and 41	MW 1540	Format-ID	W#16#8002	An extended channel diagnostics record follows bytes 40 and 41 (W#16#8002: USI for ExtChannelDiagnosis)
42 and 43	MW 1528	Channel number	W#16#8000	ID of the interrupt source: Submodule
44 and 45	MW 1530	Info and data format	W#16#0A00 Corresponds in the binary system to: 0000101000000000	W#16#0A00 corresponds to 0000101000000000 ChannelProperties.Type (bits 0 to 7)=00000000: Fixed value if ChannelNumber = W#16#8000 ChannelProperties.Accumulative (bit 8) = 0: No channel error group message ChannelProperties.Maintenance (bits 10/9) = 01: Diagnostics ChannelProperties.Specifier (bits 12/11) = 01: Incoming MaintenanceRequest ChannelProperties.Direction (bits 13 to 15) = 000: Vendor-specific

Byte	Address	Variable	Value	Description
46 and 47	MW 1532	Fault type	W#16#8007	Information about the quality of transmission on the fiber optic cable
48 and 49	MW 1548	Additional error information	W#16#8000	Information: System reserve
50 to 53	MW 1550	Additional error information	W#16#00000010	W#16#00000010 corresponds with 16dec The values are output at a resolution of 0.1 dB System reserve value: 16 x 0.1 dB = 1.6 dB

Note

The "channel number" to "error type" part may occur 0 to n times.

Migration for PC user programs

9

Content of this section

This section describes the changes that you will have to make to your PC programs when you migrate from PROFIBUS DP to PROFINET IO.

You have to distinguish between two scenarios:

- You previously used the DP base programming interface
- You used the OPC interface

The measures to be taken for migration are described for both scenarios in the next section.

9.1 Migration when operating with OPC Interface

Dynamic response

The dynamic response of PROFINET IO OPC Server is similar to that of PROFIBUS DP OPC Server, as the OPC interface is standardized.

OPC services

The handling of items for services such as read and write operations remains the same.

OPC item

An object of the "OPC Item " class represents an connection to a process variable. A process variable is an element of the address space of OPC Server, such as the input module of a programmable logic controller. An OPC Item is identified by its Item ID. The Item ID is a name specified by the server vendor and must be unique within the server address space. The value, quality, and time stamp properties are associated with each OPC Item. The quality of an OPC Item reveals whether the value of the variable can be determined reliably (for example, whether communication was up) and determines the significance of the OPC item value. The time stamp indicates the time at which the value of the process variable was determined.

Any data can be reached by means of an item, for example:

- Value of a sensor, for example, pressure, temperature, or flow volume
- Control parameters (for example, Start , Stop , Open and Close) .
- Status information (of a device, for example).
- Status of the network connection

Comparison of services

The call syntax for these services only differs in minor aspects. Migration for comparable services therefore only requires the items to be replaced.

Table 9-1 Comparison of services

Service	PROFIBUS DP OPC Server	PROFINET IO OPC Server
Read/write process data	**Example item** DP:[CP 5613]Slave005M003_EB10 **Explanation** Master CP 5613, slave 5, module 3, input byte 10	**Example item** PNIO:[CTRL3]EB10 **Explanation** Controller index 3, input byte address 10
Read/write records	**Example item** DP:[CP 5613]Slave005S003Data2,10,B7 **Explanation** Master CP 5613, slave 5, slot 3, index 2; record with a length of 10 bytes, starting at offset 7	**Example item** PNIO:[CTRL1]EDS10,DATA61450,10 **Explanation** Controller index 1, address 10, record index 61450, length 10 bytes
Determine / set information or control variable	**Example item** DP:[CP 5613]Masterstate **Explanation** Master CP 5613, operating state	**Example item** PNIO:[CTRL3]mode **Explanation** Controller index 3, operating state

9.2 Migration when operating with DP Base programming interface

Changes in the DP Base User Program

Before an existing DP Base user program can be used in PROFINET IO with the IO Base user programming interface, the following modifications are necessary:

Table 9-2 Changes in the DP Base User Program

Program Components	Convertibility
Addressing	Must be adapted.
Function calls	Must be modified.
Error codes	Must be adapted.
Event- or interrupt processing	Must be modified.
Headers and libraries	Must be replaced.
Start/stop routines	Must be modified.

Writing User Data (IO data) after Station Recovery or Insert Module Interrupt

In PROFIBUS DP, following a station recovery interrupt or an insert module interrupt, the last data written by the user and its status (GOOD or BAD) is transferred.

In PROFINET IO, if the user data has not already been written cyclically, you must make sure yourself that the user data is written again after a station recovery or insert module interrupt.

This data can, for example, be obtained by an initialization routine. As an alternative, the marked "old" data can be written again.

Reference

The following sections explain where adaptations are necessary in function calls and dynamic sequences.

9.2.1 Comparison of the Function Calls

Function calls

The table below compares the function calls of the DP Base and IO Base programming interfaces:

Table 9-3 Function calls

DP Base programming Interface	IO Base programming Interface
DP_alarm_ack	PNIO_alarm_resp
DP_close	PNIO_close
DP_delete_sema_object	n.a. [1]
DP_disable_event	n.a. [1]
DP_ds_read	PNIO_ds_read
DP_ds_write	PNIO_ds_write
DP_enable_event	n.a. [1]
DP_fast_logic_off	n.a.
DP_fast_logic_on	n.a.
DP_fetch_alarm	n.a. [1]
DP_get_actual_cfg	n.a.
DP_get_cref	n.a.
DP_get_err_txt	n.a.
DP_get_pointer	n.a.
DP_get_result	n.a.
DP_global_ctrl	n.a.
DP_init_sema_object	n.a.*
DP_open	PNIO_controller_open
DP_read_slv_par	n.a.
DP_release_pointer	n.a.

DP Base programming Interface	IO Base programming Interface
DP_reset_cp	n.a.
DP_set_mode	PNIO_set_mode
DP_slv_state	PNIO_device_activate
DP_start_cp	n.a.
DP_watchdog	n.a.
DP_write_trc	n.a.

[1] Mapped to callback mechanism.

9.2.2 Comparison of the Dynamic Calls

Hardware Interrupts

Fast Logic is not supported.

Process Image

The difference between a DP Base and IO Base user program when accessing the process image is in the addressing and the access mechanism.

Table 9-4 Access to the Process Image

Mechanisms	DP Base user program	IO Base user program
Addressing	Access using the PROFIBUS station number.	Access using addresses assigned during configuration.
Access Mechanism	1. Lock dual-port RAM. 2. Access with pointer to dual-port RAM. 3. Dual-port RAM release.	Access to the IO controller using the function calls: • PNIO_data_read • PNIO_data_write

CPs for PROFINET IO

10

10.1 Content

Content of this section

If you use certain communication processors (CPs) for Industrial Ethernet, you will add the functionality of a PROFINET IO controller to your S7 station.

- For SIMATIC S7-300: CP 343-1 (6GK7 343-1EX21-0XE0, 6GK7 343-1GX21-0XE0)
- For SIMATIC S7-400: CP 443-1 Advanced (6GK7 443-1EX40-0XE0)

This section provides an overview of the interface to PROFINET IO required when using these CPs.

10.2 CP 343-1

Application

The CP 343-1 communications processor is designed for operation in an S7-300 AS. It allows you to connect the S7-300 to Industrial Ethernet.

The CP 343-1 services for PROFINET IO provide direct access to PROFINET IO devices on Industrial Ethernet

FCs in the user program

Special FCs are available for the CPs for PROFIBUS DP and for the operation of PROFINET IO:

- FC9 (PNIO_SEND)
- FC10 (PNIO_RECV)
- FC11 (PNIO_ADDR)

Additional information

For detailed information on the functions, refer to the *S7-CPs for Industrial Ethernet - Configuration and Commissioning* manual and to the STEP 7 Online Help system.

For information on CP applications and details of the quantity framework at the PROFINET IO interface, refer to the documentation included with each CP on the *SIMATIC NET Manual Collection CD*.

10.3 CP 443-1 Advanced

Application

The CP 443-1 Advanced communication processor is designed for use in an S7-400 AS (not in H systems). It allows an S7-400 AS to be connected to Industrial Ethernet.

The CP 443-1 Advanced provides services for PROFINET IO which allow direct access to PROFINET IO devices on Industrial Ethernet.

SFBs and SFCs for CP 443-1 Advanced in the user program

Use with PROFINET IO does not require any special FBs or FCs. Distributed I/Os are interconnected directly using the SFCs/SFBs of the CPU:

- SFB 52 (RDREC)
- SFB 53 (WRREC)
- SFB 54 (RALRM)
- SCF14 (DPRD_DAT)
- SCF15 (WRRD_DAT)
- SFC 49 (LGC_GADR)
- SFC 51 (RD_SZL)

Further information

For detailed information on blocks / functions, refer to the *System Software for S7-300/400 System and Standard Functions* manual and to the STEP 7 Online Help.

For information on using the CP and details of the volume of data at the PROFINET IO interface, refer to the documentation included with each CP on the *SIMATIC NET Manual Collection CD*.

10.4 CP 1616

Application

The CP 1616 communication processor is a PCI module and is used to connect PCs and SIMATIC PGs/PCs to PROFINET IO.

The PROFINET IO services of the CP 1616 allow it to be used as an IO controller and/or IO device.

Features

Most important features: The properties of Ethernet CP 1616 are tailored to state-of-the-art solutions for industry. For example:

- Optimized for PROFINET IO
- Integrated 4-port real-time switch for implementing star and bus topologies
- Support for real-time communication
- Support of isochronous real-time communication
- Development kit for integrating the CP 1616 on any operating system platform
- Comprehensive diagnostics options

Further information

For detailed information on the functions, refer to the *S7-CPs for Industrial Ethernet - Configuration and Commissioning* and *Commissioning PC Stations - Introduction and Getting Started* Manuals.

Glossary

10 base T/F

Ethernet standard; supports transmission rates up to 10 Mbps.

100 base T/F

Ethernet standard; supports transmission rates up to 100 Mbps.

1000 base T/F

Ethernet standard; supports transmission rates up to 1000 Mbps.

API

Application **P**rocess **I**dentifier. Term used in PROFINET IO Standard IEC 61158; its value specifies the application which is used to process IO data.

The IEC assigns the APIs specific profiles which are defined in context by the PROFINET User Organisation. The standard API is 0.

The value of the API (Application Process Identifier) parameter specifies the application that is processing the IO data.

PROFINET standard IEC 61158 assigns profiles to certain APIs (PROFIdrive, PROFIslave) which are defined by the PROFINET User Organisation.

The standard API is 0.

Application

An application is a program that runs directly on the MS-DOS / Windows operating system. Applications on the PG include, for example, the STEP 5 basic package, GRAPH 5 and others.

→ *User program*

CAT 3

Twisted-pair cable comes in different versions. Several versions are specified in the Ethernet standard.

There are several categories, but only CAT 3 and CAT 5 are relevant to networks. The two types of cable differ in terms of their maximum permitted frequency and the attenuation values (weakening of signals across a specific distance).

CAT 3 is a twisted-pair cable for 10 base T Ethernet.

CAT 5 is a twisted-pair cable for 100 base T Fast Ethernet.

CAT 5

→ *CAT 3*

Category 3

→ *CAT 3*

Category 5

→ *CAT 3*

Coaxial cable

A coaxial cable, also known as "coax", is a metal conductor system used in HF transmission circuits, for example, as radio and TV antenna cable, and in modern networks demanding high data transmission rates. The inner conductor of a coaxial cable is sheathed by a tube-like outer conductor. These conductors are separated by plastic insulation. In contrast to other cables, this type of cable provides a high degree of immunity to interference and EMC compatibility.

COM

Component **O**bject **M**odel. Microsoft specification for Windows objects based on OLE.

Automation systems are mapped onto objects in PROFINET CBA. An object consists of interfaces and properties. Two objects can communicate on the basis of these interfaces and properties.

→ *DCOM*

Communication processor

Communication processors are modules used for point-to-point and bus topologies.

Consistent data

Data which are related in their contents and not to be separated are referred to as consistent data.

For example, the values of analog modules must always be handled as a whole, that is, the value of an analog module must not be corrupted as a result of read access at two different points of time.

CP

→ *Communication processor*

CPU

Central processing unit = CPU of the S7 automation system with a control and arithmetic unit, memory, operating system, and interface for programming device.

→ *CPU*

DCOM

Distributed **COM**. Enhanced COM standard for remote object communication across device boundaries. DCOM is based on the RPC protocol, which in turn is based on TCP/IP. PROFINET CBA devices deploy the DCOM technology to exchange data which are not time-sensitive, such as process data, diagnostics data and parameter data.

PROFINET V1.0 or later supports DCOM technology.

Members of the Profinet User Organisation (PNO) can request a portable DCOM protocol stack which is tailored to PROFINET applications. This prevents any dependency on Microsoft and (further) developments of this technology, while maintaining compatibility with Microsoft solutions.

→ *COM*

Diagnostics

→ *System diagnostics*

ERTEC

ERTEC = "Enhanced Real Time Ethernet Controller"

The new ERTEC200 and ERTEC400 ASICs are designed for automation applications. They support the PROFINET protocol and are required for IRT operation. Siemens AG offers these PROFINET ASICs for the development of user-specific devices, as PROFINET is an open standard. ASIC is the acronym for Application Specific Integrated Circuits. PROFINET ASICs are components which provide a wide range of functions for developing user-specific devices. They convert the requirements of the PROFINET standard into high-performance, high-density circuits.

Benefits of ERTEC:

- Easy integration of switch functionality in devices
- Easy and cost-effective setup of bus structures
- Minimized communication load on devices

Fast Ethernet

→ *100 base T/F*

FB

→ *Function block*

FC

→ *Function*

Function

According to IEC 1131-3, a function (FC) is a code block without static data. A function allows transfer of parameters in user program. Functions are therefore suitable for programming frequently occurring complex functions, e.g. calculations.

Function block

According to IEC 1131-3, a function block (FB) is a code block with static data. An FB allows the user program to pass parameters. Function blocks are therefore suitable for programming frequently occurring complex functions, e.g. controls, mode selections.

GSD file

The properties of a PROFINET device are described in a GSD file (General Station Description) that contains all the information required for configuration.

Similar to PROFIBUS, you can implement a PROFINET device in STEP 7 by installing a GSD file.

In PROFINET IO, the GSD file is in XML format. The structure of the GSD file is compliant with ISO 15734, which is the world-wide standard for device descriptions.

In PROFIBUS, the GSD file is in ASCII format.

Hub

In contrast to a switch, a hub automatically adapts itself to the lowest transmission rate of the ports and transfers the signals to all connected devices. A hub cannot prioritize signals. This creates a very high communication load on the Industrial Ethernet.

Industrial Ethernet

→ *100 base T/F*

Interface, MPI-compatible

→ *MPI*

IO controller

→ *PROFINET IO Controller*

→ *PROFINET IO Device*

→ *PROFINET IO Supervisor*

→ *PROFINET IO System*

IO device

→ *PROFINET IO Controller*

→ *PROFINET IO Device*

→ *PROFINET IO Supervisor*

→ *PROFINET IO System*

IO supervisor

→ *PROFINET IO Controller*

→ *PROFINET IO Device*

→ *PROFINET IO Supervisor*

→ *PROFINET IO System*

IO system

→ *PROFINET IO System*

LAN

Local Area Network; interconnects multiple computers within a company. The geographical topology of a LAN is limited to the local premises and is only available to the operating company or institution.

MaintenanceDemanded

Continuity and reliability of a PROFINET device depends on the early recognition and elimination of potential faults in order to prevent any loss of production.

This requires the definition of different maintenance information relating to MaintenanceDemanded.

A "MaintenanceDemanded" system alarm can be defined for various wear-and-tear parameters. For example, the alarm may recommend inspection of a component which has reached a specific number of operating hours.

A MaintenanceDemanded alarm is generated if a component must be replaced within a short period.

Example of a printer:

A MaintenanceDemanded alarm is generated to indicate that the toner / printer cartridge requires immediate replacement.

MaintenanceRequested

Continuity and reliability of a PROFINET device depends on the early recognition and elimination of potential faults in order to prevent any loss of production.

This requires the definition of different maintenance information relating to the MaintenanceRequest.

A "MaintenanceRequest" system alarm can be defined for various wear-and-tear parameters. For example, the alarm may recommend inspection of a component which has reached a specific number of operating hours.

A MaintenanceRequest alarm is generated if a component must be replaced within a conceivable period.

Example of a printer:

A MaintenanceRequest alarm is generated to indicate that the toner / printer cartridge will have to be replaced in a few days.

Master

When a master is in possession of the token, it can send data to other nodes and request data from other nodes (= active node).

→ *Slave*

MPI

The multipoint interface (MPI) represents the programming device interface of SIMATIC S7. It enables multiple nodes (PGs, text-based displays, OPs) to be operated simultaneously by one or more CPUs. Each node is identified by its unique (MPI) address.

MPI address

→ *MPI*

Network

A network consists of one or more interconnected subnets with any number of nodes. Several networks can exist alongside each other.

OB

→ *Organization blocks*

OLE

Object Linking and Embedding is a central architectural principle in Windows. OLE is a Microsoft technology which enables object linking and data exchange between programs.

OPC

OLE for Process Control is an industrial standard which defines vendor-independent access to industrial communication networks on the basis of OLE.

OPC (OLE for Process Control) defines a standard communication interface for automation technology. OPC enables access to OLE (Object Linking and Embedding). OLE is a component model of Microsoft. Components are software objects or applications which make their functionality available to other applications.

Communication via the OPC interface is based on COM/DCOM. In this case, the object is the process image.

The OPC interface was designed as an industrial standard by leading companies in the automation industry with support from Microsoft Corporation. Applications accessing process data were previously restricted to the access mechanisms of the communications network of a single vendor. The standardized OPC interface harmonizes access to the communication networks of any vendor.

→ *OPC Client*

→ *OPC Server*

OPC Client

An OPC client is a user program which accesses process data using the OPC interface. OPC Server provides access to process data.

→ *OPC*

→ *OPC Server*

OPC item

An object of the "OPC Item " class represents an connection to a process variable. A process variable is an element of the address space of OPC Server, such as the input module of a programmable logic controller. An OPC Item is identified by its Item ID. The Item ID is a name specified by the server vendor and must be unique within the server address space. The value, quality, and time stamp properties are associated with each OPC Item. The quality of an OPC Item reveals whether the value of the variable can be determined reliably (for example, whether communication was up) and determines the significance of the OPC item value. The time stamp indicates the time at which the value of the process variable was determined.

Any data can be reached by means of an item, for example:

- Value of a sensor, for example, pressure, temperature, or flow volume
- Control parameters (for example, Start , Stop , Open and Close) .
- Status information (of a device, for example).
- Status of the network connection

OPC Server

The OPC server provides a wide range of functions to an OPC client for communication over industrial networks.

For further information, refer to the *Industrial Communication with PG/PC* manual.

→ *OPC*

→ *OPC Client*

Operating system

The CPU OS organizes all functions and processes of the CPU which are not associated to a specific control task.

→ *CPU*

Organization blocks

Organization blocks (OBs) form the interface between CPU operating system and the user program. The sequence for user program execution is determined in the organization blocks.

PCD

The PROFINET component description is the description of the components you have generated in your engineering system (for example, STEP 7). The PCD is an XML file that you can import into SIMATIC iMap so that you can configure the PROFINET CBA communication.

PG

→ *Programming device*

PLC

Programmable controllers (PLCs) are electronic controllers whose function is saved as a program in the control unit. Therefore, the configuration and wiring of the unit does not dependend on the PLC function. A programmable logic controller has the structure of a computer; it consists of a CPU with memory, input/output modules and an internal bus system. The I/O and the programming language are oriented to control engineering needs.

→ *CPU*

→ *PLC*

PNO

→ *PROFIBUS International*

PROFIBUS

Process Field Bus - European fieldbus standard.

→ *PROFIBUS DP*

→ *PROFIBUS International*

PROFIBUS DP

→ *PROFIBUS*

PROFIBUS DP

A PROFIBUS with the DP protocol that complies with EN 50170. DP stands for distributed peripheral I/O (fast, real-time, cyclic data exchange). From the perspective of the user program, the distributed I/O is addressed in exactly the same way as the central I/O.

→ *PROFIBUS International*

PROFIBUS International

Technical committee that defines and further develops the PROFIBUS and PROFINET standard.

Also known as the PROFIBUS User Organization (PNO).

Home page http://www.profibus.com

PROFINET

→ *PROFIBUS International*

PROFINET ASIC

Refer to ERTEC

PROFINET Component Description

→ *PCD*

PROFINET IO Controller

Device via which the connected IO devices are addressed. This means that the IO controller exchanges input and output signals with assigned field devices. The IO controller is often the controller on which the automation program runs.

→ *PROFINET IO Device*

→ *PROFINET IO Supervisor*

→ *PROFINET IO System*

PROFINET IO Device

Distributed field device assigned to one of the IO controllers (for example, remote I/O, valve terminal, frequency converter, switches)

→ *PROFINET IO Controller*

→ *PROFINET IO Supervisor*

→ *PROFINET IO System*

PROFINET IO Supervisor

PG/PC or HMI device for commissioning and diagnostics.

→ *PROFINET IO Controller*

→ *PROFINET IO Device*

→ *PROFINET IO System*

PROFINET IO System

PROFINET IO controller with assigned PROFINET IO devices.

→ *PROFINET IO Controller*

→ *PROFINET IO Device*

Programming device

Basically speaking, PGs are compact and portable PCs which are suitable for industrial applications. They are identified by a special hardware and software for programmable logic controllers.

Real-time communication

Industrial communication in which supervisors take part in communication involves communication runtimes which are too long for production automation. PROFINET uses its own real-time channel instead of TCP/IP to communicate time-sensitive IO user data.

Repeater

→ *Hub*

Router

A router interconnects two subnets. A router works in a similar way to a switch. You can also enable/disable nodes for communication at the router. The communication nodes on various sides of a router can only communicate with one another if you have explicitly enabled communication between these nodes via the router. Real-time data cannot be exchanged beyond subnet boundaries.

SELV/PELV

The term denotes circuits with safety extra-low voltage.

Siemens SITOP power supplies provide this protection, for example.

For further information, refer to the EN 60950-1 (2001) standard.

SFB

→ *System function block*

SFC

→ *System function*

Signal module

Signal modules (SM) form the interface between the process and the PLC. There are digital input and output modules (input/output module, digital) and analog input and output modules (input/output module, analog).

SIMATIC

The term denotes Siemens products and systems for industrial automation.

SIMATIC iMap

Engineering tool for configuration, commissioning, and monitoring of modular distributed automation systems. It is based on the PROFINET standard.

SIMATIC NET

Siemens Industrial Communication division for Networks and Network Components.

Slave

A slave can only exchange data after being requested to by the master.

→ *Master*

STEP 7

Engineering system. Contains programming software for the creation of user programs for SIMATIC S7 controllers.

System diagnostics

System diagnostics refers to the detection, evaluation, and signaling of errors that occur within the PLC, for example programming errors or module failures. System errors can be indicated by LEDs or in **STEP 7**.

System function

A system function (SFC) is a function integrated in the operating system of the CPU that can be called when necessary in the STEP 7 user program.

System function block

System function blocks (SFB) are integrated in the CPU operating system and can be called in the STEP 7 user program.

System status list

The system status list contains data that describe the current status of an S7-300 or S7-400. You can always use this list to obtain an overview of:

- The configuration of the S7-300
- the current CPU configuration and configurable signal modules
- the current status and processes in the CPU and in configurable signal modules.

TCP/IP

The Ethernet system is designed solely to carry data. It is comparable to a highway as a system for transporting goods and passengers. The data is actually transported by protocols. This is comparable to cars and commercial vehicles transporting passengers and goods on the highway.

Tasks handled by the basic Transmission Control Protocol (TCP) and Internet Protocol (IP) (abbreviated to TCP/IP):

1. The sender splits the data into a sequence of packets.
2. The packets are transported over the Ethernet to the correct recipient.
3. The recipient reassembles the data packets in the correct order.
4. Faulty packets are sent again until the recipient acknowledges that they have been transferred successfully.

Most higher-level protocols use TCP/IP to handle their tasks. The Hyper Text Transfer Protocol (HTTP), for example, transfers documents on the World Wide Web (WWW) which are written in Hyper Text Markup Language (HTML). Without this technology, you would not be able to view Web sites on your Internet browser.

Token

Allows access to the bus for a limited time.

Topology

Network structure. Commonly used structures:

- Bus topology
- Ring topology
- Star topology
- Tree topology

Twisted-pair

Fast Ethernet over twisted-pair cables is based on the IEEE 802.3u standard (100 base TX). The transmission medium is a shielded 2x2 twisted-pair cable with an impedance of 100 Ohm (AWG 22). The transmission characteristics of this cable must meet the requirements of category 5 (see glossary).

The maximum length of the connection between the terminal and the network component must not exceed 100 m. The cables are wired according to the 100 base TX standard using the RJ45 connector system.

User program

In SIMATIC, a distinction is made between the operating system of the CPU and user programs. The user program contains all instructions, declarations and data for signal processing required to control a plant or a process. It is assigned to a programmable module (for example CPU or FM) and can be structured in smaller units (blocks).

→ *Operating system*

→ *STEP 7*

WAN

Wide Area Network. Network beyond LAN boundaries which allows, for example, intercontinental communication. Legal rights do not belong to the user but to the provider of the communication network.

XML

XML (Extensible Markup Language) is a flexible, easy to understand data description language which can be learned easily. Information is exchanged with the aid of readable XML documents. These documents contain continuous text supplemented with structure data.

Index

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

SIEMENS

SIMATIC

Product Information

07/2008

For the programming manual from PROFIBUS DP to PROFINET IO, Edition 10/2006, A5E00298268-03

This product information contains **important information on new and revised diagnostics data records.** The Product Information is part of the supplied product and the statements made therein are binding in case of doubt.

Siemens Aktiengesellschaft A5E01648459-01

For chapter 5 "Data records for PROFINET IO"

Overview of relevant data records for PROFINET IO

The following information is an addendum to the manual, section *5.1.2* .

Data records on the status of PROFINET interfaces

Data record number	Content and meaning	Size in bytes
W#16#802A	The data record supplies the latest port settings.	0 - 4176
W#16#802B / W#16#802F	The data record supplies the configured port settings.	0 - 4176
W#16#802D	The data record supplies the configured settings for the synchronization.	0 - 4176
W#16#8060	The data record supplies the latest settings of the optical port.	0 - 4176
W#16#8061 / W#16#8061	The data record supplies the configured settings of the optical port.	0 - 4176
W#16#8070	The data record supplies the configured settings of the PROFINET interface.	0 - 4176
W#16#8080	The data record supplies the latest settings of the PROFINET interface.	0 - 4176
W#16#F831	The data record supplies the collective data record for the configured settings of the PROFINET interface and its ports (settings of the IRT parameters only).	0 - 4176
W#16#F841	The data record supplies the collective data record for latest settings of the PROFINET interface and its ports.	0 - 4176
W#16#F842	The data record supplies the collective data record for the configured settings of the PROFINET interface and its ports.	0 - 4176

Additional information

Detailed information on data records are available in the PROFINET specification "Application Layer services for decentralized periphery and distributed automation" and "Application Layer protocol for decentralized periphery and distributed automation" in version V2.2 on the Internet pages of the PROFIBUS user organization under www.profinet.com.

Blocks of diagnostics and configuration data records

The following information is an addendum to the manual, section *5.5.9*, table 5-23.

Coding of ExtChannelErrorType for ChannelErrorType "Incorrect neighborhood"

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 – W#16#7FFF	Manufacturer-specific
W#16#8000	Incorrect neighbor – device
W#16#8001	Incorrect neighbor – port
W#16#8002	Neighbor does not support RealTime Class 3 or is not configured
W#16#8003	Error due to incorrect interface setting – full duplex and half duplex
W#16#8004	Incorrect or missing media redundancy configuration
W#16#8005	No neighbor present
W#16#8006	Neighbor does not support smooth media redundancy
W#16#8007	Error due to difference in determining cable length
W#16#8008	Same Sync domain name used in several instances / several Sync masters exist
W#16#8009 – W#16#8FFF	Reserved
W#16#9000 – W#16#9FFF	Reserved for profiles
W#16#A000 – W#16#FFFF	Reserved

Coding of ExtChannelErrorType for ChannelErrorType "Synchronization error" and "Error time basis"; table 5-25 in manual

Value (hexadecimal)	Meaning
W#16#0000	Reserved
W#16#0001 – W#16#7FFF	Manufacturer-specific
W#16#8000	Sync slave no longer synchronized
W#16#8001/ W#16#8002	Reserved
W#16#8003	Synchronization error
W#16#8004 – W#16#8FFF	Reserved
W#16#9000 – W#16#9FFF	Reserved for profiles
W#16#A000 – W#16#FFFF	Reserved

User Structure Identifier

The following information is an addendum to the manual, section *5.5.16*, table 5-42 in the manual.

Coding of USI (UserStructureIdentifier)

Value (hexadecimal)	Meaning	Description
W#16#0000 - W#16#7FFF	Manufacturer-specific	In connection with Alarm Type diagnostics incoming/going manufacturer-specific Diagnostics in AlarmNotification and Diagnosis Data. In connection with other Alarm Types, the use is manufacturer-specific.
W#16#8000	ChannelDiagnosis	Use in connection with ChannelDiagnosis in AlarmNotification and Diagnosis Data.
W#16#8001	Multiple	Only in connection with data that correspond to the structure of "(BlockHeader, Data*)*." BlockType still corresponds with the used AlarmType.
W#16#8002	ExtChannelDiagnosisData	Use only in connection with ChannelDiagnosisWithAddInfo in AlarmNotification and Diagnosis Data.
W#16#8003	Qualified	Graduated extended channel diagnostics
W#16#8004 - W#16#80FF	Reserved	
W#16#8100	Maintenance	Maintenance
W#16#8101 - W#16#81FF	Reserved	
W#16#8200	Reserved	
W#16#8201	iParameter	
- W#16#8202 - W#16#8FFF	Reserved	
W#16#9000 - W#16#9FFF	Reserved for profiles	Reserved for profiles
W#16#A000 - W#16#FFFF	Reserved	