AS-Interface Integration in PROFINET IO

Amendment 3

to

Fieldbus Integration into PROFINET IO

Version 1.0

Guideline

Version 1.0
August 2006

Order No: 7.012

Document identification: TC2-06-0004a

File name: ASI-Integration_7012_V10_Aug06

Prepared by the PROFIBUS Working Group 9 “Fieldbus Integration” in the Technical Committee 2 “Communication Profiles”.

In this specification the following key words (in **bold** text) will be used:

may: indicates flexibility of choice with no implied preference.

Should: indicates flexibility of choice with a strongly preferred implementation.

Shall: indicates a mandatory requirement. Designers **shall** implement these mandatory requirements to ensure interoperability and to claim conformance with this specification.

Publisher:
PROFIBUS Nutzerorganisation e.V.
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 – 7219 – 65 85 90
Fax: +49 – 7219 – 65 85 89
E-mail: info@profibus.com
Website: www.profibus.com

Contents

List of Figures

List of Tables

Revision Log

Identification	Version	Originator	Date	Change Note/History/Reason
RGJ	0.1	TC2 WG 9	2005/12/16	First draft
RGJ	0.2	TC2 WG 9	2006/01/19	Supplementations after meeting 2005-12-20
RGJ	0.3	TC2 WG 9	2006/02/24	Adjustments in conformity to decisions on 2006-01-30: I/O mapping of the AS-I slave profiles Behaviour of IOXS and substitute data Commands for CTT 2 slaves
RGJ	0.4	TC2 WG 9	2006/03/15	Adjustments in conformity to decisions on 2006-03-03: IOPS "Good" on CTT3 error/ periphery error Cancelling of Multiple Write Advise for upload by engineering
TC2-06-0004	0.5	TC2 WG 9	2006/04/05	Adaption of chapter 8.8.2.2 according official version of IEC 1158-5-10 References to Table 11 and Table 30 set to "strongly preferred implementation" Bugfixing Draft in PI review
RGJ	0.99	TC2 WG 9	2006/08/16	Transfer of identification to version 0.99
TC2-06-0004a	1.0	TC2WG9	2006/09/22	Final by decision of advisory board

1 Management Summary - Scope of this Document

The AS-Interface system will be applied mainly at the lowest level of a multi-level automation hierarchy in fieldbus system applications. As the use of Ethernet is extending to fieldbus technology, PROFINET IO provides a common solution for industrial communication. Because of this, a basic step is the integration of existing fieldbusses in PROFINET IO as a widely accepted common industrial Ethernet standard.

In the course of integrating existing fieldbusses in industrial Ethernet and PROFINET IO the mapping of the AS-i wire is defined onto the logical sight of PROFINET IO that will offer the following benefits:

– Customers can use their existing AS-i expertise and experience when they decide to use PROFINET IO.
– PROFINET IO does not yet provide some functions, such as auto configuration.
– The availability of a big number of actors and sensors within the AS-Interface system in combination with PROFINET IO will lead to a high benefit of the customer.
– Low installation costs for simple devices with a small amount of I/O data.

This guideline describes the basic concepts for the integration of the AS-Interface system into PROFINET IO. The following aspects are considered:

- Cyclic process data, acyclic parameterisations
- Diagnostics
- Alarms
- Start-up behaviour
- Engineering

The following diagram provides a system overview for AS-Interface and PROFINET IO.

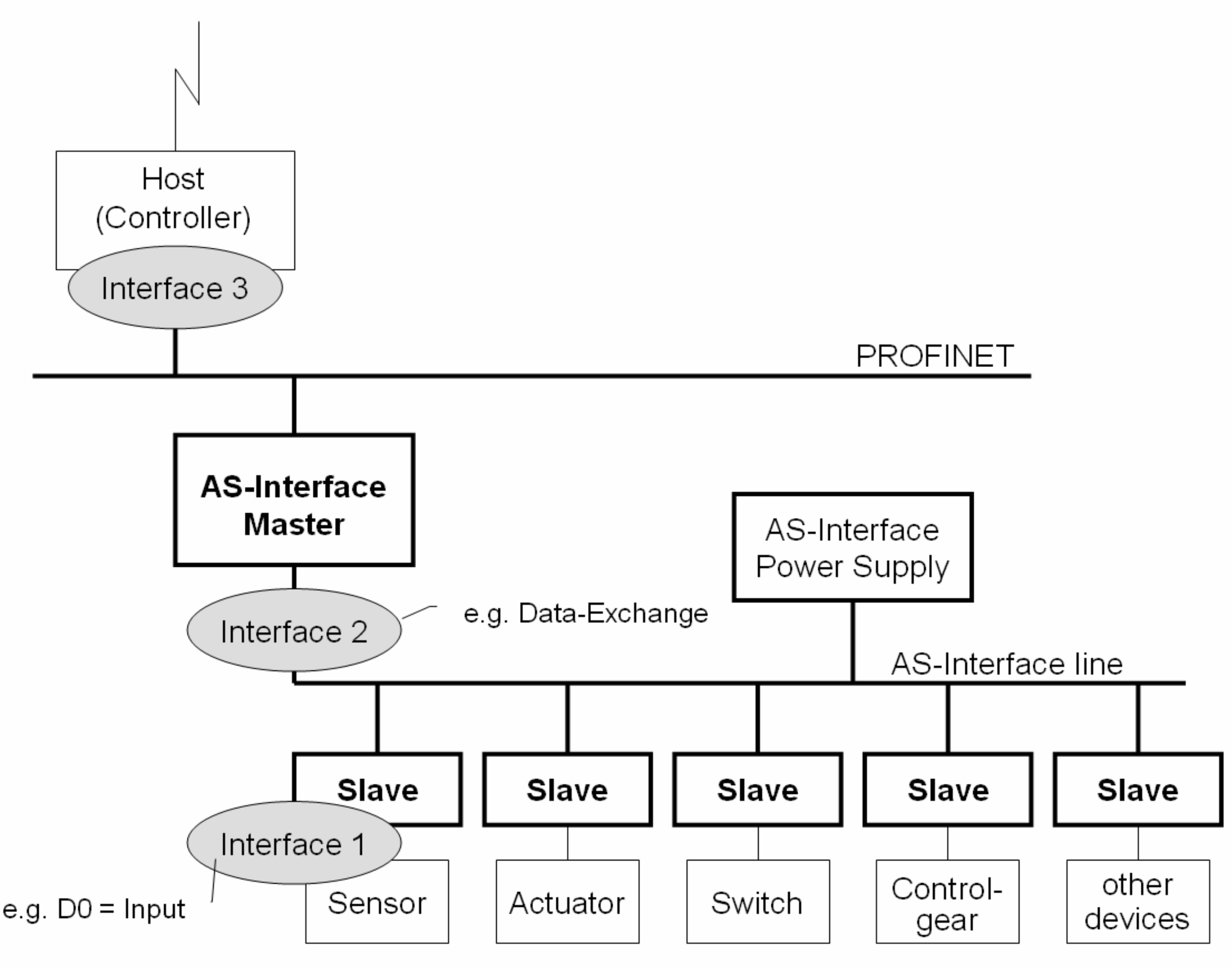

Figure 1: System overview

The definitions for AS-Interface integration are standardized. It is therefore possible to create interoperable and replaceable linking modules that connect AS-Interface to PROFINET IO.

2 List of affected patents

There is no affected patent known by the members of TC2WG9, Fieldbus Integration. The list is empty. No patent search, neither external nor internal, has been done by the members of the Working Group up to now. PROFIBUS International does not guarantee the completeness of this list.

3 Related Documents and References

References

- PROFIBUS glossary (www.profibus.com)

[1] IEC 61158/ 61784, Type 10: Digital data communication for measurement and control - Fieldbus for use in industrial control systems, Part 5 (Type 10: PROFINET)

[2] Actor Sensor Interface (AS-Interface) – Complete Specification, Version 3.0 of 9/04
Version 3.0, Revision 0

[3] Actor Sensor Interface – Profiles (Annex A and B),
Version 3.0, Revision 0

[4] Fieldbus Integration into PROFINET IO, Draft 0.5

4 Definitions and Abbreviations

4.1 Definitions

Linking module	A PROFINET IO device with integrated AS-i master functions. It is used to link AS-Interface to PROFINET IO.

4.2 Abbreviations

AIDI	Analogue Input Data Image *Analogue Input Data image of all AS-i slaves of an AS-i Line held by the AS-i master.*
AODI	Analogue Output Data Image *Analogue Output Data image of all AS-i slaves of an AS-i Line*
AR	Application Relationship *Relationship between PROFINET IO-Controller and PROFINET IO-Device or between PROFINET IO-Supervisor and PROFINET IO-Device* *IO-AR: Relationship between PROFINET IO-Controller and PROFI-NET IO-Device* *Supervisor AR Relationship between PROFINET IO-Supervisor and PROFINET IO-Device*
APDU	Application Protocol Data Unit *By the APDU, cyclic data are transferred. It contains the flags "Data Flag", "AR State Flag", "Provider State Flag", and "Problem Indicator Flag".*
API	Application Process Identifier *Every submodule contains one ore several special application processes referenced by their identifier (API).*
AS-i	Actor Sensor Interface, AS-Interface *Details can be found in "Actor Sensor Interface - Complete Specification" version 3.0 and "Actor Sensor Interface - Profiles, Annex A & B".*
CD	Configuration Data *The CD comprises the actual copies of the input/output configuration and the identification codes (ID, ID1, ID2) of the AS-i slaves. The AS-i master stores these in a list in the volatile memory.*
Data Record	Data unity *Is read or written acyclic by the IO-Controller. The data record lies on a sub-module and is addressed by an index.*
Extended address mode	Duplicates the maximum of the number of slaves at an AS-i wire from 31 (ad-dress range 1 to 32) to 62 (address range 1A/1B to 31A/31B).
GSD / GSDML	Generic Station Description / Markup Language
IDI	Input Data Image *Binary input data image of all AS-i slaves of an AS-i wire.*
I/O-Configuration (I/O-Code)	*Array with 4 bits that defines the direction of the data flow to the I/O ports of the AS-i slaves.*
Identification Code (ID-Code)	*Array of 4 bits that determines the AS-i slave type of a given I/O configuration (option: Extended ID-Codes consist of additional 2x4 bits).*
ModuleID	*Module Ident Number according to* [1]
SubmoduleID	*Submodule Ident Number according to* [1]

5 Responsibilities

Not defined yet.

6 Introduction

6.1 AS-Interface Configuration in PROFINET

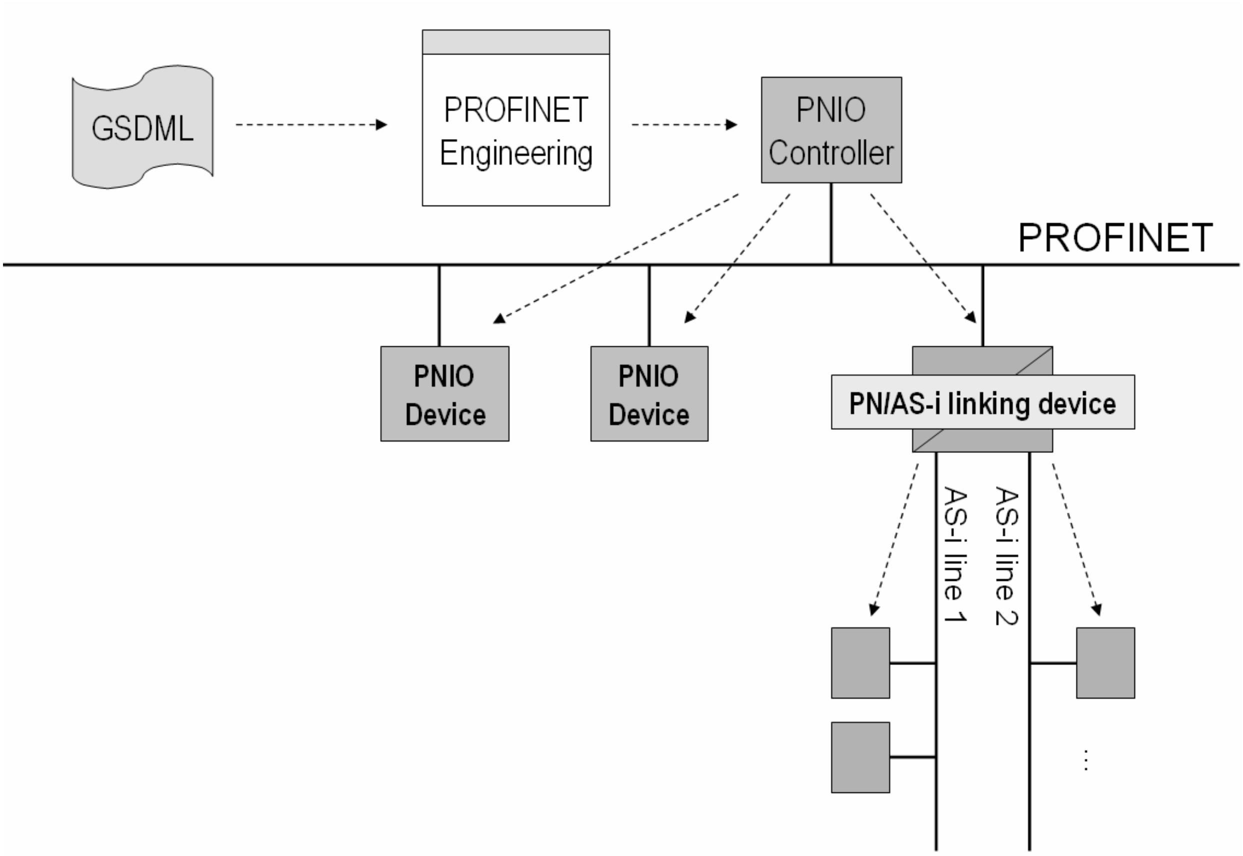

Figure 2: Integration of AS-Interface in PROFINET IO

For coupling an AS-i linking device to PROFINET IO the controller's engineering uses a unique GSDML suitable for all PN/AS-i linking devices. The GSDML is delivered together with the PN/AS-i linking devices or can be downloaded from the manufacturers' web sites.

Dropping a PN/AS-i linking device on the graphical surface of the PROFINET IO controller's engineering tool into an engineering project all usable slots of the arising linking device instance are filled with wild card modules (defining 4 bits of input and 4 bits of output data). A further editing is not necessary if the PN/AS-i linking device uses auto configuration (by pressing the push button). The download to the IO controller may be activated instantly.

Alternatively, the AS-i slaves may be parameterised and configured in the engineering according to their requests. For this purpose the initially existing wild card modules may be deleted or replaced by better suitable modules. For that reason several modules with different I/O widths and directions are offered by GSDML. For each module, the engineering displays definite PNIO start-up data records for parameterisation and configuration that may be changed by the user. Finally, the user-completed planning is loaded together with that of other PNIO devices onto the controller.

The PROFINET IO controller loads the fed configuration and the parameter data into the IO devices by sending the PROFINET IO data records to the IO devices and with these, the user adapted data into the PN/AS-i linking device, which receives parameterisation record data for the link proxy as well as for the underlying AS-i slaves.

Subsequently, if different configuration or parameter data have been loaded, the AS-i Master situated in the linking device restarts passing the offline phase parameterises its depending AS-i slaves according to the received parameterisation record data and changes to protected mode.

6.2 Intended Upload of AS-i Master's Configuration with Future Engineering Tools

In the previous chapter two ways to configure the AS-i master are described:

- Local configuration with wild card modules
- Central configuration with modules taken from the GSDML File

In addition AS-i users may wish to do a local configuration and then upload this configuration into the engineering tool for documentation and modification if necessary.

In order to provide this use case the engineering tool has to be extended in an AS-i specific way as follows:

- Upload of the Real Identification and the PCD.
- Construction of compliant engineering modules or replacement of an existing configuration by modules with appropriate Module Ident Number and Submodule Ident Number according the Real Identification.
- Adding of the uploaded PCD data to the constructed and replaced modules respectively.

These requirements are not fulfilled by existing PROFINET engineering tools, but shall be mentioned here to give advice to developers of those tooling extensions.

6.3 Structure of The Linking Device

The general structure of a PN/ AS-i linking device underlying the described mapping model below is shown in Figure 3. The defined mapping structure is to be implemented in the “Linking Layer” block. The access to Ethernet or to the 1 up to n AS-i lines may be carried out by a standard software packet or by an ASIC with appropriate functionality.

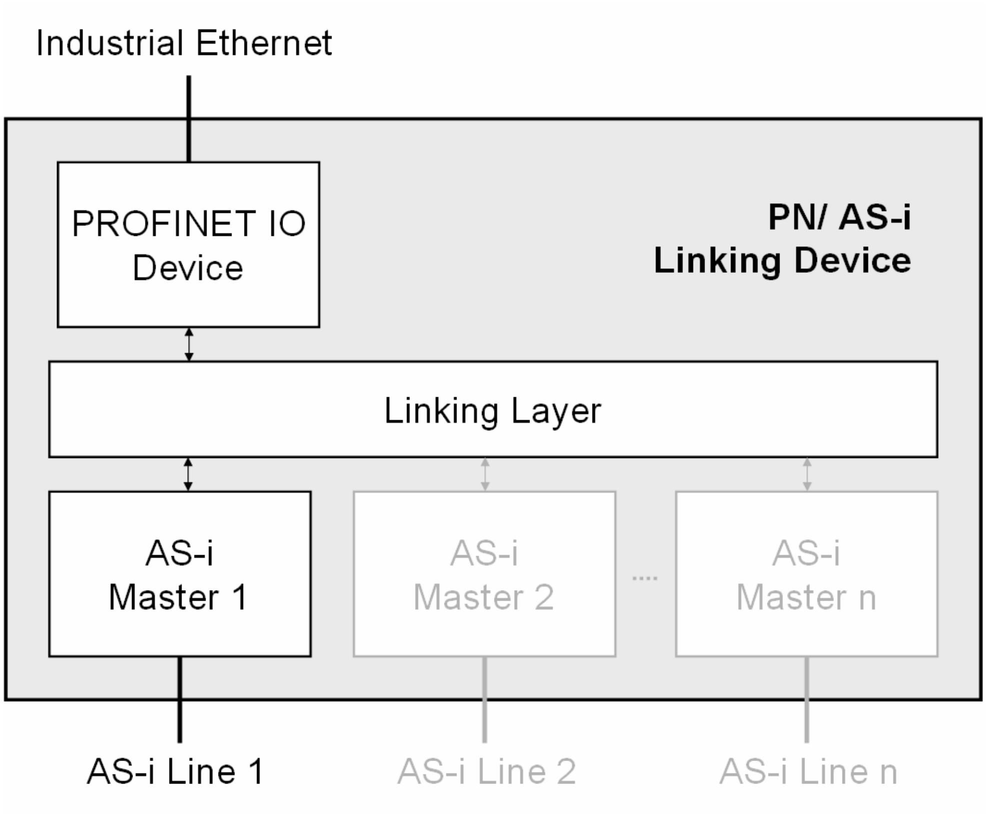

Figure 3: Internal structure of the PN/ AS-i linking device

7 Mapping Model

7.1 Selection

The integration of AS-Interface in PROFINET IO is done by means of the modular mapping. In this model, every AS-Interface device has a corresponding module in a PROFINET IO device according to the slot/subslot addressing model [1]. The device description of the PROFINET IO device contains all possible AS-Interface devices with their device parameters.

Modular mapping offers the following advantages/disadvantages:

+ **Transparent engineering:**
  Every AS-Interface slave is visible as a module in the engineering tool of the PROFINET IO controller. The device description is a generic module description in the GSDML depending on its I/O lengths.

+ **Good data availability information:**
  The data of all AS-Interface modules can be transferred in a single PROFINET IO frame. Additionally any subslot adds an IOPS/IOCS that show the actual data availability of at least slave granularity.

− **Small IO data size:**
  Binary data on AS-Interface has a maximum width of 4 bits. Being in one subslot, the binary data of every slave occupies the minimum of 1 byte. This can cause more IO data occupied than necessary. On the other hand, data of unavailable slaves is not transmitted.

+ **Slaves with fixed submodules:**
  The combined transaction type profiles of AS-i specification 3.0 define slaves with digital and analogue Data. These different kinds of data are transferred in different but fixed subslots. So depending on the possibilities of the controller these data may be mapped to different memory sections.

Conclusion: Modular mapping offers the most advantages and the fewest disadvantages for the mapping of AS-Interface to PROFINET IO. This mapping method is therefore described further within this document.

7.2 Supervisor Usage

Because of the mapping of real AS-i devices and AS-i lines to PROFINET IO modules (and so to its submodules) there is a dependency between particular submodules. So an IO Supervisor may not takeover any kind of modules as stated in Clause 1. These restrictions will be described in chapter 8.8 “Dynamic Behaviour” and result in directions of stetting certain parameters in the ModuleDiffBlock.

Clause 1:

Modules without permission for takeover are explicitly noted in 8.8.2 and **shall** be handled accordingly in the SubmoduleState (see Table 36, Table 38 and Table 39): The ‘AddInfo’ field **shall** be set to ‘Takeover is not allowed’.

8 Slot/Subslot Mapping

8.1 Overview

The following diagram illustrates AS-Interface mapping to the slot/subslot-addressing model of PROFINET IO. This diagram forms the basis of this document. All subsequent sections refer to parts of this diagram.

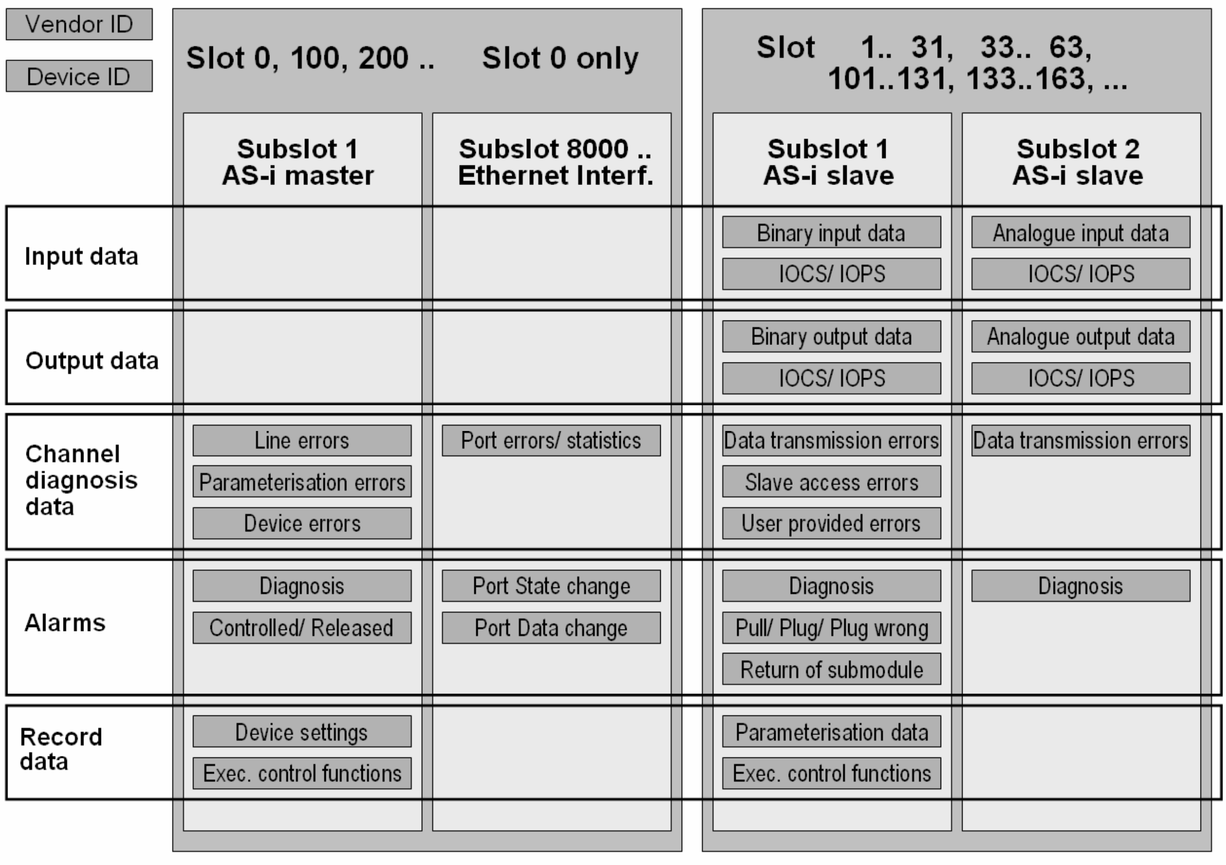

Figure 4: Mapping of AS-i properties to the slot/subslot model of PROFINET IO

The relevant fieldbus elements of AS-Interface are given in table,
whereas AS-i address $\alpha \in [1..\ 31, 33..\ 63]$
and AS-i line $\lambda \in [0..\ 15]$: $100 * \lambda + \alpha$

	Slot	**Subslot**	**Identification**	**Output data and status**	**Input data and status**	**Alarms**	Records	Channel diagnosis
Ethernet interface	0	0x8000, 0x8001, ..	V1/ V2			Usage		Port errors/ statistics
AS-i Master diagnosis	$100 * \lambda$	1	№ of AS-i line			Usage		Mapped channel error types
Execution control functions for AS-i line	$100 * \lambda$	1	№ of AS-i line				Usage	
Slave Input/Output data	$100 * \lambda + \alpha$	1, 2	AS-i line № & AS-i address	1 :1	1 :1			
Slave diagnosis	$100 * \lambda + \alpha$	1, 2	AS-i line № & AS-i address			Usage		Mapped channel error types
Execution control functions for AS-i slave	$100 * \lambda + \alpha$	1	AS-i line № & AS-i address				Usage	
AS-i Parameterisation	$100 * \lambda + \alpha$	1, 2	AS-i line № & AS-i address				Part of GSD	
AS-i Configuration	$100 * \lambda + \alpha$	1, 2	AS-i line № & AS-i address				Part of GSD	

Table 1: Functional mapping

8.2 Mapping Principals

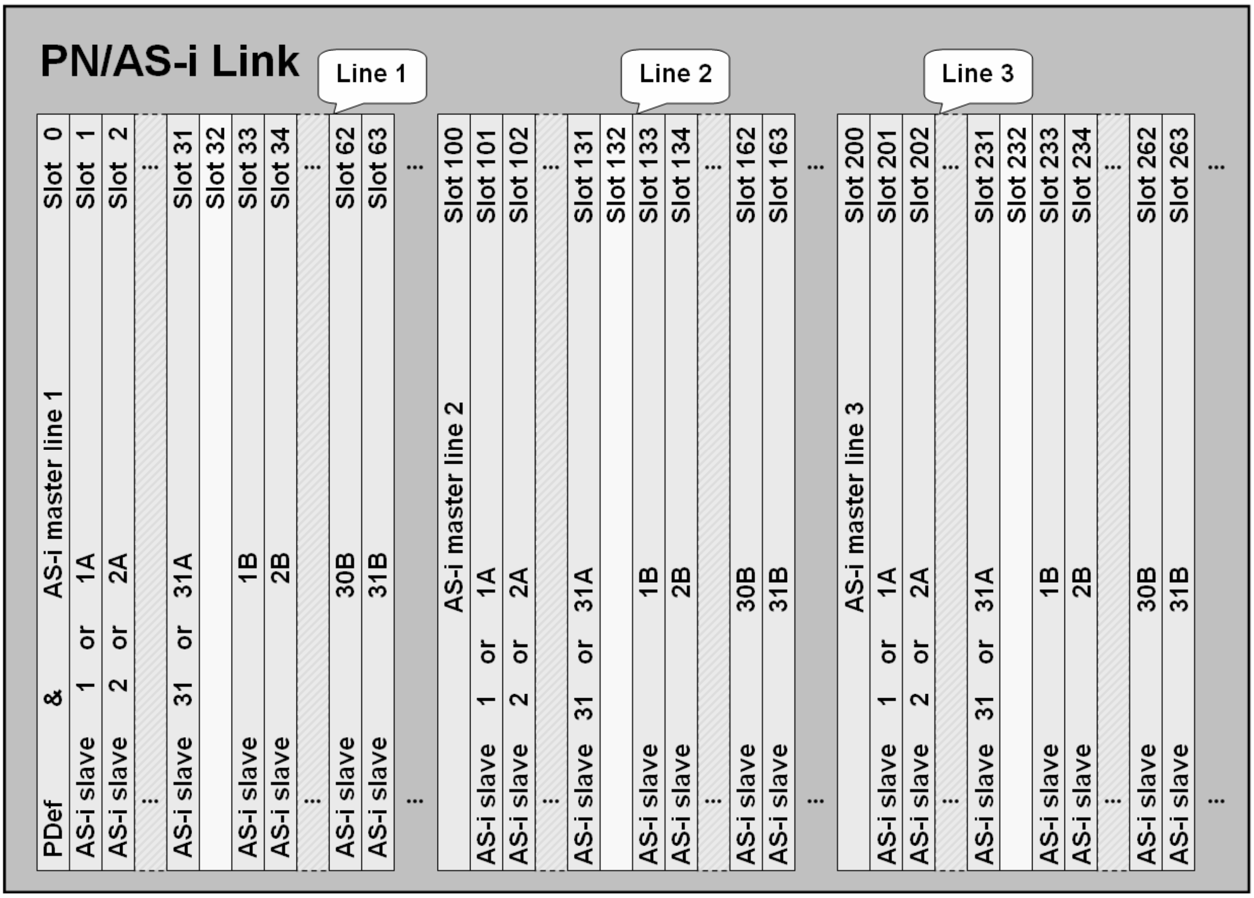

Figure 5: Modelling of an AS-i Configuration for a controller application

Being a comparatively easy composed connection possibility to PROFINET IO an AS-Interface line will be best mapped by the modular model:
The AS-i slaves **shall** be represented as modules of a PROFINET IO device. The module of a line λ (multiple masters apply to λ > 1) with an AS-i address n **shall** be placed with the following rule for computing the slot number

Element	Slot	Range of values	
Standard slaves and A-slaves	100 *λ +α	AS-i address	α ∈ [1.. 31],
B-slaves	100 *λ +α +32	AS-i line	λ ∈ [0.. 15]

Table 2: Mapping rule for AS-i slave addresses to PROFINET IO slots

If a module is used for an AS-i slave with standard addressing mode in slot ς, no other (B-slave) module must be placed into slot ς+32. Combined transaction type 2 slaves are holding one sub module for the digital and another for the analogue value on subslot addresses 1 (digital channel) and 2 (analogue channel). These modules will below be called “AS-i slave proxies”.

The sub module in slot 0/ subslot 1 **shall** represent the link proxy and the proxy of the 1st AS-i line respectively. It is additionally the device access point (DAP) of the PROFINET IO device. Further AS-i lines **shall** have their own proxy submodule 1 in the slots with the numbers defined in Table 3.

Element	Slot	Range of values	
Master of the 1st AS-i line	100 *λ	AS-i line	λ = 0
Masters of further AS-i lines		AS-i line	λ ∈ [1.. 15]

Table 3: Mapping rule for AS-i line’s masters to PROFINET IO slots

These objects report alarms for each AS-i line, and commands are transmitted to them that refer to this line overall. These proxy modules below will be called “AS-i line proxies”.

If a PN/AS-i link is used as a PROFINET V2 compatible IO device, the PROFINET IO PDev in slot 0 **shall** be accessible at subslot 0x8000 et sqq.

8.3 Structure of The Module Ident Number

The ModuleIdentNumber (short ModuleID) **should** have the following structure:

Bit 31..24	Bit 23..16	Bit 15..4	Bit 3..0
Type	Select	Reserved = 0x000	Count

Figure 6: Structure of the ModuleIdentNumber

The parameters values defined in the following selection list **should** be accepted:

<table>
<tr><th>Parameter</th><th colspan="2">Description</th><th>Values</th></tr>
<tr><td>Type</td><td colspan="2">Type Id of the module</td><td>0x01 AS-i line proxy
0x02 AS-i slave proxy</td></tr>
<tr><td rowspan="4">Select</td><td rowspan="3">If Type = 1
Identification of DAP and number of lines</td><td>Bit 23..21</td><td>000_{bin} reserved</td></tr>
<tr><td>Bit 20</td><td>0_{bin} DAP for PROFINET V1.1 controllers
1_{bin} DAP for PROFINET V2.x controllers</td></tr>
<tr><td>Bit 19..16</td><td>0x0 AS-i master with 1 line
0x1 AS-i master with 2 lines
0x2 AS-i master with 3 lines
…
0xF AS-i master with 16 lines</td></tr>
<tr><td colspan="2">If Type = 2
Identification of the AS-i I/O sizes</td><td>0x01 1 byte in binary
0x02 1 byte out binary
0x03 1 byte in/out binary
0x04 1 word in analogue
0x05 1 word out analogue
0x06 1 word in/out analogue
0x07 2 word in analogue
0x08 2 word out analogue
0x09 2 word in/out analogue
0x0A 4 word in analogue
0x0B 4 word out analogue
0x0C 4 word in/out analogue
0x21 1 byte out binary, 1 word in analogue
0x22 1 byte in/out binary, 2 words in/out analogue
0x23 1 byte in/out binary, 4 words in/out analogue
0x40 0 byte module for CTT5 slaves, 2^{nd} – 4^{th} address</td></tr>
<tr><td>Reserved</td><td colspan="2"></td><td>0x000</td></tr>
<tr><td rowspan="2">Count</td><td colspan="2">If Type = 1
Identification of the AS-i line</td><td>0x0 line 1
0x1 line 2
0x2 line 3
0x3 line 4</td></tr>
<tr><td colspan="2">If Type = 2
Number of subslots</td><td>0x1 module with 1 subslot
0x2 module with 2 subslots</td></tr>
</table>

Table 4: Parameters of the structure ModuleID

8.4 Structure of The Submodule Ident Number

The SubmoduleIdentNumber (short SubmoduleID) **should** have the following structure:

Bit 31..24	Bit 23..16	Bit 15..8	Bit 7..0

Type	Reserved = 0x00	Input-length	Output-length

Figure 7: Structure of a SubmoduleIdentNumber

The parameters values defined in the following selection list **should** be accepted:

Parameter	Description	Values
Type	Type Id of the submodule	0x01 no data 0x02 binary data 0x03 analogue data
Reserved		0x00
Input-length	Length of input data in bytes	0 if Type = 0x01 0, 1 if Type = 0x02 0.. 2, 4, 8 if Type = 0x03
Output-length	Length of output data in bytes	0 if Type = 0x01 0, 1 if Type = 0x02 0.. 2, 4, 8 if Type = 0x03

Table 5: Parameters of the structure SubmoduleID

8.5 AS-i Master Module

On a PN/AS-i linking device, there exist two sorts AS-i master modules. Slot 0 of a PROFINET V2.0 device (or later) **shall** keep submodules that represent the hardware Ethernet ports (see [1]). Additionally the AS-i master submodule in subslot 1 **may** be capable of

- Receiving device specific parameterisation by record data that are not valid for the other AS-i master modules (line 2 to 15 maximum)
- Reporting device specific channel diagnoses that are not set by other AS-i master modules.

8.5.1 Modelling of The DAP

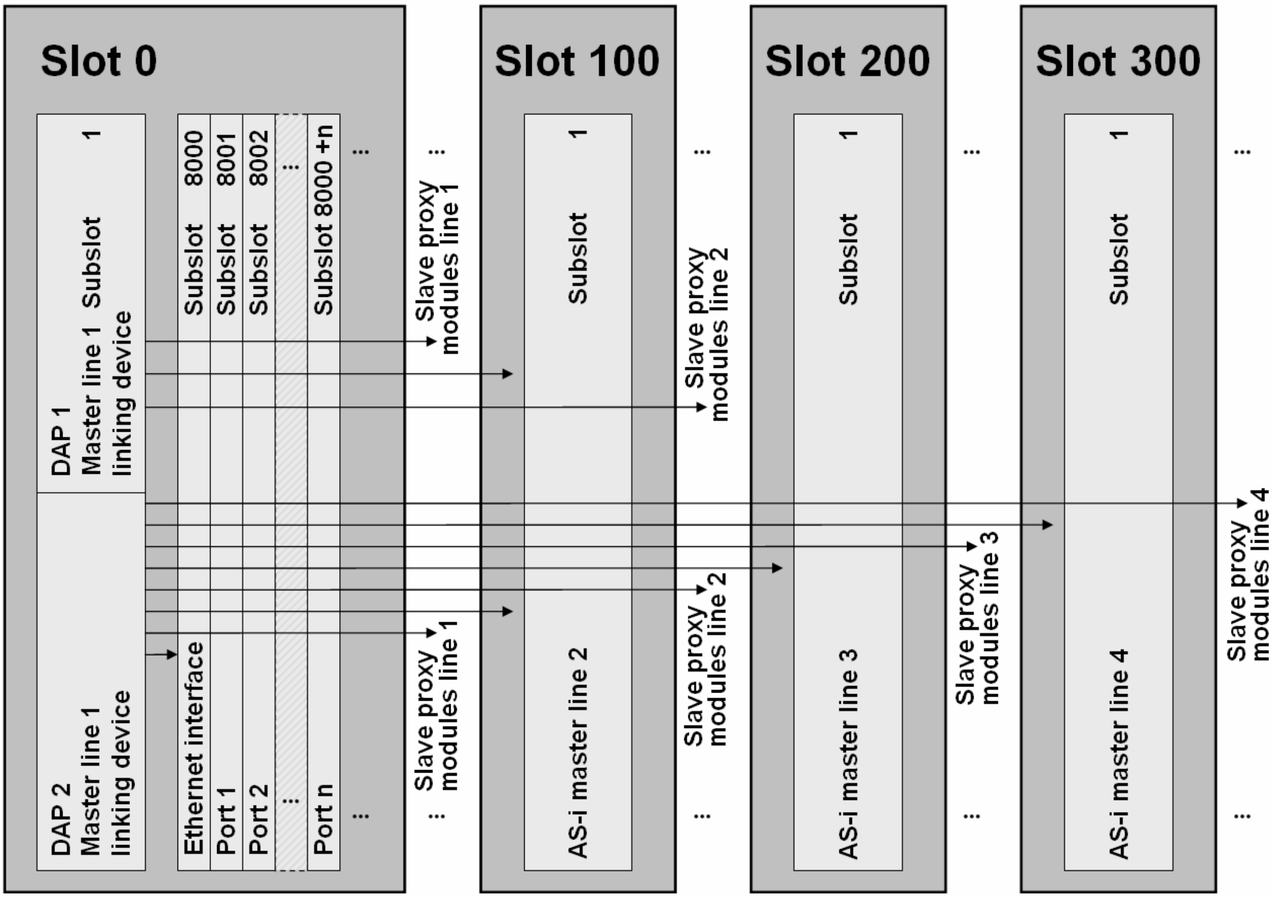

Figure 8: Modelling of DAPs

The user **should** be offered different DAPs for a multiple master linking device. Each DAP **should** represent a device with a defined number of AS-i lines. As an example, see Figure 8. In Figure 8 DAP 1 offers a PN/AS-i linking device with only two lines in use and referring to the PROFINET specification V1.1 (no port diagnosis), whereas DAP 2 references all possible 4 AS-i lines for V2.0 controllers (or later).

For usage within an IRT network the linking device **may** offer a third party access. An establishment of an end-to-end relationship of IRT applications is not defined in this document.

8.5.2 Modelling of The AS-i lines' masters

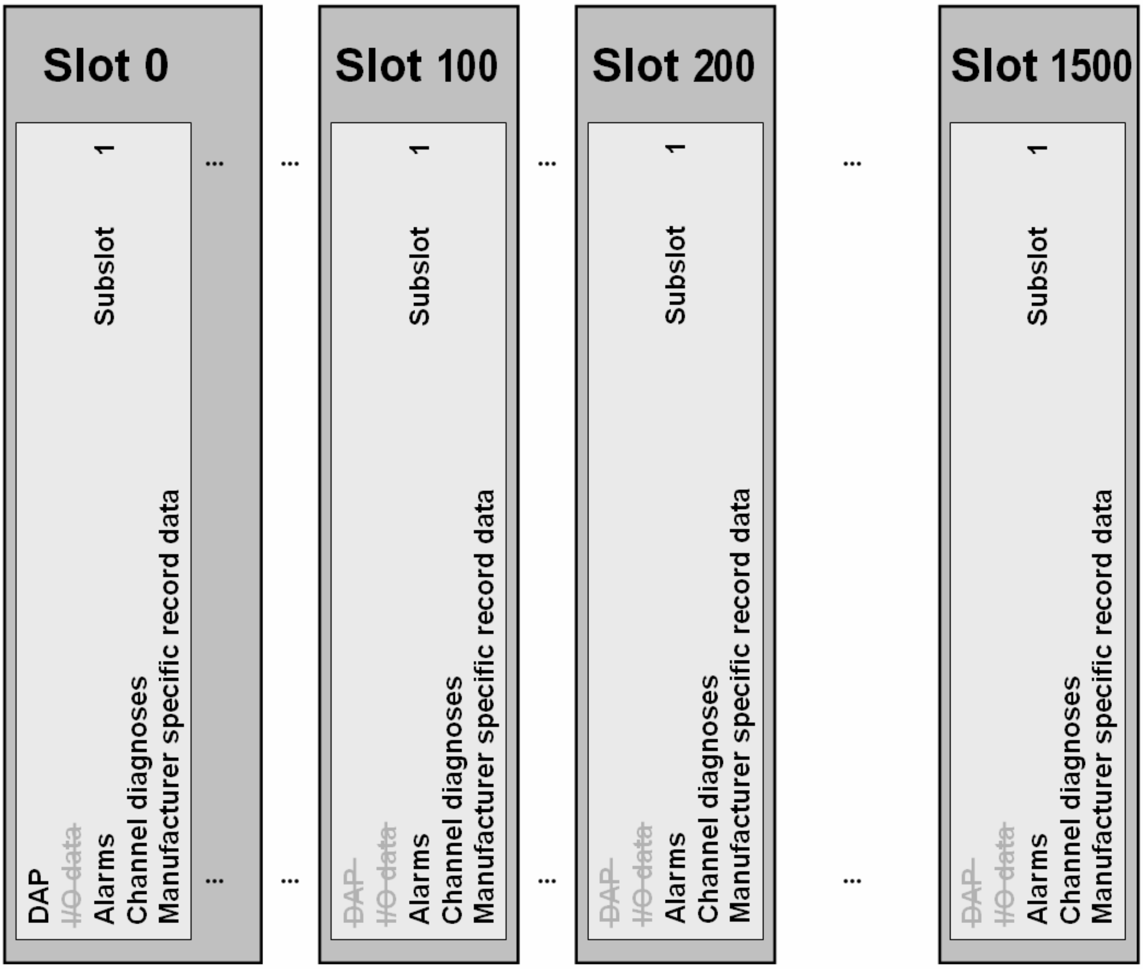

Figure 9: Modelling of the submodules of AS-i master proxies at the different lines

The mapping of the AS-i line's masters to PROFINET IO modules is defined in Table 3. Each submodule representing the AS-i master of a used AS-i line that is used **shall** be fixed in subslot 1.

For AS-i master proxy modules the parameter values of the structure "ModuleID" **should** be restricted to Type 1 (AS-i line proxy) in Table 4.

In addition, for AS-i master proxy modules the parameter values of the structure "SubmoduleID" **should** be restricted to Type 1 (no data) in Table 5.

The submodules of the AS-i master proxies **shall** support elements of following PROFINET IO ASEs:

- Alarm details see 8.5.3
- Channel diagnosis details see 8.5.4
- Data records details see 8.5.5 and 8.5.6

8.5.3 Alarms

Table 6 defines the alarm types and its special usages for the AS-i integration that **shall** be provided by AS-i master proxy modules. For information with under which conditions these alarms have to be initiated, see chapter 8.8.3.

Type	Subslot	Usage
Diagnosis	1	Signals the event of a channel diagnosis, see Table 7.
	1	An AS-i master proxy module signals the return of a not configured slave and its failure, respectively. This alarm type is used if there is no module plugged for the appropriate AS-i slave address.
Process		No special use
Pull		No special use
Plug		No special use
Status	1	An AS-i master proxy module signals a change in the state of an AS-i master to protected mode or configuration.
Update		No special use
Redundancy		No special use
Controlled by supervi-sor		No special use
Released		No special use
Plug Wrong Submod-ule		No special use
Return of Submodule		No special use
Diagnosis disappears	1	Signals a disappearing diagnosis event within a submodule, see Table 41. If this alarm is conveyed without an Alarm Item, all previous diagnosis is gone.
	1	An AS-i master proxy module signals the withdrawal of a not config-ured slave. This alarm type is used if there is no module plugged for the appropriate AS-i slave address.
Multicast Communication Mismatch		No special use
Port Data Change Notification		No special use
Sync Data Change Notification		No special use
Isochronous Mode Problem Notification		No special use

Table 6: Special usages of alarms by AS-i master modules

8.5.4 Channel Diagnosis

Table 7 defines Channel Diagnoses and error types to be supported according the column 'Requirement'. For information with under which conditions Channel diagnosis alarms have to be initiated, see chapter 8.8.3.

Error type	Error description	Sub slot	Channel Number	Reason	Requirement
0x0011	Failure of power module	1	8000	AS-i master detects power fail (e.g. short circuit)	**Shall**
0x0010	Parameterisation error	1	AS-i address 0: **0** Standard- and A-slaves: **α**, B-slaves: **α+32**	Appearing: AS-i master sets in LDS the bit for a not configured slave (i.e. not in the LPS) with AS-i address α Disappearing: AS-i master resets in LDS the bit for a not configured slave (i.e. not in the LPS) with AS-i α. Also valid for slaves without a configured GSDML module that are defined in the LPS (appearing and disappearing).	**Shall**
0x0180	Earth fault	1	8000	AS-i master detects earth fault at the appropriate AS-i Line	**Shall**

Table 7: Supported Error types of Channel Diagnoses at AS-i master modules

8.5.5 AS-i Line Parameterisation Record Data (Index 130/ 0x82)

The AS-i line parameterisation record data **shall** be defined in the GSDML according Figure 10. It **shall** be used if manufacturer specific options are offered.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	Option 7	Option 6	Option 5	Option 4	Option 3	Option 2	Option 1	Option 0
3	Reserved = 0x00							

Figure 10: Structure of Parameterisation Record Data for an AS-i line proxy

The option bits of byte 2 **should** be preset in both the GSDML and the PN/AS-i linking device by the default values defined in Table 8.

Parameter	Description	Setting	Default
Manufacturer defined 0 .. Manufacturer defined 7	Parameter usage **may** be manufacturer specific.	<Undefined>	0_{bin}

Table 8: Setting of the bit parameters in structure of the parameterisation record data

8.5.6 Execution Control Functions

In order to provide AS-i Execution Control functions on a host controller this interface **shall** be mapped to PROFINET IO data records (see Figure 1). For this reason, AS-i Interface 3 offers an open access to AS-i masters of PN/AS-i linking devices. Missing definitions in [2] and [3] for this expanded context are established in the following.

PN/AS-i linking devices **may** implement the AS-i Execution control functions listed in Table 9 for AS-i master proxies. The specified data record indices **shall** be taken neither for other purposes of use nor for similar implementations of these functions.

AS-i	PROFINET IO		Details see chapter
Execution control function	**Service**	**Index**	
Read_IDI	RecordDataRead	0x0001	8.5.6.1
Write_ODI	RecordDataWrite	0x0002	8.5.6.2
Store_Actual_Parameters	RecordDataWrite	0x0007	8.5.6.3
Store_Actual_Configuration	RecordDataWrite	0x000A	8.5.6.4
Set_LPS	RecordDataWrite	0x000C	8.5.6.5
Get_LPS	RecordDataRead	0x000D	8.5.6.6
Get_LAS	RecordDataRead	0x000E	8.5.6.7
Get_LDS	RecordDataRead	0x000F	8.5.6.8
Get_Flags	RecordDataRead	0x0010	8.5.6.9
Set_Operation_Mode	RecordDataWrite	0x0011	8.5.6.10
Set_Offline_Mode	RecordDataWrite	0x0012	8.5.6.11
Set_Data_Exchange_Active	RecordDataWrite	0x0013	8.5.6.12
Change_Slave_Address	RecordDataWrite	0x0014	8.5.6.13
Set_Auto_Addr_Enable	RecordDataWrite	0x0015	8.5.6.14
Get_Auto_Addr_Enable	RecordDataRead		8.5.6.15
Execute_Command	RecordDataWrite	0x0016	8.5.6.16
Get_LPF	RecordDataRead	0x0017	8.5.6.17
	RecordDataRead		
Write_Extended_ID-Code_1	RecordDataWrite	0x0018	8.5.6.18
Read_AIDI	RecordDataRead	0x0019	8.5.6.19
Write_AODI	RecordDataWrite	0x001A	8.5.6.20

Table 9: Execution control functions for AS-i master proxy

Clause 2:

The values of the AS-i result parameter ‘Status’ is mapped to PROFINET IO record data services as defined in Table 10 and Table 11. Following coding of ErrorCode2 **may** be used for mapping manufacturer specific results: 0x20 through 0x07F. All other values are reserved and **shall** be reserved for further specification.

If mapped to the PROFINET IO service RecordDataRead, Execution control functions generate in the response the field IODWriteResHeader.PNIOStatus according to Table 10. The parameters are valid for the service IODReadRes.

PROFINET response	AS-i result parameter "Status"	
	OK	NOK
Service	Read.rsp(+)	Read.rsp(-)
ErrorCode	0x00	0xDE
ErrorDecode	0x00	0x80
ErrorCode1	0x00	0xA0
ErrorCode2	0x00	0xFF

Table 10: Mapping of AS-i result "Status" to "PNIOStatus" in RecordDataRead

If mapped to the PROFINET IO service RecordDataWrite, Execution control functions generate in the response the field IODWriteResHeader.PNIOStatus according to Table 11. The parameters are valid for the service IODWriteRes (referred to as '~' in Table 11).

PROFINET response	AS-i result parameter 'Status' in IODWriteRes									
	OK	NOK	SND	SD0	SD2	DE	RE	SE	AT	ET
Service	~.rsp(+)	~.rsp(-)	~.rsp(-)	~.rsp(-)	~.rsp(-)	~.rsp(-)	~.rsp(-)	~.rsp(-)	~.rsp(-)	~.rsp(-)
ErrorCode	0x00	0xDF	0xDF	0xDF	0xDF	0xDF	0xDF	0xDF	0xDF	0xDF
ErrorDecode	0x00	0x80	0x80	0x80	0x80	0x80	0x80	0x80	0x80	0x80
ErrorCode1	0x00	0xA1	0xA1	0xA1	0xA1	0xA1	0xA1	0xA1	0xA1	0xA1
ErrorCode2	0x00	0xFF	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C

Table 11: Mapping of AS-i result "Status" to "PNIOStatus" in RecordDataWrite

8.5.6.1 Read_IDI (Index 1/ 0x0x1)

The AS-i execution control function is used by the host interface in order to request that the execution control shall deliver the input data values from the input data image to the host [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0001 addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the AS-i result argument 'Image' – i.e. the IDI – as user data.
  - The IDI structure 'Input_Data_Image' **shall** be composed of two blocks for the standard and A/B-slaves according to [3], Table A5-3 and Table A5-4, with a fixed value of Max_Data = 31 each. For details, see Figure 11.
  - The IDI data block **shall** have top position in the RecordDataRead.Data field.
- **shall** contain the AS-i result argument 'Status' as defined in Clause 2/ Table 10.
- **should** contain no further arguments in the RecordDataRead.Data field.

Byte №	Bits 7-4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bits 3-0
0		Major Version = 0x00								
1		Minor Version = 0x00								
2	Reserved	0	0	0	0	Bit 4	Bit 3	Bit 2	Bit 1	Slave 1/ 1A
3	Slave 2/ 2A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 3/ 3A
4	Slave 4/ 4A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 5/ 5A
5	Slave 6/ 6A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 7/ 7A
6	Slave 8/ 8A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 9/ 9A
7	Slave 10/ 10A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 11/ 11A
8	Slave 12/ 12A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 13/ 13A
9	Slave 14/ 14A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 15/ 15A
10	Slave 16/ 16A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 17/ 17A
11	Slave 18/ 18A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 19/ 19A
12	Slave 20/ 20A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 21/ 21A
13	Slave 22/ 22A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 23/ 23A
14	Slave 24/ 24A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 25/ 25A
15	Slave 26/ 26A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 27/ 27A
16	Slave 28/ 28A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 29/ 29A
17	Slave 30/ 30A	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 31/ 31A
18	Reserved	0	0	0	0	Bit 4	Bit 3	Bit 2	Bit 1	Slave 1B
19	Slave 2B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 3B
20	Slave 4B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 5B
21	Slave 6B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 7B
22	Slave 8B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 9B
23	Slave 10B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 11B
24	Slave 12B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 13B
25	Slave 14B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 15B
26	Slave 16B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 17B
27	Slave 18B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 19B
28	Slave 20B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 21B
29	Slave 22B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 23B
30	Slave 24B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 25B
31	Slave 26B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 27B
32	Slave 28B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 29B
33	Slave 30B	Bit 4	Bit 3	Bit 2	Bit 1	Bit 4	Bit 3	Bit 2	Bit 1	Slave 31B
34		Reserved = 0x0000								
35										

Figure 11: Structure of RecordDataWrite/ RecordDataRead.Data: parameters IDI and ODI

8.5.6.2 Write_ODI (Index 2/ 0x0x2)

The AS-i execution control function is used by the host interface in order to request that the execution control shall receive the data values from the host to be put in the Output_Data_Image (ODI). The data, which is delivered from the Host interface, shall be inverted before being written into the Output_Data_Image as demanded in [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0002 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the AS-i parameter 'Image' – i.e. the ODI – in the RecordDataWrite.Data field.
  - o The ODI structure 'Output_Data_Image' **shall** be of two blocks for the standard and A/B-slaves according to [3], Table A5-5 and Table A5-6, with a fixed value of Max_Data = 31 each. For details, see Figure 11.
  - o The ODI data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** contain the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.5.6.3 Store_Actual_Parameters (Index 7/ 0x7)

The AS-i execution control function Store_Actual_Parameters is used by the host interface in order to request that the execution control shall copy the parameter values of the parameter image (PI) array into the permanent parameter (PP) array [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0007 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **should** contain an empty RecordDataWrite.Data field.

The response

- **should** be sent after the AS-i parameterisation data are stored into the non-volatile memory.
- **should** contain the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.5.6.4 Store_Actual_Configuration (Index 10/ 0xA)

The AS-i execution control function Store_Actual_Configuration is used by the host interface in order to request that the execution control shall copy the values of the configuration data image (CDI) array (without the value for slave 0) into the permanent configuration data (PCD) array and the values of the list of activated slaves (LAS) into the list of projected slaves (LPS) [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x000A addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **should** contain an empty RecordDataWrite.Data field.

The response

- **should** be sent after the AS-i configuration data is stored into the non-volatile memory.
- **should** contain the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.5.6.5 Set_LPS (Index 12/ 0xC)

The AS-i execution control function Store_Actual_Configuration is used by the host interface in order to request that the execution control shall set the values of LPS [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x000C addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the LPS in the RecordDataWrite.Data field according to the structure defined in Figure 12.
  - o The values of the bits parameters **shall** set according to [2] and [3].
  - o The LPS data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** be sent after the LPS bits are stored into the non-volatile memory.
- **should** contain the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	List_wS0 [1]							
3								
4								
5								
6	List [2]							
7								
8								
9								
10	Reserved = 0x0000							
11								

Figure 12: Structure of RecordDataWrite / RecordDataRead.Data: parameters LPS/ LAS/ LDS/ LPF

8.5.6.6 Get_LPS (Index 13/ 0xD)

The AS-i execution control function Get_LPS is used by the host interface in order to request that the execution control shall return the values from the list of projected slaves (LPS) [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x000D addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the LPS as user data according to the structure defined in Figure 12.
  - The values of the bit parameters **shall** set according to [2] and [3].
  - The LPS data block **shall** have top position in the RecordDataRead.Data field.
- **shall** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 10.
- **should** contain no further arguments in the RecordDataRead.Data field.

8.5.6.7 Get_LAS (Index 14/ 0xE)

The AS-i execution control function Get_LAS is used by the host interface in order to request that the execution control shall return the values from the list of activated slaves (LAS) [2].

This function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x000E addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the LAS as user data according to the structure defined in Figure 12.
  - The values of the bit parameters **shall** be set according to [2] and [3].
  - The LDS data block **shall** have top position in the RecordDataRead.Data field.
- **shall** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 10.
- **should** contain no further arguments in the RecordDataRead.Data field.

8.5.6.8 Get_LDS (Index 15/ 0xF)

The AS-i execution control function Get_LDS is used by the host interface in order to request that the execution control shall return the values from the list of detected slaves (LDS).

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x000F addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the LDS as user data according to the structure defined in Figure 12.
  - The values of the bit parameters **shall** be set according to [2] and [3].
  - The LDS data block **shall** have top position in the RecordDataRead.Data field.

[1] For definition of structure and values of the slave list parameter List_wS0, see Annex B of Actor Sensor Interface – Profiles ([3])

[2] For definition of structure and values of the slave list parameter List, see Annex B of Actor Sensor Interface – Profiles ([3])

- **shall** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 10.
- **should** contain no further arguments in the RecordDataRead.Data field.

8.5.6.9 Get_Flags (Index 16/ 0x10)

The AS-i execution control function Get_Flags is used by the host interface in order to request that the execution control shall return the status of the execution control flags [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0010 addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the execution control flags as user data according to the structure defined in Table 12 and Figure 13.
  - o The values of the bit parameters **shall** be set according to [2] and [3].
  - o The Flags’ data block **shall** have top position in the RecordDataRead.Data field.
- **shall** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 10.
- **should** contain no further arguments in the RecordDataRead.Data field.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	Flags[3]							
3								

Figure 13: Structure of RecordDataRead.Data: Flags

Byte №	Bit №	Flag name *(Ordering according to* [3]*)*
2	0	Config_OK
	1	LDS.0
	2	Auto_Address_Assign
	3	Auto_Address_Available
	4	Configuration_Active
	5	Normal_Operation_Active
	6	APF/not APO
	7	Offline_Ready
3	0	Periphery_OK
	1	Data_Exchange_Active
	2	Off-line

Table 12: Definition of AS-i parameter ‘Flags’: structure Ex_Control_Flags

8.5.6.10 Set_Operation_Mode (Index 17/ 0x0x11)

The AS-i execution control function Set_Operation_Mode is used by the host interface in order to request that the execution control shall set the mode of operation of the execution control (protected mode or configuration mode) [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0011 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the requested mode change in the RecordDataWrite.Data field according to the structure defined in Figure 14.
  - o The parameter ‘Flag’ **shall** be described by: 0_{bin}: Change to protected mode
    1_{bin}: Change to configuration mode
  - o The Flag data block **shall** have top position in the RecordDataWrite.Data field.

[3] Description of the associated structure Ex_Control_Flags as per [2] and [3], see Table 12

- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** be sent after the flag Configuration_Active is modified in the non-volatile memory.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	0	0	0	0	0	0	0	Flag
3	Reserved = 0x00							

Figure 14: Structure of RecordDataWrite/ RecordDataRead.Data: modes and flags

8.5.6.11 Set_Offline_Mode (Index 18/ 0x0x12)

The AS-i execution control function Set_Offline_Mode is used by the host interface in order to request that the execution control shall leave or change to offline phase [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0012 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the requested mode change in the RecordDataWrite.Data field according to the structure defined in Figure 14.
  - o The parameter 'Flag' **shall** be described by: 0_{bin}: Leave offline phase; 1_{bin}: Change to offline phase
  - o The Flag data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.5.6.12 Set_Data_Exchange_Active (Index 19/ 0x0x13)

The AS-i execution control function Set_Data_Exchange_Active is used by the host interface in order to request that the execution control shall set the flag Data_Exchange_Active that starts or stops the cyclic data exchange during normal operation [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0013 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the requested mode change in the RecordDataWrite.Data field according to the structure defined in Figure 14.
  - o The parameter 'Flag' **shall** describe by: 0_{bin}: Reset flag Data_Exchange_Active; 1_{bin}: Set flag Data_Exchange_Active
  - o The Flag data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** be sent after the flag Data_Exchange_Active is modified in the non-volatile memory.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.5.6.13 Change_Slave_Address (Index 20/ 0x0x14)

The AS-i execution control function Change_Slave_Address is used by the host interface in order to request that the execution control shall change a specific slave address to a new address [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0014 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the requested mode change in the RecordDataWrite.Data field according to the structure defined in Figure 15.
    - o The Addresses' data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrtite.Data field.

The response

- **shall** be sent after the slave address has been changed.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	Old AS-i slave address							
3	New AS-i slave address							

Figure 15: Structure for changing slave address in mapping AS-i execution control functions

8.5.6.14 Set_Auto_Address_Enable (Index 21/ 0x0x15)

The AS-i execution control function Set_Auto_Address_Enable is used by the host interface in order to request that the execution control shall set the automatic address assignment active or inactive [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0015 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the requested mode change in the RecordDataWrite.Data field according to the structure defined in Figure 14.
    - o Parameter 'Flag' **shall** be described by: 0_{bin}: Automatic address assignment inactive
      1_{bin}: Automatic address assignment active
    - o The Flag data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments.

The response

- **should** be sent after the Auto_Address_Enable-bit is modified in the non-volatile memory.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.5.6.15 Get_Auto_Address_Enable (Index 21/ 0x0x15)

The AS-i execution control function Get_Auto_Address_Enable is used by the host interface in order to request whether the automatic address assignment is active or inactive [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0015 addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the desired mode change as user data according to the structure defined in Figure 14.
    - o Parameter 'Flag' **shall** be described by: 0_{bin}: Automatic address assignment inactive
      1_{bin}: Automatic address assignment active
    - o The Flag data block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **shall** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 10.

8.5.6.16 Execute_Command (Index 22/ 0x0x16)

The data record index 22 is reserved for implementations of AS-i function 22 according to each manufacturer's needs. The usage of both RecordDataRead and RecordDataWrite **shall** be restricted to implementations of Execute_Command primitives.

8.5.6.17 Get_LPF (Index 23/ 0x0x17)

The function is used by the host interface in order to request that the execution control shall return the values from the list of periphery faults (LPF) [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0017 addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the LPF as user data according to the structure defined in Figure 12.
  - The values of the bit parameters **shall** be set according to [2] and [3].
  - The LPF data block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **shall** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 10.

8.5.6.18 Write_Extended_ID-Code_1 (Index 24/ 0x0x18)

The function is used by the host interface in order to change the Extended ID-Code 1 of a slave with zero address [2].

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0018 addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the ID1 code as user data according to the structure defined in Figure 16.
  - The values of the bit parameters **shall** be set according to [2] and [3].
  - The Extended_ID-Code_1 block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** contain the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

Byte №	
0	Major Version = 0x00
1	Minor Version = 0x00
2	ID1 code
3	Reserved

Figure 16: Structure of Extended_ID-Code_1

8.5.6.19 Read_AIDI (Index 25/ 0x0x19)

This function is used to read the analogue input data provided by analogue slaves, which are supported by the master.

This AS-i function should be mapped to a PROFINET IO service RecordDataRead/ index 0x0019 addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** contain the AS-i result argument ‘AImage’ – i.e. the AIDI – as user data.
  - The AIDI structure ‘Analog_Input_Data_Image’ **shall** be composed according to [3], Table A5-7, details see Figure 17.
  - The AIDI data block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **shall** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 10.

Byte №	
0	Major Version = 0x00
1	Minor Version = 0x00
2	Analogue input data slave 1, channel 0 or analogue input data slave 1A, channel 0
3	
4	Analogue input data slave 1, channel 1 or analogue input data slave 1A, channel 1
5	
6	Analogue input data slave 1, channel 2 or analogue input data slave 1B, channel 0
7	
8	Analogue input data slave 1, channel 3 or analogue input data slave 1B, channel 1
9	
10	Analogue input data slave 2, channel 0 or analogue input data slave 2A, channel 0
11	
12	Analogue input data slave 2, channel 1 or analogue input data slave 2A, channel 1
13	
14	Analogue input data slave 2, channel 2 or analogue input data slave 2B, channel 0
15	
16	Analogue input data slave 2, channel 3 or analogue input data slave 2B, channel 1
16	
242	Analogue input data slave 31, channel 0 or analogue input data slave 31A, channel 0
243	
244	Analogue input data slave 31, channel 1 or analogue input data slave 31A, channel 1
245	
246	Analogue input data slave 31, channel 2 or analogue input data slave 31B, channel 0
247	
248	Analogue input data slave 31, channel 3 or analogue input data slave 31B, channel 1
249	
250	Reserved
251	

Figure 17: Analog data image (AIDI or AODI) of an extended AS-i master

8.5.6.20 Write_AODI (Index 26/ 0x0x1A)

This function is used by the host interface to provide analogue output data for the analogue slaves, which are supported by the master.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x001A addressed to subslot 1 of the appropriate AS-i master proxy module.

The request

- **shall** contain the AS-i parameter ‘AImage’ – i.e. the AODI – in the RecordDataWrite.Data field.
  - The AIDI structure ‘Analog_Output_Data_Image’ **shall** be composed according to [3], details see Figure 17.
  - The AIDI data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

8.5.7 Diagnosis Record Data (Index 96/ 0x60)

This service provides statistic counters to the host interface of to an engineering system.

The service **may** be mapped to a RecordDataRead/ index 0x0060 addressed to subslot 1 of the appropriate AS-i master proxy module.

The response

- **shall** provide the Parameters defined in Figure 18 at the described offset.
- **may** contain further arguments after the unit counter “Faulty master telegrams”.
  - The reserved field **may** be used for vendor specific purposes.

- o An advanced data structure longer than 32 bytes **should** be aligned to a double word.
- – **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	AS-i power fails						high byte	
3							low byte	
4	Ground faults						high byte	
5							low byte	
6	Unit counter: Slave failures						high word / high byte	
7							high word / low byte	
8							low word / high byte	
9							low word / low byte	
10	Unit counter: Missing slave telegrams						high word / high byte	
11							high word / low byte	
12							low word / high byte	
13							low word / low byte	
14	Unit counter: Faulty slave telegrams						high word / high byte	
15							high word / low byte	
16							low word / high byte	
17							low word / low byte	
18	Unit counter: Slave periphery faults						high word / high byte	
19							high word / low byte	
20							low word / high byte	
21							low word / low byte	
22	Unit counter: Slave protocol errors						high word / high byte	
23							high word / low byte	
24							low word / high byte	
25							low word / low byte	
26	Unit counter: Faulty master telegrams						high word / high byte	
27							high word / low byte	
28							low word / high byte	
29							low word / low byte	
30	Reserved (vendor specific)							
31								

Figure 18: Structure of RecordDataRead.Data: Accumulative diagnoses of an AS-i master proxy

8.6 AS-i Slave Modules

8.6.1 Modelling of AS-i Slaves

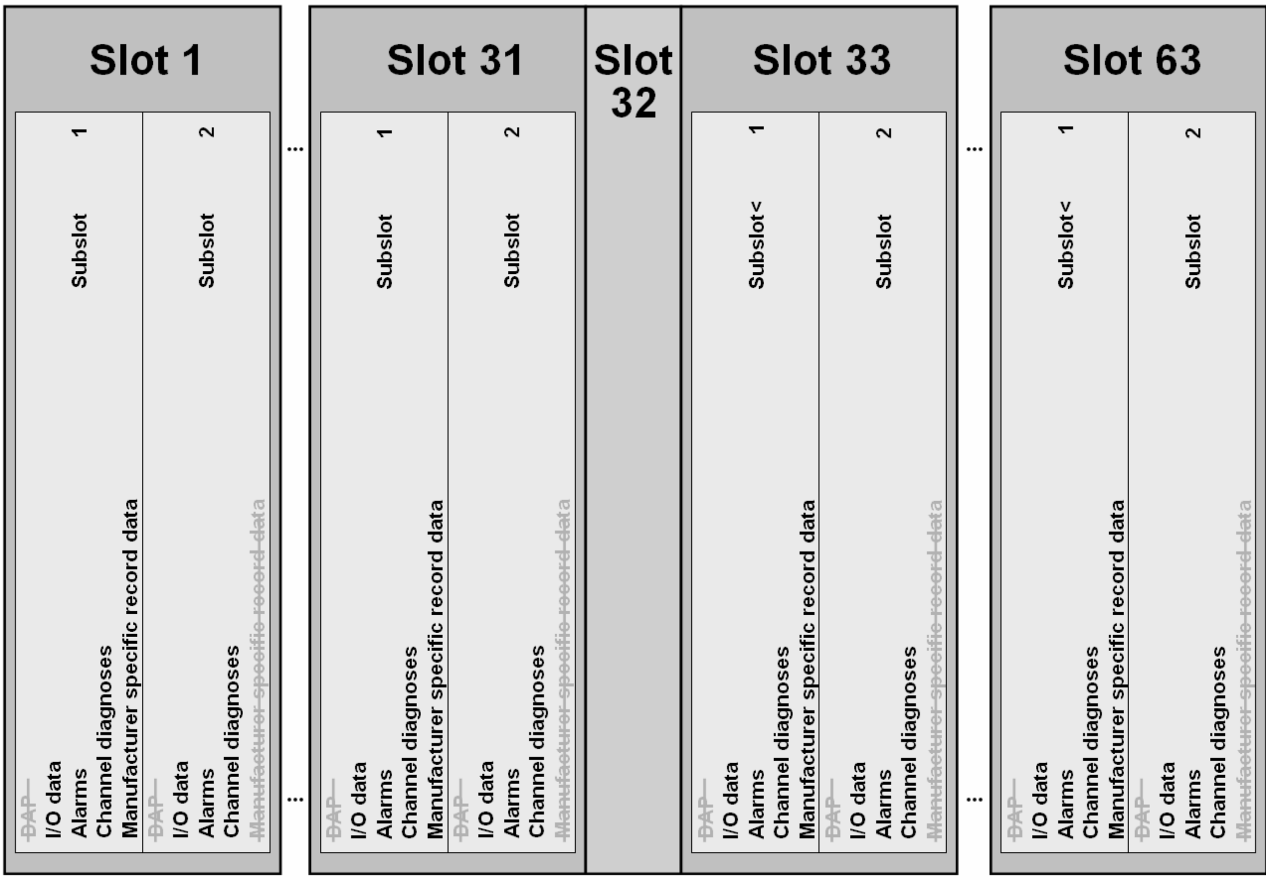

Figure 19: Modelling of the submodules of AS-i slaves at the 1st line

The mapping of the AS-i line's masters to PROFINET IO modules is defined in Table 2. The submodules of each AS-i slave **should** be restricted to two in maximum. The submodules **shall** be fixed to ascending subslots starting with 1. Each subslot **should** represent either the slave's set of channels with data mode "binary" or the set of channels with data mode "analogue".

For AS-i slave proxy modules the parameter values of the structure "ModuleID" **should** be restricted to Type 2 (AS-i slave proxy) in Table 4.

Also for AS-i slave proxy modules the parameter values of the structure "SubmoduleID" **should** be restricted to the Types 2 and 3 (binary data and analogue data) in Table 5.

The submodules of the AS-i master proxies **shall** support elements of following PROFINET IO ASEs:

- I/O data — details see 8.6.2, 8.6.3 and 8.6.4
- Alarm — details see 8.6.6
- Channel diagnosis — details see 8.6.7
- Data records — details see 8.6.8 and 8.6.9

8.6.2 Input Data and Status

The IOPS and the transfer values **shall** be set according to Table 13. The IOCS of the inputs **should** be ignored and as well a failure of the IO Controller and the APDU state. For outputs AS-i defines no substitute values, so ‘0’ **should** be reached.

Condition		**Reaction**	
Slave bit in LPS	**Slave bit in LAS**	**IOPS**	**Transfer value**
0	0	Good	Substitute 0
0	1	Good	Substitute 0
1	0	Bad	Substitute 0
1	1	Good	Original

Table 13: Composing IOPS and transfer values of the inputs

8.6.3 Output Data and Status

The IOCS of the outputs **shall** be set according to Table 14, the transfer values **should** be determined according to Table 15. The substitute values of output data are specified in [2]/ [3].

Condition		**Reaction**
Slave bit in LPS	**Slave bit in LAS**	**IOPS**
0	-	0
1	0	1
1	1	1

Table 14: Composing IOCS of the outputs

Condition		**Reaction**
IOPS	**Slave bit in LAS**	**Transfer value**
0	-	Substitute
1	0	Substitute
1	1	Original

Table 15: Transfer values of the outputs

8.6.4 Over-control and Under-control of I/Os

The term 'over-control' describes the usage of an I/O module that is oversized for the corresponding AS-i slave module. This allows reserving additional channels or additional I/Os of an existing channel for later usage together with a different AS-i device that will provide the additional I/Os. Table 16 defines to which modules over-control **should** be restricted.
Clause 3 (with Figure 20 and Figure 21) specifies the cast details that **shall** be applied for affected values.

Advised module[4]	**Allowed modules for over-control**				
0x0201 0001/ 0x0200 0100	0x0203 0001/ 0x0200 0101	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808		
0x0202 0001/ 0x0200 0001	0x0203 0001/ 0x0200 0101	0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808	
0x0203 0001/ 0x0200 0101	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808			
0x0204 0001/ 0x0300 0200	0x0206 0001/ 0x0300 0202	0x0207 0001/ 0x0300 0400	0x0209 0001/ 0x0300 0404	0x020A 0001/ 0x0300 0800	0x020C 0001/ 0x0300 0808
	0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808		
0x0205 0001/ 0x0300 0002	0x0206 0001/ 0x0300 0202	0x0208 0001/ 0x0300 0004	0x0209 0001/ 0x0300 0404	0x020B 0001/ 0x0300 0008	0x020C 0001/ 0x0300 0808
	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808			
0x0206 0001/ 0x0300 0202	0x0209 0001/ 0x0300 0004	0x020C 0001/ 0x0300 0808	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808	
0x0207 0001/ 0x0300 0400	0x0209 0001/ 0x0300 0404	0x020A 0001/ 0x0300 0800	0x020C 0001/ 0x0300 0808	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808
0x0208 0001/ 0x0300 0004	0x0209 0001/ 0x0300 0404	0x020B 0001/ 0x0300 0008	0x020C 0001/ 0x0300 0808	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808
0x0209 0001/ 0x0300 0404	0x020C 0001/ 0x0300 0808	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808		
0x020A 0001/ 0x0300 0800	0x020C 0001/ 0x0300 0808	0x0223 0002/ 0x0200 0101/ 0x0300 0808			
0x020B 0001/ 0x0300 0008	0x020C 0001/ 0x0300 0808	0x0223 0002/ 0x0200 0101/ 0x0300 0808			
0x020C 0001/ 0x0300 0808	0x0223 0002/ 0x0200 0101/ 0x0300 0808				

[4] Representation: ModuleID/ SubmoduleID in subslot 1/ SubmoduleID in subslot 2. The IDs of related submodules are framed by boxes of similar type (i.e.: box type 1 – box type 2).

Advised module[4]	**Allowed modules for over-control**				
0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808			
0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0223 0002/ 0x0200 0101/ 0x0300 0808				
0x0240 0001/ 0x0100 0000	All modules of Table 32 permitted				

Table 16: Allowed modules for over-control

The term 'under-control' describes the usage of an I/O module that is undersized in respect to the corresponding AS-i slave module's I/Os. This allows fading out unused channels or unused I/Os of an existing channel in order to save I/O memory of the controller. Table 17 defines to which modules under-control **should** be restricted.

Clause 3 (with Figure 20 and Figure 21) specifies the cast details that **shall** be applied for affected values.

Advised module[4]	**Allowed modules for under-control**					
0x0203 0001/ 0x0200 0101	0x0201 0001/ 0x0200 0100	0x0202 0001/ 0x0200 0001	0x0240 0001/ 0x0100 0000			
0x0206 0001/ 0x0300 0202	0x0204 0001/ 0x0300 0200	0x0205 0001/ 0x0300 0002	0x0240 0001/ 0x0100 0000			
0x0207 0001/ 0x0300 0400	0x0204 0001/ 0x0300 0200	0x0240 0001/ 0x0100 0000				
0x0208 0001/ 0x0300 0004	0x0205 0001/ 0x0300 0002	0x0240 0001/ 0x0100 0000				
0x0209 0001/ 0x0300 0404	0x0204 0001/ 0x0300 0200	0x0205 0001/ 0x0300 0002	0x0206 0001/ 0x0300 0202	0x0207 0001/ 0x0300 0400	0x0208 0001/ 0x0300 0004	0x0240 0001/ 0x0100 0000
0x020A 0001/ 0x0300 0800	0x0204 0001/ 0x0300 0200	0x0207 0001/ 0x0300 0400	0x0240 0001/ 0x0100 0000			
0x020B 0001/ 0x0300 0008	0x0205 0001/ 0x0300 0002	0x0208 0001/ 0x0300 0004	0x0240 0001/ 0x0100 0000			
0x020C 0001/ 0x0300 0808	0x0204 0001/ 0x0300 0200	0x0205 0001/ 0x0300 0002	0x0206 0001/ 0x0300 0202	0x0207 0001/ 0x0300 0400	0x0208 0001/ 0x0300 0004	0x0209 0001/ 0x0300 0404
	0x020A 0001/ 0x0300 0800	0x020B 0001/ 0x0300 0008	0x0240 0001/ 0x0100 0000			
0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0202 0001/ 0x0200 0001	0x0204 0001/ 0x0300 0200	0x0240 0001/ 0x0100 0000			
0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0201 0001/ 0x0200 0100	0x0202 0001/ 0x0200 0001	0x0203 0001/ 0x0200 0101	0x0204 0001/ 0x0300 0200	0x0205 0001/ 0x0300 0002	0x0206 0001/ 0x0300 0202
	0x0207 0001/ 0x0300 0400	0x0208 0001/ 0x0300 0004	0x0209 0001/ 0x0300 0404	0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0240 0001/ 0x0100 0000	
0x0223 0002/ 0x0200 0101/ 0x0300 0808	0x0201 0001/ 0x0200 0100	0x0202 0001/ 0x0200 0001	0x0203 0001/ 0x0200 0101	0x0204 0001/ 0x0300 0200	0x0205 0001/ 0x0300 0002	0x0206 0001/ 0x0300 0202
	0x0207 0001/ 0x0300 0400	0x0208 0001/ 0x0300 0004	0x0209 0001/ 0x0300 0404	0x020A 0001/ 0x0300 0800	0x020B 0001/ 0x0300 0008	0x020C 0001/ 0x0300 0808

Advised module[4]	Allowed modules for under-control					
	0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0222 0002/ 0x0200 0101/ 0x0300 0404	0x0240 0001/ 0x0100 0000			

Table 17: Allowed modules for under-control

Configuration with a different AS-i device, which will be replaced at a later point of time, makes it necessary to use mixed over-controlled or under-controlled modules. Table 18 defines to which modules with mixed over-control and under-control **should** be restricted. Clause 3 (with Figure 20 and Figure 21) specifies the cast details that **shall** be applied for affected values.

Advised module[4]	Allowed modules for mixed over-control and under-control	
0x0203 0001/ 0x0200 0101	0x0221 0002/ 0x0200 0001/ 0x0300 0200	
0x0206 0001/ 0x0300 0202	0x0221 0002/ 0x0200 0001/ 0x0300 0200	
0x0207 0001/ 0x0300 0400	0x0221 0002/ 0x0200 0001/ 0x0300 0200	
0x0209 0001/ 0x0300 0404	0x0221 0002/ 0x0200 0001/ 0x0300 0200	
0x020A 0001/ 0x0300 0800	0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0222 0002/ 0x0200 0101/ 0x0300 0404
0x020B 0001/ 0x0300 0008	0x0222 0002/ 0x0200 0101/ 0x0300 0404	
0x020C 0001/ 0x0300 0808	0x0221 0002/ 0x0200 0001/ 0x0300 0200	0x0222 0002/ 0x0200 0101/ 0x0300 0404

Table 18: Allowed modules for mixed usage of over-control and under-control

Clause 3:

Figure 20 and Figure 21 show how newly-created and suppressed channels **should** behave in case of over-control and under-control. Newly-created input submodules or I/Os within an existing submodule, which have no corresponding real AS-i slave I/Os behave as follows:

- Inputs **shall** be filled with the default values according to [2] and [3] respectively.
- Outputs **should** be filled with the default values according to [2] and [3] respectively.

Values of newly-created output submodules without corresponding slave outputs **should** be ignored by the linking device.

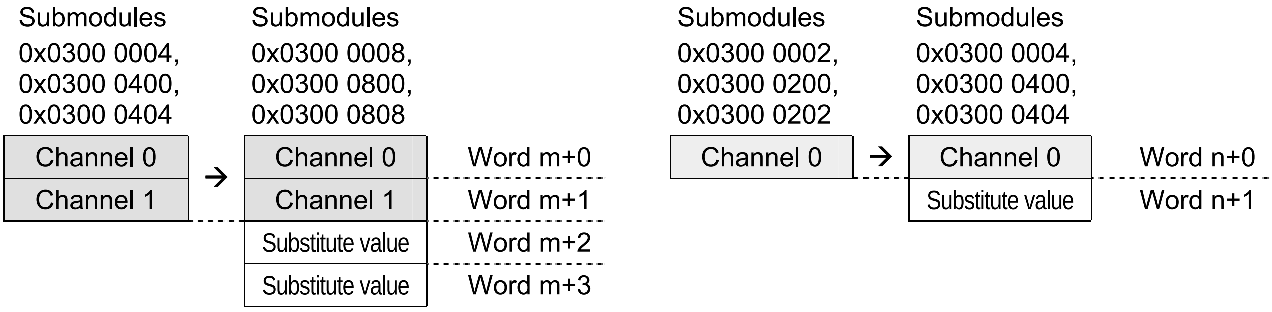

Figure 20: Over-control of analogue input or output modules - newly-created channels

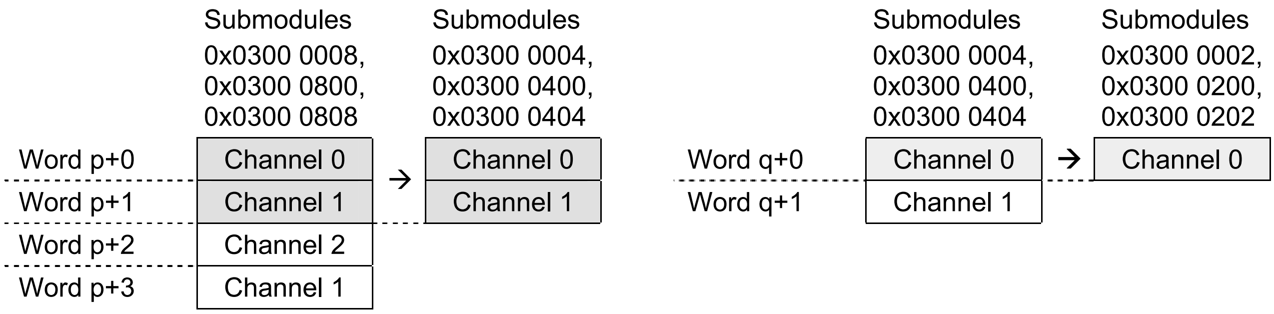

Figure 21: Under-control of analogue input or output modules - suppressed channels

8.6.5 Input/ Output Mapping of The AS-i Slave Profiles

8.6.5.1 Standard Slave Profile

This chapter states the advised I/O modules for AS-i slaves of the standard slave profiles.

N.B! For all of these modules there is the possibility of over-control and under-control. When establishing a PROFINET Application Relation (AR) both the modules advised by Figure 15 advised modules and the modules adapted according to chapter 8.6.4 **shall** be accepted.

I/O Code	ID Code	2	Description	Module ID	Submodule ID	Sub slot
0	C	E, F	Remote I/Os	0x0201 0001	0x0200 0100	1
	1	E, F	Remote I/Os with dual-signals			
	A	0, 2, E, F	Slaves in extended addressing mode			
	B	0, 1 E, F	Safety slaves			
	F	E, F	Free profiles			
1	0	E, F	Remote I/Os	0x0203 0001	0x0200 0101	1
	1	E, F	Single sensor with extended control			
	A	0, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			
2	0	E, F	Remote I/Os	0x0203 0001	0x0200 0101	1
	F	E, F	Free profiles			
3	0	E, F	Remote I/Os	0x0203 0001	0x0200 0101	1
	1	E, F	Remote I/Os with dual-signals			
	A	0,1,2,E,F	Free profiles in extended addressing mode			
	F	E, F	Free profile			
4	0	E, F	Remote I/Os	0x0203 0001	0x0200 0101	1
	A	0, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			
5	0	E, F	Remote I/Os	0x0203 0001	0x0200 0101	1
	A	0, E	Free profiles in extended addressing mode			
	F	E, F	Free profiles			

I/O Code	ID Code	2	Description	Module ID	Submodule ID	Sub slot
6	A	0, E	Free profiles in extended addressing mode	0x0203 0001	0x0200 0101	1
	F	E, F	Free profiles			
7	0	E, F	Remote I/Os	0x0203 0001	0x0200 0101	1
	A	0, 2, E, F	Free profiles in extended addressing mode			
	B	0, 1, E, F	Safety slaves			
	D	0, .., 6, E, F	Motor control devices			
	E	0, .., 6, E, F	Motor control devices			
	F	E, F	Free profiles			
8	0	E, F	Remote I/Os	0x0202 0001	0x0202 0001	1
	1	E, F	Remote I/Os with dual-signals			
	A	0, 2, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			
9	A	2, E, F	Free profiles in extended addressing mode	0x0202 0001	0x0202 0001	1
	F	E, F	Free profiles			
A	0	E, F	Remote I/Os	0x0202 0001	0x0202 0001	1
	F	E, F	Free profiles			
B	1	E, F	Dual actor with feedback	0x0202 0001	0x0202 0001	1
	A	0, 2, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			
C	0	E, F	Remote I/Os	0x0202 0001	0x0202 0001	1
	A	0, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			
D	1	E, F	Single actor with monitoring	0x0202 0001	0x0202 0001	1
	A	0, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			
E	0	E, F	Remote I/Os	0x0202 0001	0x0202 0001	1
	A	0, E, F	Free profiles in extended addressing mode			
	F	E, F	Free profiles			

Table 19: Correlation of ModuleID/ SubmoduleID to AS-i standard slave profiles

The slave's binary I/O data **shall** be located in the PROFINET IO frame for cyclic data according to Figure 22. The cancelled input bits and also those inputs that are unsupported by some profiles **should** be filled with substitute values according to [2] and [3].

Module / Provider	Byte address	Affected PROFINET frame sections[5]							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Subslot 1/ Inputs, IO Device	n + 0	X	X	X	X	D 4	D 3	D 2	D 1
Subslot 1/ Outputs, IO Controller	m + 0	X	X	X	X	D 4	D 3	D 2	D 1

Figure 22: Mapping of AS-i I/O data bits to PROFINET IO frame bytes for standard slave profiles

8.6.5.1.1 Optional Channel Diagnosis with AS-i Profile „Single Actuator with monitoring" (S-D.1)

Optionally a channel diagnosis **may** be generated for slave proxy modules of slave profile S-D.1 on change of input bit D2 or D3 according to Table 20. The diagnosis **should** only be provided in addition to the cyclic inputs.

[5] The input or output frame byte exists only if the chosen module/ submodule combination defines corresponding input or output data. Otherwise the substitution strategy defined in Table 16, Table 17 and Table 18 **should** be applied.

Error Type	Meaning	Condition	Diagnosis	Subslot
0x0009	Indicates a fault condition	Bit D2 switches from 0 to 1	Appearing	1
		Bit D2 switches from 1 to 0	Disappearing	
0x0018	Monitors the availability of the actuator	Bit D3 switches from 0 to 1	Appearing	1
		Bit D3 switches from 1 to 0	Disappearing	

Table 20: Error types of optional channel diagnosis for slaves of profile S-D.1

8.6.5.2 Combined Transaction Type 1 Profiles

This chapter states the advised I/O modules for AS-i slaves of the CTT 1 profiles.

N.B! For all of these modules there is the possibility of over-control and under-control. When establishing a PROFINET Application Relation (AR) both the modules advised by Table 21 advised modules and the modules adapted according to chapter 8.6.4 **shall** be accepted.

These slave profiles are designed to support the transfer of analogue values in the firmware of AS-Interface Masters [3]. The advised PROFINET IO modules of them consist of a single submodule specially designed for analogue values (IDs beginning with 0x0300...).

Furthermore CTT 1 profiled slaves allow the data exchange of digital two byte values with sensors and actuators for which also the analogue submodules are advised. Alternatively by means of CTT 1 slaves are defined with four bit mode (4I/4O). These I/Os are linked to the host interface of the PROFINET IO Controller by a binary module (ID beginning with 0x0200…). In addition profile S-7.4 offers extended functions, among other things for loading parameter sets into the slave in order to handle more complex slaves. These services **should** be provided as outlined in the chapters 8.6.9.8 through 8.6.9.11:

Read_Parameter_String (Index 27/ 0x0x1B), Write_Parameter_String (Index 28/ 0x0x1C), Read_Diagnostic_String (Index 29/ 0x0x1D) and Read_Identification_String (Index 30/ 0x0x1E)).

I/O Code	ID Code			Description	Module ID	Submodule ID	Sub Slot
		2	1				
7	3	0		CTT 1, transparent mode, 1 channel out	0x0205 0001	0x0300 0200	1
		1		CTT 1, transparent mode, 2 channels out	0x0208 0001	0x0300 0004	1
		2		CTT 1, transparent mode, 4 channels out	0x020B 0001	0x0300 0008	1
		4		CTT 1, analogue values, 1 channel out	0x0205 0001	0x0300 0002	1
		5		CTT 1, analogue values, 2 channels out	0x0208 0001	0x0300 0004	1
		6		CTT 1, analogue values, 4 channels out	0x020B 0001	0x0300 0008	1
		8		CTT 1, transparent mode, 1 channel in	0x0204 0001	0x0300 0200	1
		9		CTT 1, transparent mode, 2 channels in	0x0207 0001	0x0300 0400	1
		A		CTT 1, transparent mode, 4 channels in	0x020A 0001	0x0300 0800	1
		C		CTT 1, analogue values, 1 channel in	0x0204 0001	0x0300 0200	1
		D		CTT 1, analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
		E		CTT 1, analogue values, 4 channels in	0x020A 0001	0x0300 0800	1
		F	F				
	4	0		CTT 1 extended, 4 bit mode (4I/ 4O)	0x0203 0001	0x0200 0101	1
		4		CTT 1 extended, 1 channel out	0x0205 0001	0x0300 0002	1
		5		CTT 1 extended, 2 channels out	0x0208 0001	0x0300 0004	1
		6		CTT 1 extended, 4 channels out	0x020B 0001	0x0300 0008	1
		C		CTT 1 extended, 1 channels in	0x0204 0001	0x0300 0200	1
		D		CTT 1 extended, 2 channels in	0x0207 0001	0x0300 0400	1
		E		CTT 1 extended, 4 channel in	0x020A 0001	0x0300 0800	1
		F	F				

Table 21: Correlation of ModuleID/ SubmoduleID to Combined Transaction Type 1

The slave's binary I/O data **shall** be located in the PROFINET IO frame for cyclic data according to Figure 23. Unsupported input channels, the profile respecting, **shall** lead to substitute values according to [2] and

[3]. The identification of the bits in Figure 23 (E1, E2, E3, and D1 … D16) refers to the wording of chapters 4.4.3.5 and 4.4.4.5 in [3]. The value representation **shall** be host level accordingly.

Module / Provider	Byte address	Channel			Affected PROFINET frame sections							
		E3	E2	E1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Subslot 1	m + 0	0	0	0	D8	D7	D6	D5	D4	D3	D2	D1
	m + 1				D16	D15	D14	D13	D12	D11	D10	D9
	m + 2	0	0	1	D8	D7	D6	D5	D4	D3	D2	D1
	m + 3				D16	D15	D14	D13	D12	D11	D10	D9
	m + 4	0	1	0	D8	D7	D6	D5	D4	D3	D2	D1
	m + 5				D16	D15	D14	D13	D12	D11	D10	D9
Inputs	m + 6	0	1	1	D8	D7	D6	D5	D4	D3	D2	D1
IO Device	m + 7				D16	D15	D14	D13	D12	D11	D10	D9
Subslot 1	n + 0	0	0	0	D8	D7	D6	D5	D4	D3	D2	D1
	n + 1				D16	D15	D14	D13	D12	D11	D10	D9
	n + 2	0	0	1	D8	D7	D6	D5	D4	D3	D2	D1
	n + 3				D16	D15	D14	D13	D12	D11	D10	D9
	n + 2	0	1	0	D8	D7	D6	D5	D4	D3	D2	D1
	n + 3				D16	D15	D14	D13	D12	D11	D10	D9
Outputs	n + 2	0	1	1	D8	D7	D6	D5	D4	D3	D2	D1
IO Controller	n + 3				D16	D15	D14	D13	D12	D11	D10	D9

Figure 23: Mapping of AS-i I/O data bits to PROFINET IO frame bytes for CTT 1 profiles

8.6.5.3 Combined Transaction Type 2 Profiles

This chapter states the advised I/O modules for AS-i slaves of the CTT 2 profiles.
N.B! For all of these modules there is the possibility of over-control and under-control. When establishing a PROFINET Application Relation (AR) both the modules advised by Table 22 and the modules adapted according to chapter 8.6.4 **shall** be accepted.

These profiles are intended for “combined field devices”, i.e. AS-i slaves that have digital inputs/outputs as well as serial inputs/outputs and for “serial field devices” that have only serial inputs/outputs [3].

- For the combined filed devices the advised modules are made up of a binary submodule located in subslot 1 and an analogue submodule in subslot 2.
- The pure serial field devices are linked to the IO Controller’s host interface by modules consisting of a single analogue submodule with suitable I/O lengths.

Beside of the exchange of input and/or output data the serial interface may be used additionally for device identification, complex configuration and parameter data, detailed diagnosis information. This interface **should** be provided by record data services, described in chapter 8.6.10.

I/O Code	ID Code		Description	Module ID	Submodule ID	Sub slot
		2				
7	5	5	CTT 2, 1-4 transparent/ analogue channels in/ out and 2 bits in/ out	0x0223 0002	0x0200 0101	1
					0x0300 0808	2
	A	5	CTT 2, 1-2 transparent/ analogue channels in/ out and 2 bits in/ out	0x0222 0002	0x0200 0101	1
					0x0300 0404	2
B	A	5	CTT 2, 2 analogue channels in/ out	0x0209 0001	0x0300 0404	1

Table 22: Correlation of ModuleID/ SubmoduleID to Combined Transaction Type 2

The slave’s binary I/O data **shall** be located in the PROFINET IO frame for cyclic data according to Figure 24. The cancelled binary input bits **should** be set to substitute values according to [2] an [3].

Note to Figure 24: The 6th to 9th serial input and output data bytes or rather the channels 2 and 3 are only available in S-7.5.5 that is working in standard address mode. Furthermore the binary output bit D3 is not available in S-7.A.5.

Module[6] / Provider	Byte address	Channel	Affected PROFINET frame sections							
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Subslot/ Inputs	p + 0								D1	D0
Binary Subslot/ Outputs	q + 0						D3	D2		
Analogue subslot / Inputs IO Device	m + 0	0	2nd byte of serial data out[7]							
	m + 1		3rd byte of serial data out							
	m + 2	1	4th byte of serial data out							
	m + 3		5th byte of serial data out							
	m + 4	2	6th byte of serial data out							
	m + 5		7th byte of serial data out							
	m + 6	3	8th byte of serial data out							
	m + 7		9th byte of serial data out							
Analogue subslot / Outputs IO Controller	n + 0	0	2nd byte of serial data out[7]							
	n + 1		3rd byte of serial data out							
	n + 2	1	4th byte of serial data out							
	n + 3		5th byte of serial data out							
	n + 2	2	6th byte of serial data out							
	n + 3		7th byte of serial data out							
	n + 2	3	8th byte of serial data out							
	n + 3		9th byte of serial data out							

Figure 24: Mapping of AS-i I/O data bits to PROFINET IO frame bytes for CTT 2 profiles

8.6.5.4 Combined Transaction Type 3 Profile

This chapter states the advised I/O modules for AS-i slaves of the CTT 3 profiles.
N.B! For all of these modules there is the possibility of over-control and under-control. When establishing a PROFINET Application Relation (AR) both the modules advised by Table 23 and the modules adapted according to chapter 8.6.4 **shall** be accepted.

This slave profile is intended for 4I/4O and 8I/8 slaves in extended addressing mode [3] with modules consisting of single binary or analogue submodules respectively.

I/O Code	ID Code			Description	Module ID	Submodule ID	Sub slot
		2	1				
7	A	7	7	CTT 3, 4I/ 4O	0x0203 0001	0x0200 0101	1
		A	6	CTT 3, 8I/ 8O non consistent input & output	0x0206 0001	0x0300 0202	1
			7	CTT 3, 8I/ 8O consistent input and output			

Table 23: Correlation of ModuleID/ SubmoduleID to Combined Transaction Type 3

The slave's binary I/O data **shall** be located in the PROFINET IO frame for cyclic data according to Figure 25. The cancelled input bits and also the inputs D4 to D7 **shall** be filled with substitute values, the

---

[6] The binary subslot is subslot 1 for S-7.5.5 and S-7.A.5 and does not exist for S-B.A.5. The analogue subslot is subslot 1 for S-B.A.5 and subslot 2 for S-7.5.5 and S-7.A.5.

[7] The 1st byte of serial data with the command/ response code is not transferred in the PROFINET frame.

bits that are unsupported in S-7.A.A **should** be filled with substitute values according to [2] an [3]. The identification of the bits in and Figure 25 (DI0 … DI7 and DO0 … DO7) refers to the wording of chapter 4.4.6.3 in [2].

Note to Figure 25, Figure 24: The bits DI4 to DI7 and DO4 to DO7 are only available in S-7.A.7.

Module / Provider	Byte address	Affected PROFINET frame sections							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Subslot 1/ Inputs, IO Device	m + 0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	m + 1								
Subslot 1/ Outputs, IO Controller	n + 0	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	n + 1								

Figure 25: Mapping of AS-i I/O data bits to PROFINET IO frame bytes for CTT 3 profiles

8.6.5.5 Combined Transaction Type 4 Profile

This chapter states the advised I/O modules for AS-i slaves of the CTT 4 profiles.
N.B! For all of these modules there is the possibility of over-control and under-control. When establishing a PROFINET Application Relation (AR) both the by Table 24 advised modules and the according to chapter 8.6.4 adapted modules **shall** be accepted.

This profile offers the possibility of transmitting of up to 16 bit data from the slave in extended addressing mode to the master. One output bit may be used as binary output of a slave or to switch over between two input channels [3]. Advised grades are modules combined of a binary and an analogue submodule for slaves with an additional output and modules with a single analogue module for the latter.

I/O Code	ID Code			Description	Module ID	Submodule ID	Sub slot
		2	1				
7	A	8	3	CTT 4, 8 bits transparent data, 1 channel in and 1 bit out	0x0221 0002	0x0200 0001	1
						0x0300 0200	2
			4	CTT 4, 12 bits transparent data, 1 channel in and 1 bit out	0x0221 0002	0x0200 0001	1
						0x0300 0200	2
			5	CTT 4, 16 bits transparent data, 1 channel in and 1 bit out	0x0221 0002	0x0200 0001	1
						0x0300 0200	2
			6	CTT 4, 12 bits transparent data, 1 channel in and 1 bit out	0x0221 0002	0x0200 0001	1
						0x0300 0200	2
			7	CTT 4, 14 bits transparent data, 1 channel in and 1 bit out	0x0221 0002	0x0200 0001	1
						0x0300 0200	2
		9	0	CTT 4, 14 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			1	CTT 4, 12 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			2	CTT 4, 14 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			3	CTT 4, 14 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			4	CTT 4, 14 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			5	CTT 4, 14 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			6	CTT 4, 12 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1
			7	CTT 4, 14 bits analogue values, 2 channels in	0x0207 0001	0x0300 0400	1

Table 24: Correlation of ModuleID/ SubmoduleID to Combined Transaction Type 4

The slave's binary I/O data **shall** be located in the PROFINET IO frame for cyclic data according to Figure 26. The cancelled input bits **should** be filled with substitute values according to [2] an [3]. The identification of the bits in Figure 26 (D2, DI0 … D15) refers to the wording of chapters 5.7.5.1 [2].

Note to Figure 26: The binary output bit I2 and so the total binary subslot **should** not available in S-7.A.9. Furthermore only in S-7.A.9 two channels are available. For the profile S-7.A.8 the bits of channel 1 will not apply.

Module / Provider	Byte address	Channel D2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary Subslot/ Out	m + 0							I2		
Analogue subslot / Inputs IO Device	n + 0	0	DI7	DI6	DI5	DI4	DI3	DI2	DI2	DI0
	n + 1		DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
	n + 2	1	DI7	DI6	DI5	DI4	DI3	DI2	DI2	DI0
	n + 3		DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8

Figure 26: Mapping of AS-i I/O data bits to PROFINET IO frame bytes for CTT 4 profiles

8.6.5.6 Combined Transaction Type 5 Profile

This chapter states the advised I/O modules for AS-i slaves of the CTT 5 profiles.
N.B! For all of these modules there is the possibility of over-control and under-control. When establishing a PROFINET Application Relation (AR) both the modules advised by Table 25 and the modules adapted according to chapter 8.6.4 **shall** be accepted.

This profile describes a method of high speed transmission of 8, 12 or 16 bit bi-directional consistent data using 2, 3 or 4 consecutive slave addresses [3]. For each address an own module **shall** be used whereas the first of the consecutive modules **shall** keep the total data.

I/O Code	ID Code	2	Description	Module ID	Submodule ID	Sub slot
6	0	2	CTT 5, 8 bits transparent values	0x0206 0001	0x0300 0202	1
		5		0x0240 0001	0x0100 0000	1
6	0	A	CTT 5, 8 bits analogue values	0x0206 0001	0x0300 0202	1
		5		0x0240 0001	0x0100 0000	1
6	0	3	CTT 5, 12 bits transparent values	0x0206 0001	0x0300 0202	1
		6		0x0240 0001	0x0100 0000	1
		5		0x0240 0001	0x0100 0000	1
6	0	B	CTT 5, 12 bits analogue values	0x0206 0001	0x0300 0202	1
		6		0x0240 0001	0x0100 0000	1
		5		0x0240 0001	0x0100 0000	1
6	0	4	CTT 5, 16 bits transparent values	0x0206 0001	0x0300 0202	1
		7		0x0240 0001	0x0100 0000	1
		6		0x0240 0001	0x0100 0000	1
		5		0x0240 0001	0x0100 0000	1
6	0	C	CTT 5, 16 bits analogue values	0x0206 0001	0x0300 0202	1
		7		0x0240 0001	0x0100 0000	1
		6		0x0240 0001	0x0100 0000	1
		5		0x0240 0001	0x0100 0000	1

Table 25: Correlation of ModuleID/ SubmoduleID to Combined Transaction Type 5

The slave's binary I/O data **shall** be located in the PROFINET IO frame for cyclic data according to Figure 27 through Figure 29. The cancelled input bits **should** be filled with substitute values according to [2] an [3]. The identification of the bits in these figures (D0 … D15) refers to the wording "Data bit #" in chapters 4.4.8.5 and 4.4.8.6 of [3].

Module / Provider	Byte address	Affected PROFINET frame sections							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AS-i address									
Subslot 1/ IO Device inputs	m + 0								
AS-i address		α_1 +1				α_1			
Subslot 1/ IO Device inputs	m + 1	D15	D14	D13	D12	D11	D10	D9	D8
AS-i address									
Subslot 1/ IO Device outputs	n + 0								
AS-i address		α_1 +1				α_1			
Subslot 1/ IO Device outputs	n + 1	D15	D14	D13	D12	D11	D10	D9	D8

Figure 27: Correlation ModuleID/ SubmoduleID to S-6.0.2/S-6.0.5 and S-6.0.A/S-6.0.5

Module / Provider	Byte address	Affected PROFINET frame sections							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AS-i address		α_1							
Subslot 1/ IO Device inputs	m + 0	D7	D6	D5	D4				
AS-i address		α_1 +2				α_1 +1			
Subslot 1/ IO Device inputs	m + 1	D15	D14	D13	D12	D11	D10	D9	D8
AS-i address		α_2							
Subslot 1/ IO Device outputs	n + 0	D7	D6	D5	D4				
AS-i address		α_2 +2				α_2 +1			
Subslot 1/ IO Device outputs	n + 1	D15	D14	D13	D12	D11	D10	D9	D8

Figure 28: Correlation ModuleID/ SubmoduleID to S-6.0.3/S-6.0.6/S-6.0.5 and S-6.0.B/S-6.0.6/S-6.0.5

Module / Provider	Byte address	Affected PROFINET frame sections							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AS-i address		α_1 +1				α_1			
Subslot 1/ IO Device inputs	m + 0	D7	D6	D5	D4	D3	D2	D1	D0
AS-i address		α_1 +3				α_1 +2			
Subslot 1/ IO Device inputs	m + 1	D15	D14	D13	D12	D11	D10	D9	D8
AS-i address		α_2 +1				α_2			
Subslot 1/ IO Device outputs	n + 0	D7	D6	D5	D4	D3	D2	D1	D0
AS-i address		α_2 +3				α_2 +2			
Subslot 1/ IO Device outputs	n + 1	D15	D14	D13	D12	D11	D10	D9	D8

Figure 29: Correlation ModuleID/ SubmoduleID to S-6.0.4/S-6.0.7/S-6.0.6/S-6.0.5 and S-6.0.C/S-6.0.7/S-6.0.6/S-6.0.5

8.6.6 Alarms

Table 26 defines the alarm types and its special usages for the AS-i integration that **shall** be provided by AS-i slave proxy modules. For information with under which conditions these alarms have to be initiated, see chapter 8.8.3.

Type	Subslot	Usage
Diagnosis	1	Signals the event of a channel diagnosis, see Table 27.
Process		No special use
Pull	1	An AS-i Slave proxy module signals the failure of a slave. The alarm affects all subslots of a module though signalled always at slot 1.
Plug	1	An AS-i Slave proxy module signals the return of a configured slave (and so a new need for parameterisation).
Status		No special use
Update		No special use
Redundancy		No special use
Controlled by supervisor		This alarm **shall** be implemented unless the linking device supports no Supervisor AR
Released		This alarm **shall** be implemented unless the linking device supports no Supervisor AR
Plug Wrong Submodule	1	An AS-i Slave proxy module signals the return of a not configured slave.
Return of Submodule	1, 2	An AS-i Slave proxy module signals that a submodule is ready to switch its IOCS/IOPS from “BAD” to “GOOD” again without new parameterisation.
Diagnosis disappears	1	Signals a disappearing diagnosis event within a submodule, see Table 41. If this alarm is conveyed without an Alarm Item, all previous diagnosis is gone.
Multicast Communication Mismatch		No special use
Port Data Change Notification		No special use
Sync Data Change Notification		No special use
Isochronous Mode Problem Notification		No special use

Table 26: Special usages of alarms by the AS-i integration

8.6.7 Channel Diagnosis

Table 27 defines Channel Diagnoses and error types to be implemented according the column "Requirement" for slave proxy modules. For information with under which conditions Channel diagnosis alarms have to be initiated, see chapter 8.8.3.

Error type	Error de-scription	Sub slot	Channel Num-ber	Reason	Re-quirement
0x0009	Error	1	8000	Only if slave profile S-D.1 Appearing: I/O bit D2: 1 → 0 Disappearing: I/O bit D2: 0 → 1	**May**
0x0018	Output is switched off	1	8000	Only if slave profile S-D.1 Appearing: I/O bit D3: 1 → 0 Disappearing: I/O bit D3: 0 → 1	**May**
0x001A	External error	1, 2	8000	Appearing: AS-i master sets in LPF the bit of a slave with AS-i address α (α+32 if B-slave). Disappearing: AS-i-Master resets in LPF the bit of a slave with AS-i address α (α+32 if B-slave)	**Shall**
0x0200	Data trans-mission error	1, 2	0-3, 8000	Data transmission failed	**Shall**

Table 27: Supported Error types with Channel Diagnosis at AS-i slave proxy modules

8.6.8 Parameterisation Record Data

8.6.8.1 PCD And Activation Parameters (Index 160/ 0xA0)

PN/AS-i linking devices **shall** implement the start-up data record with index 0x00A0 defined in Figure 30 for all AS-i slave proxies. This data record is also offered in the GSDML for configuration and parameterisation at the submodule in subslot 1 of any slave module.

<table>
<tr><th>Byte №</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>0</td><td colspan="8">Major Version = 0x00</td></tr>
<tr><td>1</td><td colspan="8">Minor Version = 0x00</td></tr>
<tr><td>2</td><td colspan="4">I/O code</td><td colspan="4">ID code</td></tr>
<tr><td>3</td><td colspan="4">ID1 code</td><td colspan="4">ID2 code</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="4">Activation parameters</td></tr>
<tr><td>P3</td><td>P2</td><td>P1</td><td>P0</td></tr>
<tr><td>5</td><td colspan="8" rowspan="3">Reserved = 0x000000</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
</table>

Figure 30: Structure of the start up data record "PCD and activation parameters"

Parameters:
I/O code, ID code, ID1 code and ID2:
Values **shall** be set according to chapter 8.6.5, default for all parameters: 0xF
Activation parameters P0, P1, P2, P3:
Values/ default values **shall** be set according to [2].

8.6.9 Execution Control Functions

PN/AS-i linking devices **may** implement the AS-i Execution control functions listed in Table 28 for AS-i slave proxies. The specified data record indices **shall** be taken neither for other purposes of use nor for similar implementations of these functions.

AS-i	PROFINET IO		Details see chapter
Execution control function	**Service**	**Index**	
Set_Permanent_Parameter	RecordDataWrite	0x0003	8.6.9.1
Get_Permanent_Parameter	RecordDataRead	0x0004	8.6.9.2
Write_Parameter	RecordDataWrite	0x0005	8.6.9.3
Read_Parameter	RecordDataRead	0x0006	8.6.9.4
Set_Permanent_Configuration	RecordDataWrite	0x0008	8.6.9.5
Get_Permanent_Configuration	RecordDataRead	0x0009	8.6.9.6
Read_Actual_Configuration	RecordDataRead	0x000B	8.6.9.7
Read_Parameter_String	RecordDataRead	0x001B	8.6.9.8
Write_Parameter_String	RecordDataWrite	0x001C	8.6.9.9
Read_Diagnostic_String	RecordDataRead	0x001D	8.6.9.10
Read_Identification_String	RecordDataRead	0x001E	8.6.9.11

Table 28: Execution control functions for AS-i slave proxy

8.6.9.1 Set_Permanent_Parameter (Index 3/ 0x0x3)

The function is used by the host interface in order to request that the execution control shall store the delivered parameter value for a specific address non-volatile into the permanent parameter (PP) array.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0003 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The request

- **shall** contain the selected slave specific parameter values in the RecordDataWrite.Data field according to the structure defined in Figure 31 and Table 29.
    - The slave specific parameter block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field, particularly not the AS-i parameter 'Addr'. This value is implicitly passed by addressing the PROFINET IO RecordDataWrite service to the appropriate slot.

The response

- **should** be sent after the slave specific parameters are saved in non-volatile memory.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	Reserved = 0x0				Param[8]			
3	Reserved = 0x00							

Figure 31: Structure of RecordDataWrite / RecordDataRead.Data: slave specific parameters

For slaves working in the extended addressing mode the Select Bit (normally coded in Bit 3) cannot be forced. The value is ignored. The determination of the slave type being A or B is made by selecting the appropriate slot.

[8] Description of the associated structure 'S_Param' (in deviation from [2] and [3]) see Table 29

Addressing mode	Bit 3	Bit 2	Bit 1	Bit 0
Standard	P3	P2	P1	P0
Extended	-	P2	P1	P0

Table 29: Definition of the AS-i parameter 'Param': structure S_Param

8.6.9.2 Get_Permanent_Parameter (Index 4/ 0x0x4)

The function is used by the host interface in order to request that the execution control shall return the parameter value for a specific address from the Permanent_Parameter (PP) array.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0004 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the selected slave specific parameter values in the RecordDataRead.Data field according to the structure defined in Figure 31 and Table 29.
  - The Select Bit **shall** be filled according to the configured use.
  - The slave specific parameter block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.6.9.3 Write_Parameter (Index 5/ 0x0x5)

The function is used by the host interface in order to request that the execution control shall store the delivered parameter value for a specific address into the Parameter_Image (PI) array and that the execution control shall write this value in the next free management phase to the addressed slave using a Master_Request (TT:= MT, MD:= Write_Parameter). The Write_Parameter responds with the parameter value echoed from the slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0005 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The request

- **shall** contain the selected slave specific parameter values in the RecordDataWrite.Data field according to the structure defined in Figure 31 and Table 29.
  - The Select Bit **shall** be filled according to the configured use.
  - The slave specific parameter block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **should** provide the AS-i result value 'RS_Param', the echoed slave parameter value, as parameter IODWriteResHeader.AdditionalValue1.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.6.9.4 Read_Parameter (Index 6/ 0x0x6)

The function is used by the host interface in order to request that the execution control shall return the parameter value for a specific address from the parameter image (PI) array.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0006 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the selected slave specific parameter values in the RecordDataRead.Data field according to the structure defined in Figure 31 and Table 29.
  - The Select Bit **shall** be filled according to the configured use.
  - The default values **shall** be each 0xF for IO-Code, ID-Code, Extended ID-Code 1 and Extended ID-Code 2.
  - The slave specific parameter block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.

- **should** provide the AS-i result value ‘S_Param’, the echoed slave parameter value, as parameter IODReadHeader.AdditionalValue1.
- **should** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

8.6.9.5 Set_Permanent_Configuration (Index 8/ 0x0x8)

The function is used by the host interface to request that the execution control shall store the delivered configuration value for a specific address into the permanent configuration data (PCD) array (non-volatile).

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x0008 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The request

- **shall** contain the Configuration Data of the specific slave in the RecordDataWrite.Data field according to the structure defined in Figure 32.
  - For ‘Max_Slaves’ being one, the CDI **shall** shrink to a simple Configuration Data element.
  - The CDI block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field, particularly not the AS-i parameter ‘Addr’. This value is implicitly passed by addressing the PROFINET IO RecordDataWrite service to the appropriate slot.

The response

- **should** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

Byte №	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Major Version = 0x00							
1	Minor Version = 0x00							
2	I/O-Code				ID-Code			
3	Extended ID-Code 1				Extended ID-Code 2			

Figure 32: Structure of RecordDataWrite / RecordDataRead.Data: Configuration Data

8.6.9.6 Get_Permanent_Configuration (Index 9/ 0x0x9)

The function is used by the host interface in order that the execution control shall return the configuration data for a specific address from the permanent configuration data (PCD) array.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0009 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the PCD of the selected slave in the RecordDataRead.Data field according to the structure defined in Figure 32.
  - For ‘Max_Slaves’ being one, the PCD **shall** shrink to a simple Configuration Data element.
  - The PCD block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

8.6.9.7 Read_Actual_Configuration (Index 11/ 0x0xB)

The function is used by the host interface in order that the execution control shall return the parameter value for a specific address from the configuration data image (CDI) array.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x0011 addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the CDI of the specific slave in the RecordDataRead.Data field according to the structure defined in Figure 32.

  - o For 'Max_Slaves' being one, the CDI **shall** shrink to a simple Configuration Data element.
  - o The default values **shall** be each 0xF for IO-Code, ID-Code, Extended ID-Code 1 and Extended ID-Code 2.
  - o The CDI block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.6.9.8 Read_Parameter_String (Index 27/ 0x0x1B)

This function is used by the host interface to read a parameter string if provided by the slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x001B addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the parameter string of the specific slave in the RecordDataRead.Data field according to the structure defined in Figure 33.
  - o The service **should** provide up to 220 string bytes, signalising the actual number in byte № 2.
  - o In deviation from [2] and [3] the PROFINET service RecordDataRead **shall** return only the net bytes omitting the 3rd byte of each triple.
  - o The parameter string block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

Byte №	
0	Major Version = 0x00
1	Minor Version = 0x00
2	Number of string bytes
3	String byte 1
4	String byte 2
221	String byte 219 (where required)
222	String byte 220 (where required)
223	Reserved = 0x00

Figure 33: Structure of RecordDataWrite / RecordDataRead.Data: parameter string

8.6.9.9 Write_Parameter_String (Index 28/ 0x0x1C)

This function is used by the host interface to write a parameter string to a slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x001C addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The request

- **shall** contain the parameter string of the specific slave in the RecordDataWrite.Data field according to the structure defined in Figure 33.
  - o The service **should** provide up to 220 string bytes, the actual limit is given in byte № 2.
  - o In deviation from [2] and [3] the PROFINET in the service RecordDataWrite only the net string bytes **shall** be transferred. The linking device **shall** create the missing 3rd byte for supplementation of the AS-i transfer triples.
- **should** contain no further arguments in the RecordDataWrite.Data field, particularly not the AS-i parameter 'Addr'. This value is implicitly passed by addressing the PROFINET IO RecordDataWrite service to the appropriate slot.

The response

- **should** provide the AS-i result argument 'Status' as defined in Clause 2/ Table 11.

8.6.9.10 Read_Diagnostic_String (Index 29/ 0x0x1D)

This function is used by the host interface to read a diagnostic string if provided by the slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x001D addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the diagnostic string of the specific slave in the RecordDataRead.Data field according to the structure defined in Figure 33.
  - The service **should** provide up to 220 string bytes, signalising the actual number in byte № 2.
  - In deviation from [2] and [3] the PROFINET service RecordDataRead **shall** return only the net bytes omitting the 3rd byte of each triple.
  - The diagnostic string block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

8.6.9.11 Read_Identification_String (Index 30/ 0x0x1E)

This function is used by the host interface to read an identification string if provided by the slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x001E addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** contain the identification string of the specific slave in the RecordDataRead.Data field according to the structure defined in Figure 33.
  - The service **should** provide up to 220 string bytes, signalising the actual number in byte № 2.
  - In deviation from [2] and [3] the PROFINET service RecordDataRead **shall** return only the net bytes omitting the 3rd byte of each triple.
  - The identification string block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument ‘Status’ as defined in Clause 2/ Table 11.

8.6.10 CTT 2 Strings

8.6.10.1 Write_CTT_2_String (Index 31/ 0x1F)

This function is used by the host interface to write acyclic data for a CTT 2 slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataWrite/ index 0x001F addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The request

- **shall** contain the CTT2 specific output data in the RecordDataWrite.Data field according to the structure defined in Figure 33.
  - The data structure (command/ response coding, CTT 2 data) that **shall** be used is described in [3], chapter 4.4.5.4.
  - Following command and response primitives **should** be supported:
    - *acyclic standard write service request*
    - *acyclic Vendor specific write service request*
    - *acyclic standard read service response not ok*
    - *acyclic Vendor specific read service response*
    - *acyclic Vendor specific read service response not ok*
  - Each primitive **should** provide up to 220 string bytes, signalising the actual number in byte № 2.
  - The parameter string data block **shall** have top position in the RecordDataWrite.Data field.
- **should** contain no further arguments in the RecordDataWrite.Data field.

The response

- **shall** not be returned until the respective service response is returned by the AS-i slave.
- **shall** provide the AS-i result argument ‘Standard Error Code’ as defined in Table 30.

PROFINET response	Parameter 'Standard Error Code' and 'Diagnostic Code' in CTT 2 string responses					
	No error	Illegal index	Illegal length	Request not implemented	Busy	General error
ErrorCode	0x00	0xDF	0xDF	0xDF	0xDF	0xDF
ErrorDecode	0x00	0x80	0x80	0x80	0x80	0x80
ErrorCode1	0x00	0xA1	0xA1	0xA1	0xA1	0xA1
ErrorCode2	0x00	0x01	0x02	0x03	0x04	0xFF

Table 30: Mapping of AS-i result "Error Code" in CTT 2 string to "PNIOStatus" in RecordDataWrite

8.6.10.2 Read_CTT_2_String (Index 31/ 0x1F)

This function is used by the host interface to read acyclic data for a CTT 2 slave.

This AS-i function **should** be mapped to a PROFINET IO service RecordDataRead/ index 0x001F addressed to subslot 1 of the AS-i slave proxy module with the appropriate AS-i address.

The response

- **shall** not be returned until the respective service response is returned by the AS-i slave.
- **shall** contain the CTT2 specific output data in the RecordDataRead.Data field according to the structure defined in Figure 33.
  - o As defined in [2] and [3] the PROFINET service RecordDataRead **shall** return the data structure (command/ response coding, CTT 2 data) as described in [3], chapter 4.4.5.4.
  - o Following command and response primitives **should** be supported:
    - *acyclic standard read service request*
    - *acyclic Vendor specific read service request*
    - *acyclic standard write service response*
    - *acyclic standard write service response not ok*
  - o Each primitive **should** provide up to 220 string bytes, signalising the actual number in byte № 2.
  - o The identification string block **shall** have top position in the RecordDataRead.Data field.
- **should** contain no further arguments in the RecordDataRead.Data field.
- **should** provide the AS-i result argument 'Standard Error Code' as defined in Table 30.

8.7 Device Description

8.7.1 Channel Diagnoses

All Channel Diagnoses Error Types to be supported can be found in Table 7 and Table 27. In addition suitable texts have to be deposited in the GSDML for error description and info text. These texts may verify vendor specific as against the contents **shall** be identical.

8.7.2 Parameterisation Data Records

Parameterisation data records are defined in detail by Figure 10, Table 8 and Figure 30. One distinctive feature is to be defined yet: The labelling of the parameters IO code, ID code, ID1 code and ID2 code **should** result from Table 31. In support of the user the I/O sizes or even the range of modules **may** be offered in a help text.

Text components of ID code, ID1 code, ID2 code															
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Table 31: Labelling of parameters I/O code, ID code, ID1 code and ID in record data with index 160/ 0xA0

8.7.3 GSDML Properties

8.7.3.1 Device Access Point

Single and multi-masters **may** be selected by choosing different DAPs that are defined in the same GSDML. The according DAPs differ in the number of the fixed AS-i master modules, and the following sets of usable slots for AS-i slave modules.

The same technique of selecting different DAPs **may** be applied for selecting the linking device in order to use it as a PROFINET V1 or a V2 device. The details have already been worked out in chapter 8.5.

8.7.3.2 Physical Slots

A definition of a slot list together with the attribute “PysicalSlots” causes undefined slots not to be shown in the engineering surface. So in a GSDML description **should** be provided with an appropriate slot list according to Table 2 and Table 3. That is for example {0..31, 33..63} for a single master or {0..31, 33..63, 100..131, 133..163} for a dual master. This results in a clearly arranged sight in the engineering where only the usable slots will be seen.

8.7.3.3 Usable Modules

The AS-i master module(s) **shall** be described in the GSDML with the attribute ’FixedInSlots’ and the list of the appropriate line slots (see Table 3 and Table 4).

In the GSDML **shall** contain the modules listed in Table 32 with the attribute ‘AllowedInSlots’ having the values of all slave proxy slots defined by Table 2. All submodules in Table 32 **should** be defined in the GSDML as a VirtualSubmoduleList and so fixed in the according slots.

As a matter of principle there are some few dependencies between modules concerning the allocation of the slots which cannot be proofed by the engineering system. These interconnections are to be observed by the user are (whereas AS-i line $\lambda \in [0..15]$, AS-i address $\alpha \in [1..31]$) and are limited to:

- Slots 100 *λ +α are only allowed for AS-i standard slaves and slaves in extended addressing mode.
- 100 *λ +α +32 are only allowed if the AS-i slave α is used in extended addressing mode and there is no module in slot 100 *λ +α that represents a slave in standard addressing mode.
- CTT5 slave modules have to be used in succession as defined in Table 25.

Module types	ModuleID	SubmoduleID /IDs
1 byte input binary	0x0201 0001	0x0200 0100
1 byte output binary	0x0202 0001	0x0200 0001
1 byte in/out binary	0x0203 0001	0x0200 0101
1 word input analogue	0x0204 0001	0x0300 0200
1 word output analogue	0x0205 0001	0x0300 0002
1 word in/out analogue	0x0206 0001	0x0300 0202
2 word input analogue	0x0207 0001	0x0300 0400
2 word output analogue	0x0208 0001	0x0300 0004
2 word in/out analogue	0x0209 0001	0x0300 0404
4 words input analogue	0x020A 0001	0x0300 0800
4 words output analogue	0x020B 0001	0x0300 0008
4 words in/out analogue	0x020C 0001	0x0300 0808
1 Byte output binary, 1 word input analogue	0x0221 0002	0x0200 0001, 0x0300 0200
1 Byte in/out binary, 2 words in/out analogue	0x0222 0002	0x0200 0101, 0x0300 0404
1 Byte in/out binary, 4 words in/out analogue	0x0223 0002	0x0200 0101, 0x0300 0808
0 byte module for CTT5 slaves, 2^{nd} – 4^{th} address	0x0240 0001	0x0100 0000

Table 32: Usable modules for AS-i slaves in the GSDML

8.7.4 Wild-Card Module

There is no special wild-card module defined. Wild-card modules are modules with the default values of "PCD And Activation Parameters (Index 160/ 0xA0)".

As default module (attribute "UsedInSlots") **should** be used the module "1 byte in/out binary" of Table 32.

8.8 Dynamic Behaviour

8.8.1 AS-i Master Operation Modes

The AS-i masters of a PN/AS-i linking device provide the operation states, 'configuration mode' and 'protected mode'. Dependent on these modes special PROFINET IO Application Service Elements perform differently[9] as stated in Table 33. In addition, some actions force an implicit mode change.

Mode	ASE	Behaviour
Configuration mode	AR	The ApplicationReady service **shall** report all module and submodule states respectively as being okay (see Table 37 and Table 38).
	Alarm	AS-i configuration alarms **shall** be disabled (Table 41). Return of Submodule shall be initiated according to Table 42.
	Record Data	All valid and permitted Record Data services **should** be accepted.
	I/O Data & States	See Table 13, Table 14 and Table 15.
Change from **configuration mode** to **protected mode**	AR	All connects of CRs **shall** be delayed or rejected after timeout.
	Alarm	When changing to protected mode all disabled alarms **shall** be initiated ex post where required (see Table 41). Return of Submodule shall be initiated according to Table 42.
	Record Data	All Record Data services **shall** be delayed or rejected after timeout.
	I/O Data & States	See Table 13, Table 14 and Table 15.
Protected mode	AR	The ApplicationReady **shall** report locked submodules, missing and faulty slaves SubmoduleStates with error condition (see Table 39).
	Alarm	All alarms described in Table 6 and Table 26 and used for mapping purposes **shall** be initiated when the suitable condition occurs (see Table 41).
	Record Data	The write service of following DataRecords **shall** be blocked: Set_Permanent_Configuration (Index 8/ 0x0x8) Store_Actual_Configuration (Index 10/ 0xA) Set_LPS (Index 12/ 0xC) All other valid and permitted DataRecord services **should** be accepted.
		The write service **shall** force a change to Configuration Mode if the service's configuration data differs from the current configuration: PCD And Activation Parameters (Index 160/ 0xA0).
	I/O Data & States	See Table 13, Table 14 and Table 15.
Change from **protected mode** to **configuration mode**	AR	All connects of CRs **shall** be delayed or rejected after timeout.
	Alarm	When changing to configuration mode all pending configuration alarms have to be withdrawn (see Table 41).
	Record Data	All Record Data services **shall** be blocked.
	I/O Data & States	See Table 13, Table 14 and Table 15.

Table 33: Interconnection between special ASE and AS-i operation modes

8.8.2 Expected Identification and Real Identification

There are two basic ways to obtain the Real Identification for a PN/ AS-i linking device.

[9] In addition to the itemized restrictions there are additional constraints specified by [2] and [3].

1. A User configures explicitly the AS-i line and its devices with an Engineering system followed by loading this information to the PROFINET IO-Controller.
2. The configuration is learned in a different way and already present at the linking device.

When a PROFINET IO-Controller takes control over submodules establishing the IO-CR, there is to be decided on the linking device whether the Expected Identification fits to the Real Identification. Particularly following parameters **shall** be checked:

- Module Ident Number
- Submodule Ident Number
- Availability of the module
- AS-i Configuration Data (i.e. I/O_Code, ID_Code, ID1_Code and ID2_Code are valid values for the slave)

The definition, which Module/ Submodule Ident Number of an AS-i slave proxy module **shall** be accepted for a real AS-i slave, is given in chapters 8.6.4 and 8.6.5.

A module of AS-i slave proxy is available

- for an IO Controller if none of its submodules is controlled by IO Supervisor (via a Supervisor AR) and not controlled by another IO Controller.
- for an IO Supervisor if no controller has taken any of its submodules unless the attribute ‘Takeover Allowed’ of the taken submodule is set to ‘ALLOWED’.

Table 34 describes how to acquire the expected CD configuration of an AS-i slave.

Condition before AR connect		**Expected CD after EndOfParameter.ind**	
CD[10] in DR (PCD And Activation Parameters (Index 160/ 0xA0**)**	**Slave bit in LPS**	**Slave bit in LPS**	**PCD**
= 0xFFFF	0	0	PCD out of remanent memory of PN/AS-i linking device
	1	1	
	0	0	
	1	1	
≠ 0xFFFF	0	1	CD of DR “PCD And Activation Parameters (Index 160/ 0xA0)”
	1		
Module not configured	0	0	None
	1		

Table 34: Rules to determine the expected AS-i Configuration Data accordant to parameterisation

8.8.2.1 Rules For AS-i Line Proxy Modules

The ModuleState in the ModuleDiffBlock **should** be set according to Table 35.

Preconditions	**ModuleState**
AS-i line supported by linking device	
False	no module
True	proper module

Table 35: Parameterisation of AS-i line modules - ModuleDiffBlock.ModuleState

The AS-i line modules **shall** not be taken over by an IO Supervisor. Therefore, the Parameter SubmoduleState.AddInfo for these modules **should** be set in the ModuleDiffBlock according to Table 36 (FormatIndicator required to be ‘1’).

[10] CD = 0xFFFF goes as short form of I/O_Code = 0xF, ID_Code = 0xF, ID1_Code = 0xF, ID2_Code = 0xF

Rule №	Preconditions	SubmoduleState				
		Format Indicator	AddInfo	DiagInfo	ARInfo	IdentInfo
1	<None>	1	Takeover is not allowed	No Diagnosis Data available	Own	OK

Table 36: Parameterisation of AS-i line modules - ModuleDiffBlock.SubmoduleState

8.8.2.2 Rules For Slave Proxy Modules

The comparison of the Expected Identification and Real Identification data is not possible for all module types at AR connect time. The missing description of the PCD is transferred afterwards as record data (see chapter 8.6.8.1). So, always a ModuleDiffBlock **should** be returned with the ApplicationReady.ind.

Table 37 defines, specific for any AS-i master proxy module, how the correct Module state for the ModuleDiffBlock (and other context class services) **should** be determined.

Preconditions		ModuleState
Comparison of CD[11] and ModuleID/ SubmoduleIDs	Bit in LDS	
-	0	no module
ModuleID and SubmoduleIDs defined for IO Code and ID Code of PCD can either not be found in Table 19, Table 21, Table 22, Table 23, Table 24 or Table 25 or cannot be substituted according to Table 16, Table 17 or Table 18	1	wrong module
IO Code and ID Code of PCD fit exactly to ModuleID and SubmoduleIDs defined in Table 19, Table 21, Table 22, Table 23, Table 24 or Table 25.	1	proper module
ModuleID and SubmoduleIDs defined for IO Code and ID Code of PCD in Table 19, Table 21, Table 22, Table 23, Table 24 or Table 25 may be substituted according to Table 16, Table 17 or Table 18	1	substitute

Table 37: Parametrisation of slave proxy modules – ModuleDiffBlock.ModuleState

Table 38 and Table 39 define which SubmoduleState **should** be returned in the ModuleDiffBlock (and other context class services). The standard handling **should** be supported (defined by rules 2 to 5, 9 and 10), the supervisor handling **may** be supported and in this case the rules 7 to 8, 11 12 **should** be used.

Rule №	Preconditions		SubmoduleState				
	Bit in LDS	Module Constraint	Format Indicator	AddInfo	DiagInfo	ARInfo	IdentInfo
2	0	None	*<none>*	*<none>*	*<none>*	*<none>*	*<none>*
3	1	Correct CD of the slave[11] and module is available[12] and is correct[13]	1	*<do not care>*	No Diagnosis Data available	Own	OK

[11] The assumed CD **shall** be taken out of the slave proxy module's data record "PCD And Activation Parameters (Index 160/ 0xA0)".

[12] IO-AR: A module is available, if all of its submodules are available. A module is locked if at least one submodule is locked by IO Supervisor or another IO controller.

[13] A module is correct iff

- The Module Ident Number and the fixed Submodule Ident Numbers are an allowed combination for the slave, and
- The CD (I/O_Code, ID_Code, ID1_Code and ID2_Code) have valid values for the slave.

Rule №	Preconditions		SubmoduleState				
	Bit in LDS	Module Constraint	Format Indicator	AddInfo	DiagInfo	ARInfo	IdentInfo
4	1	Wrong CD of the slave[11] and module is available[12] and correct[13]	1	*<do not care>*	No Diagnosis Data available	Application Ready Pending	Wrong
5	1	Correct CD of the slave[11] and substitute module is available[12] but no corresponding submodule	1	*<do not care>*	No Diagnosis Data available	Application Ready Pending (ARP)	NoSubmodule
6	1	Correct CD of the slave[11] and substitute module is available[12] and submodule was subject to over-control or unter-control	1	*<do not care>*	No Diagnosis Data available	Application Ready Pending (ARP)	Substitute
7	1	Module is locked by Supervisor	1	None	No Diagnosis Data available	Locked By IO Supervisor (IOS)	No
8	1	Module is locked by Controller[12]	1	takeover is not allowed	No Diagnosis Data available	Lockes By IO Controller (IOC)	No

Table 38: Parameterisation of slave proxy modules - ModuleDiffBlock.SubmoduleState in protected mode

Rule №	Preconditions		SubmoduleState				
	Bit in LDS	Module Constraint	Format Indicator	AddInfo	DiagInfo	ARInfo	IdentInfo
9	0	None	1	*<do not care>*	No Diagnosis Data available	Own	OK
10	1	Module is availlable[12]	1	*<do not care>*	No Diagnosis Data available	Own	OK
11	1	Module is locked[12]	1	None	No Diagnosis Data available	Own	OK
12	1	Module is locked by Controller[12]	1	Takeover is not allowed	No Diagnosis Data available	Own	OK

Table 39: Parameterisation of slave proxy modules - ModuleDiffBlock.SubmoduleState in configuration mode

8.8.3 Alarms Rules

Configuration alarms split into several alarm types. At which point at time a special alarm **shall** be initiated in can be learned from Table 41. For this purpose Table 40 defines alarm states of an AS-i slave proxy module according to the stated preconditions. The initial alarm state **shall** be “No alarm”.

Preconditions					**Alarm state**
Module in expected configuration	**Module takeover by Supervisor**	**Slave bit in LPS**	**Slave bit in LDS**	**Comparison expected/ actual configuration**	
-	False	0	0	-	No alarm
-	False	0	1	-	Diagnosis pending
False	False	1	-	-	
True	False	1	0	-	Pulled
True	False	1	1	PCD ≠ CDI	Plugged wrong
True	False	1	1	PCD = CDI	No alarm
-	True	0	1	-	Diagnosis pending & Supervisor
False	True	1	-	-	
True	True	1	1	-	Supervisor locked

Table 40: Definition of alarm states of an AS-i slave proxy

Alarm state	**Alarms to be initiated if condition is fulfilled**		
	Entering protected mode	**Being in protected mode**	**Entering configuration mode**
No alarm	-	-	-
Diagnosis pending	Status, Channel Diagnosis[14]	Channel Diagnosis[14]	Channel Diagnosis disappears[14], Status
Pulled	Status, Pull	Pull	Plug, Status
Plugged wrong	Status, Plug Wrong Submodule	Plug Wrong Submodule	Pull, Status
Supervisor locked	Status, Controlled by Supervisor	Controlled by Supervisor	Released, Status
Diagnosis pending & Supervisor	Status, Channel Diagnosis disappears[14], Controlled by Supervisor	Channel Diagnosis disappears[14], Controlled by Supervisor	Released, Channel Diagnosis[14], Status

Table 41: Dependency of alarms types on master mode changes and a slave proxy’s alarm state

Independent of the AS-i proxy’s alarm states and of the AS-i modes any change of an input IOXS from ‘bad’ to ‘good’ **shall** go along with an alarm ‘Return of submodule’:

Change of input IOPS	**Alarm on entering configuration mode**
Bad → good	Return of Submodule
Good → bad, no change	-

Table 42: Dependency of alarms types on master mode changes and a slave proxy’s input IOPS

[14] Channel Diagnoses that are initiated in this alarm state may stick also at the AS-i master modules, see both Table 7 and Table 27

9 Requirements to certification tests

The requirements of the PROFINET IO certification are covered by the certification test for PROFINET IO-devices. There are no additional requirements to be considered. The certification of the fieldbus part is covered by the established certification procedure of AS-International Assotiation.

PROFIBUS Nutzerorganisation e.V.
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 96 58 590
Fax: +49 (0) 721 / 96 58 589
e-mail: info@profibus.com
http://www.profibus.com