

CONSOLIDATED VERSION

VERSION CONSOLIDÉE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2016 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

CONSOLIDATED VERSION

VERSION CONSOLIDÉE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.05; 35.110

ISBN 978-2-8322-3498-3

Warning! Make sure that you obtained this publication from an authorized distributor. Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.
--

REDLINE VERSION

VERSION REDLINE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope	8
2 Normative references	8
3 Terms, definitions, symbols, abbreviated terms and conventions	9
3.1 Terms and definitions	9
3.2 Symbols and abbreviated terms	9
3.3 Conventions	10
3.3.1 Data type names and references to data types	10
3.3.2 Vocabulary for requirements.....	10
3.3.3 Use of UML.....	10
4 Bus category	10
5 Access to instance and device data	11
5.1 Process Channel objects provided by DTM.....	11
5.2 DTM services to access instance and device data	11
6 Protocol specific behavior	11
6.1 PROFIBUS device model.....	11
6.2 Configuration and parameterization of PROFIBUS devices	12
6.2.1 General	12
6.2.2 Monolithic DTM for a modular PROFIBUS device.....	13
6.2.3 Modular DTM for a modular PROFIBUS device	13
6.3 Support for DPV0 configuration.....	14
6.4 PROFIBUS slaves operating without a cyclic PROFIBUS master	14
6.5 PROFIBUS-related information of a slave DTM	14
6.5.1 General	14
6.5.2 Bus Master Configuration Part (BMCP).....	15
7 Protocol specific usage of general data types.....	25
8 Protocol specific common data types	27
9 Network management data types	27
9.1 General.....	27
9.1.1 Configuration	27
9.1.2 Process Channel.....	28
9.1.3 Parameterization.....	28
9.2 Master-bus parameter set.....	29
9.3 Slave bus parameter set.....	29
9.4 Module and channel data.....	30
9.5 GSD information.....	33
9.5.1 General	33
9.5.2 GSD for gateway devices	33
10 Communication data types.....	34
10.1 General.....	34
10.2 Error information provided by Communication Channel.....	34
10.3 DPV0 communication.....	34
10.4 DPV1 communication.....	41
11 Channel parameter data types	44

12 Device identification	47
12.1 General	47
12.2 Protocol specific handling of the data type STRING	47
12.3 Common device type identification data types	47
12.4 Topology scan data types	52
12.5 Scan identification data types	52
12.6 Device type identification data types – provided by DTM	55
12.7 Identification information in GUI	58
13 ProfiSafe	58
13.1 Motivation	58
13.2 General parameter handling	58
13.3 ProfiSafe individual device parameter	59
Bibliography	61
 Figure 1 – Part 303-1 of the IEC 62453 series	7
Figure 2 – FDT PROFIBUS device model	12
Figure 3 – Example for IO data within datagrams	31
Figure 4 – F-Parameter and individual device parameter	59
Figure 5 – Data structure of ProfiSafe individual device parameters	60
 Table 1 – Protocol identifiers	10
Table 2 – Physical layer identifiers	10
Table 3 – BMPC BMCP Part1 – General configuration	16
Table 4 – BMPC BMCP Part2 – Parameter data	16
Table 5 – BMPC BMCP Part3 – Configuration data	17
Table 6 – Part 4: Address table and slave user parameters	18
Table 7 – Part 4: Extended Prm data	18
Table 8 – Complete BMCP	19
Table 9 – Protocol specific usage of general data types	25
Table 10 – Bus parameter set for master device	29
Table 11 – Bus parameter set for slave device	30
Table 12 – Signal channels within the data frame	32
Table 13 – Simple DPV0 communication data types	35
Table 14 – Structured DPV0 Communication data types	35
Table 15 – Availability of services for Master Class1 (C1)	40
Table 16 – Availability of services for Master Class2 (C2)	40
Table 17 – Simple DPV1 communication data types	41
Table 18 – Structured DPV1 communication data types	42
Table 19 – Mapping of DPV1 data types to FDT data types	44
Table 20 – Simple Channel Parameter data types	45
Table 21 – Structured Channel Parameter data types	46
Table 22 – Identification data types with Profibus DP specific mapping	48
Table 23 – Identification data types with Profibus I&M specific mapping	49
Table 24 – Identification data types with Profibus PA specific mapping	51

Table 25 – Simple identification data types with protocol independent semantics	52
Table 26 – Structured identification data types with protocol independent semantics	52
Table 27 – Simple topology scan data types	52
Table 28 – Structured topology scan data types.....	52
Table 29 – Simple scan identification data types.....	53
Table 30 – Structured scan identification data types	53
Table 31 – Structured device identification data types	56
Table 32 – DataLink Layer Identifiers	10

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

DISCLAIMER

This Consolidated version is not an official IEC Standard and has been prepared for user convenience. Only the current versions of the standard and its amendment(s) are to be considered the official documents.

This Consolidated version of IEC 62453-303-1 bears the edition number 1.1. It consists of the first edition (2009-06) [documents 65E/127/FDIS and 65E/140/RVD] and its amendment 1 (2016-06) [documents 65E/336/CDV and 65E/395A/RVC]. The technical content is identical to the base edition and its amendment.

In this Redline version, a vertical line in the margin shows where the technical content is modified by amendment 1. Additions are in green text, deletions are in strikethrough red text. A separate Final version with all changes accepted is available in this publication.

International Standard IEC 62453-303-1 been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

Each part of the IEC 62453-3xy series is intended to be read in conjunction with IEC 62453-2.

The French version of this standard has not been voted upon.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 62453 series, under the general title *Field Device Tool (FDT) interface specification*, can be found on the IEC website.

The committee has decided that the contents of the base publication and its amendment will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The “colour inside” logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.

INTRODUCTION

This part of IEC 62453 is an interface specification for developers of FDT (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

With the integration of fieldbusses into control systems, there are a few other tasks which need to be performed. In addition to fieldbus- and device-specific tools, there is a need to integrate these tools into higher-level system-wide planning- or engineering tools. In particular, for use in extensive and heterogeneous control systems, typically in the area of the process industry, the unambiguous definition of engineering interfaces that are easy to use for all those involved is of great importance.

A device-specific software component, called DTM (Device Type Manager), is supplied by the field device manufacturer with its device. The DTM is integrated into engineering tools via the FDT interfaces defined in this specification. The approach to integration is in general open for all kinds of fieldbusses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

Figure 1 shows how IEC 62453–303-1 is aligned in the structure of the IEC 62453 series.

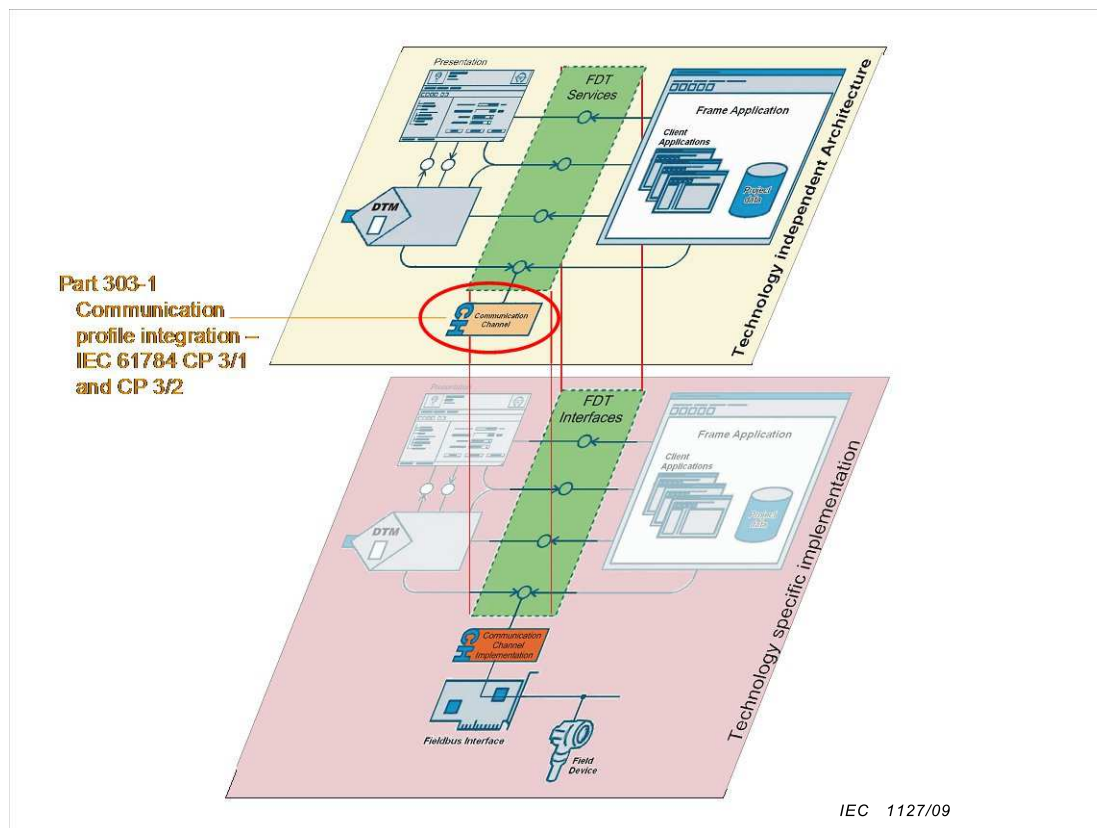


Figure 1 – Part 303-1 of the IEC 62453 series

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2

1 Scope

Communication Profile 3/1 and Communication Profile 3/2 (commonly known as PROFIBUS™¹) defines communication profiles based on IEC 61158-2 Type 3, IEC 61158-3-3, IEC 61158-4-3, IEC 61158-5-3, and IEC 61158-6-3. The basic profiles CP 3/1 (PROFIBUS DP) and CP 3/2 (PROFIBUS PA) are defined in IEC 61784-1.

This part of IEC 62453 provides information for integrating the PROFIBUS protocol into the FDT interface specification (IEC 62453–2).

This part of the IEC 62453 specifies communication and other services.

This specification neither contains the FDT specification nor modifies it.

2 Normative references

The following referenced documents are indispensable for the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*

IEC 61158-2:2014, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-3-3, *Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3, *Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-5-3, *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements*

IEC 61158-6-3, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

¹ PROFIBUS™ is a trade name of the non-profit organization PROFIBUS Nutzerorganisation e.V. (PNO). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for PROFIBUS™. Use of the registered logos for PROFIBUS™ requires permission of PNO.

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62453-1 and IEC 62453-2 apply.

3.1.1

bus interface module

module of a field device that provides the connection to the fieldbus

3.1.2

CP 3/1

Communication profile of CPF3, featuring asynchronous transmission; RS 485 (ANSI TIA/EIA RS-485-A); optional RS 485-IS; plastic fiber; glass multi mode fiber or glass single mode fiber; PCF fiber

3.1.3

CP 3/2

Communication profile of CPF3, featuring synchronous transmission; manchester coded and bus powered (MBP); optional intrinsically safe (MBP-IS) and lower power (MBP-LP)

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviations given in IEC 62453-1, IEC 62453-2 and the following apply.

ANSI	American National Standards Institute (http://www.ansi.org)
BIM	Bus Interface Module
BMCP	Bus Master Configuration Part
CFG	Configuration data used during initialization of PROFIBUS slave device
DCS	Distributed Control System
DP	Decentralized Peripherals
EIA	Electronic Industries Alliance
FDL	Fieldbus Data Link layer
FMA	Fieldbus Management layer
FMS	Fieldbus Message Specification
GSD	General Station Description
MBP	Manchester coded Bus Powered
I&M	Identification and maintenance functions
PA	Process Automation
PCF	Polymer Clad Fibre
PROFIBUS	Process Field Bus
RS	Radio Sector / Recommended Standard
TIA	Telecommunications Industry Association

3.3 Conventions

3.3.1 Data type names and references to data types

The conventions for naming and referencing of data types are explained in IEC 62453-2, Clause A.1

3.3.2 Vocabulary for requirements

The following expressions are used when specifying requirements.

Usage of “shall” or “mandatory”	No exceptions allowed.
Usage of “should” or “recommended”	Strong recommendation. It may make sense in special exceptional cases to differ from the described behaviour.
Usage of “conditional”	Function or behaviour shall be provided, depending on defined conditions.
Usage of “can” or “optional”	Function or behaviour may be provided, depending on defined conditions.

3.3.3 Use of UML

Figures in this document are using the UML notation as defined in Annex A of IEC 62453–1.

4 Bus category

CP 3/1 and CP 3/2 protocols are identified in the protocolId element of the structured data type 'fdt:BusCategory' by the following unique identifiers (Table 1):

Table 1 – Protocol identifiers

Identifier value	ProtocolId name	Description
036D1497-387B-11D4-86E1-00E0987270B9	'Profibus DP/V0'	Support of Profibus DP V0 protocol
036D1499-387B-11D4-86E1-00E0987270B9	'Profibus DP/V1'	Support of Profibus DP V1 protocol

CP 3/1 AND CP 3/2 protocols are using the following unique identifiers in physicalLayer members within PhysicalLayer data type (Table 2):

Table 2 – Physical layer identifiers

Identifier value	Name	Description
036D1590-387B-11D4-86E1-00E0987270B9	MBP	IEC 61158-2 (MBP, Profibus PA)
036D1591-387B-11D4-86E1-00E0987270B9	RS485	IEC 61158-2:2014, Clause 22 (RS485, PROFIBUS DP)
036D1592-387B-11D4-86E1-00E0987270B9	Fiber Optic	IEC 61158-2:2014, Clause 23 (Fiber Optic cable, PROFIBUS DP)
036D1593-387B-11D4-86E1-00E0987270B9		Ethernet (depreciated, do not use)

Table 32 defines which DataLinkLayer shall be used in combination with the BusCategory values defined in Table 32.

Table 32 – DataLink Layer Identifiers

Identifier value	Name	Description
50A21B35-7EE7-4999-8174-70396929C0B4	PROFIBUS FDL	PROFIBUS FDL
CDF338DC-E9A3-4D13-91AC-60A43DCB2904	PROFIBUS FMA1/2	PROFIBUS FMA1/2

5 Access to instance and device data

5.1 Process Channel objects provided by DTM

The minimum set of provided data should be: Process values modeled as channel objects including the ranges and scaling.

5.2 DTM services to access instance and device data

The services InstanceDataInformation and DeviceDataInformation shall provide access to at least all parameters of the Physical Block and the status and Out value of the Function Blocks shall be exposed.

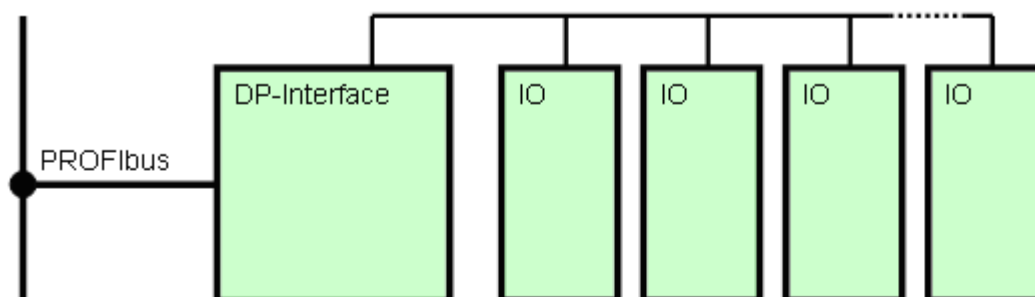
According to IEC 62453-2, at least one set of semantic information (one per supported fieldbus protocol) shall be provided for each accessible data object, using the 'SemanticInformation' general data type. The corresponding data type 'applicationDomain' shall have a value defined for Profibus and the data type 'semanticId' shall have an appropriate value, as specified in Table 9.

6 Protocol specific behavior

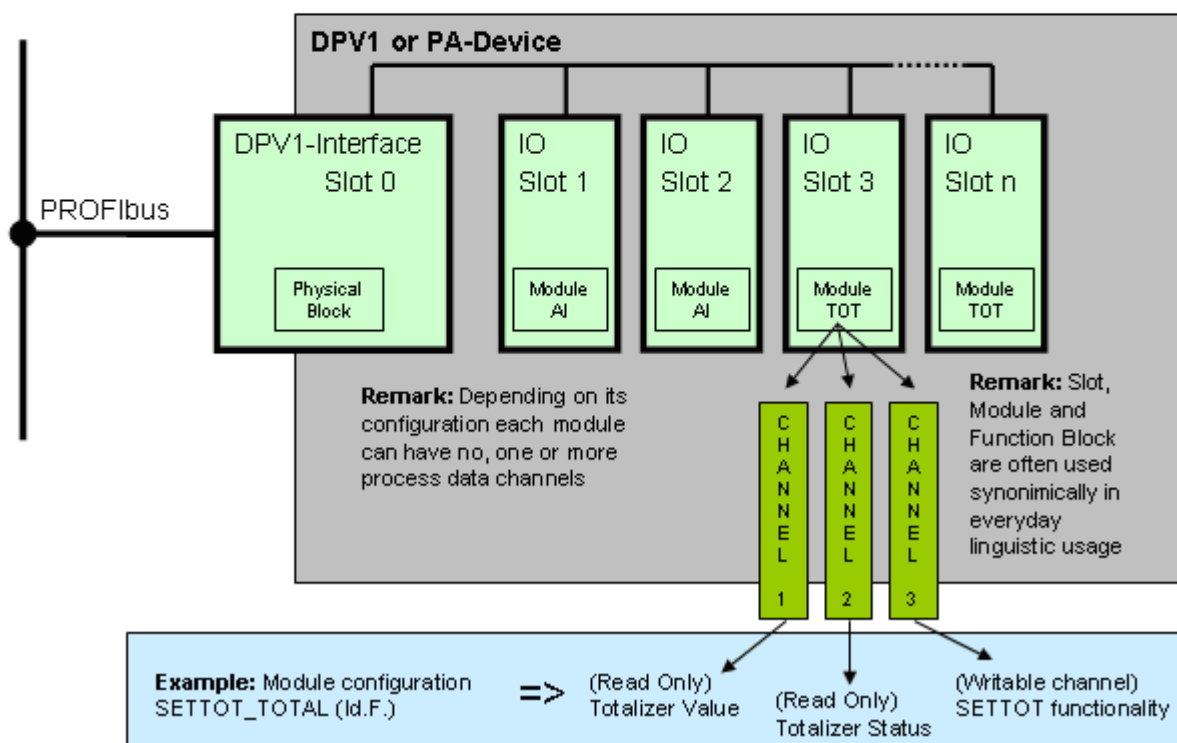
6.1 PROFIBUS device model

FDT extends the PROFIBUS device model by using Process Channels for description of I/O values (see Figure 2).

Classical View of PROFIBUS device



PROFIBUS notations from a monolithic DPV1 or PA device DTMs point of view



IEC 1128/09

Figure 2 – FDT PROFIBUS device model

6.2 Configuration and parameterization of PROFIBUS devices

6.2.1 General

In a GSD-based configuration tool the user defines the configuration and sets the appropriate parameters for the modules. The configuration tool creates the configuration string and the parameter string that are used to set up the slave properly.

With FDT the configuration and parameterization of the devices is no longer executed only by a central component; it moved partly into the DTMs. A DTM is responsible for providing configuration and parameterization information for the cyclic master that puts the PROFIBUS slaves in operation.

A DTM is used to adjust a field device to its specific application. Within PROFIBUS, there are three different aspects of adjustment:

- parameterization: usr prm data (used in the PROFIBUS service SET_PRM for setting up the cyclic communication and the specific behavior of the device);

- application parameterization: application specific parameters (transmitted via acyclic read/write PROFIBUS services);
- configuration: configuration data (used in the PROFIBUS service CHK_CFG for definition of the format and length of the input/output data that are transmitted within cyclic communication).

The application parameterization transmitted via acyclic communication is not in the scope of this document. Within this document the term parameterization represents communication parameterization (SET_PRM).

6.2.2 Monolithic DTM for a modular PROFIBUS device

A monolithic DTM is one single DTM that represents the complete device with its Bus Interface Module (BIM) and its I/O modules. In general, such a DTM offers a configuration dialog (presentation object) that allows definition of the used BIM and modules. The configuration dialog must be available via the FDT standard function “Configure” (see [1] 4.3 Operation Configuration).

Not all PROFIBUS devices require a configuration dialog. That is why not all DTMs provide the “Configure” function. This is valid only for non-modular PROFIBUS devices if the Usr_Prm-Data cannot be changed.

The configuration dialog allows changing the data only in offline mode if the data set can be locked.

6.2.3 Modular DTM for a modular PROFIBUS device

Separate DTMs represent the BIM (Device DTM) and the particular I/O modules (Module DTMs). The effort developing such a modular DTM is normally higher than in the case of a monolithic DTM, because:

- a private protocol has to be implemented between BIM and I/O modules to ensure that only a Module DTM can be added to the BIM DTM. This requires an own protocol ID and the adaptation / creation of communication;
- in some cases, additional private interfaces are necessary to exchange information between Device DTM for BIM and Module DTMs.

Implementing a Modular DTM results in the following advantages:

- the project represents the device structure;
- the user is able to access module-related information directly as a function of the Module DTM;
- the FDT specification defines a mechanism to identify DTMs. With these mechanisms it is possible to provide support for scanning the modules below the BIM and generate the topology automatically;
- supporting a new type of BIM or I/O module requires an additional DTM “only” and does not affect existing components. This may result in reduced test effort.

The configuration data to set up the PROFIBUS configuration must be provided by the Device DTM (representing the BIM). This configuration data may be generated from information of the instantiated Child DTMs and by using a configuration dialog.

Modular DTMs should be provided for modular devices (e.g. a plant operator may add/remove modules). Monolithic DTMs are used to represent devices that show no modularity (e.g. PA devices).

6.3 Support for DPV0 configuration

A PROFIBUS slave is configured by a cyclic master and communicates via PROFIBUS DP. In addition to this the slave may support DPV1 communication.

A Gateway DTM for a PROFIBUS slave does not have to provide communication for the DPV0 communication schema. For example, there is a remote I/O system with HART modules. It may have a Gateway DTM that requires the DPV1 protocol and provides the HART protocol (defined in the information document and in the parameter document). This enables HART Device DTMs to communicate with their devices via the Gateway DTM and via Communication DTM for DPV1. Following the specification the Gateway DTM delivers channel parameter documents for both protocols DPV1 and HART. The protocolId is a member of NetworkManagementInfo data type.

The Process Channels ~~must~~ shall provide Channel Parameter documents for DPV1 including all information to allow integration into the control system (e.g. DPAddress of the IO value if available).

6.4 PROFIBUS slaves operating without a cyclic PROFIBUS master

In most cases, a PROFIBUS slave is configured and parameterized by a cyclic PROFIBUS master device. So a running master device in the network is required.

Some slaves are able to allow acyclic communication without a running cyclic master. Especially in the case of gateway functionality this is an eminent advantage because they allow the parameterization of field devices connected to them by using an acyclic bus master. So instrument specialists are able to work with field devices also in case the controller is not yet working.

If a master starts communication, these devices start to detect bus speed and settings to react properly. This may take some time.

In the following, two cases are described that a user may keep in mind when working with such devices.

Use case 1:

The user performs a network scan. The Communication DTM tries to read diagnostic data via a GetDiagnose Request but does not receive a response. The device is not detected by the Communication DTM. This occurs mostly when the device has a low PROFIBUS address. The reason is that the device has not completed bus speed / bus setting detection as it was asked for their diagnostic data. The workaround is to give these devices a higher PROFIBUS address.

Use Cases 2:

The user tries to connect a field device linked to the gateway that supports DPV1 without a running cyclic master. This can lead to an error message because the gateway device has not completed bus speed / bus setting detection as it was asked for a connection. So the user has to try to connect again. This happens only in very rare situations.

6.5 PROFIBUS-related information of a slave DTM

6.5.1 General

The information used by a cyclic master device to set up the PROFIBUS network properly and allow cyclic communication between control system and slave devices is provided by a DTM in

- Bus Master Configuration Part (BMCP);
- GSD information;
- internal topology,
- process channels.

A DTM of a PROFIBUS slave must deliver these parts of PROFIBUS-related information to get integrated into a FDT-based Engineering System. In the next subclauses, a more detailed description is given on how to generate and how to provide this information. This depends on the kind of DTM (see 6.2 Configuration and parameterization of PROFIBUS devices).

6.5.2 Bus Master Configuration Part (BMCP)

6.5.2.1 BMCP introduction

The BMCP of one single DTM instance describes the actual parameter and configuration data of the corresponding PROFIBUS slave. Each DTM representing a PROFIBUS slave device must provide a Bus Master Configuration Part. The BMCP is provided in the `busMasterConfigurationPart` member of `NetworkManagementInfo` data type. This information is obtained by calling service `NetworkManagementInfoRead`.

The BMCP includes information about the configuration and the parameters for the slave. The BMCP is provided by the DTM and is required in order to generate the master configuration.

The BMCP contains data which might be changed during master configuration. That means that the BMCP may be changed and transferred back to the slave DTM by calling `NetworkManagementInfoWrite`. A Slave DTM must accept the new information and recompute the configuration / internal parameters to match the new BMCP.

DTM must check whether the new values are according to the capabilities of the device. The `NetworkManagementInfoWrite` call will be refused if the device can not handle the new values.

6.5.2.2 Creating the BMCP

This subclause explains the meaning of the individual elements of the BMCP in details.

The BCMP may be generated from the GSD information of a PROFIBUS device.

The BMCP is divided into four parts that are explained in the following subclauses.

The explanations use the GSD keywords and reference the PROFIBUS specification. See also the table with complete BMCP in 6.5.2.2.1 .

6.5.2.2.1 Part 1: From Slave_Para_Len to Octet 15

The first part consists of a fixed set of bytes described in the table below.

Table 3 – ~~BMPC~~ BMCP Part1 – General configuration

Byte	Name	Defined in	Notice
0	Slave_Para_Len	[5] 6.2.12.1	Length of the BMCP including this value
1			
2	SI_Flag	[5] 6.2.12.2	The following GSD values are used [6]: • Bit 0 : reserved, • Bit 1: Extra_Alarm_SAP, • Bit 2: DPV1_Data_Types, • Bit 3: DPV1_Supported, • Bit 4: Publisher_Enable, • Bit 5 : Fail_Safe, . The following bits are not based on GSD values: • Bit 6 : New_Prm, • Bit 7 : Active
3	Slave Type	[5] 6.2.12.3	Value is 0 (= DP-Slave)
4	Max_Diag_Data_Len	[5] 6.2.12.4	The following GSD value is used [6]: Max_Diag_Data_Len
5	Max_Alarm_Len	[5] 6.2.12.5	
6	Max_Channel_Data_Len	[5] 6.2.12.6	This field defines how much data can be transferred between slave and master. In this case, the maximum of these GSD values [6] must be calculated: • Max_Data_Len, • C1_Max_Data_Len plus 4 Bytes (Function Num, Slot_Number, Length)
7	Diag_Update_Delay	[5] 6.2.12.7	The following GSD value is used [6]:Diag_Update_Delay
8	Alarm_Mode	[5] 6.2.12.8	
9	Add_SI_Flag	[5] 6.2.12.9	
10	C1_Timeout	[5] 6.2.12.10	The following GSD value is used [6]:C1_Response_Timeout
11			
12	Reserved		
13			
14			
15			

6.5.2.2.2 Part 2: Parameter data

Part 2 of the BMCP are the slave parameter data.

Table 4 – ~~BMPC~~ BMCP Part2 – Parameter data

Byte	Name	Defined in	Notice
16	Prm_Data_Len	[5] 6.2.12.11	Length of the Parameter Data including this value
17			
18	Station Status	[5] 6.2.4.1	The following GSD values are used [6]: • Bit 0-2: reserved, • Bit 3: WD_On, • Bit 4: Freeze_Req, • Bit 5: Sync_Req, • Bit 6,7(Lock/Unlock Request): [5] Table 402
19	WatchDog1	[5] 6.2.4.2	These values (WD_Fact_1 and WD_Fact_2) depend on the baud rate. A master should set these values and slaves should handle new values
20	WatchDog2	[5] 6.2.4.3	
21	Min Tsdr	[5] 6.2.4.4	Default value is 11 bit times [1]
22	Ident_Number	[5] 6.2.3.5	The following GSD value is used [6]: Ident_Number

Byte	Name	Defined in	Notice
23			
24	Group_Ident	[5] 6.2.4.5	Indicates the group assignment of the slave in a bitwise coded form
25	DPV1_Status_1	[5] 6.2.4.7	The following GSD values are used [6]: • Bit 0,1: reserved, • Bit 2: WD_Base_1ms, • Bit 3 – 5: reserved, • Bit 6: Fail_Safe, • Bit 7 : DPV1_Enable, These bits report the slave capabilities to the master and are changed by the MS0 master following its capabilities
26	DPV1_Status_2	[5] 6.2.4.8	The following GSD values are used [6]: • Bit 0: Check_Cfg_Mode, • Bit 1: reserved, • Bit 2: Enable_Update_Alarm, • Bit 3: Enable_Status_Alarm, • Bit 4: Enable_Manufacturer_Specific_Alarm, • Bit 5: Enable_Diagnostic_Alarm, • Bit 6: Enable_Process_Alarm, • Bit 7: Enable_Pull_Plug_Alarm, These bits report the slave capabilities to the master and are changed by the MS0 master following its capabilities
27	DPV1_Status_3	[5] 6.2.4.9	• Bit 0-2: - , • Bit 3: Prm_Structure, • Bit 4: IsoM_Req, • Bit 5-6: reserved, • Bit 7: Prm_Cmd
28	User_Prm_Data		From Byte 28 of the BMCP, a DTM has to insert additional user parameter data. The information is given in the GSD file via value of User_Prm_Data
...			
NOTE 1 It is possible that a module does not have user parameter data. In this case, no parameter string is inserted into the BMCP for this module. NOTE 2 Some slaves have some fixed modules besides the BIM. Even if these modules do not appear in the configuration dialog or are not represented as Module DTMs, they have to be considered in the BCMP if they have parameters. NOTE 3 The three DPV1 status bytes are defined by the MS0 master.			

6.5.2.2.3 Part 3: Configuration data

The configuration data are provided as part 3:

Table 5 – ~~BMPC~~ BMCP Part3 – Configuration data

Byte	Name	Source	Notice
16+Prm_Data_Len	Cfg_Data_Len	[5] 6.2.12.13	Length including this value
16+Prm_Data_Len + 1			
16+Prm_Data_Len +2	Cfg_Data		Configuration data (if available)
...			

From byte position (16+Prm_Data_Len +2), the configuration strings for the BIM and the modules are provided in ascending order. The information is given in the GSD file via value module.

NOTE 1 Some slaves have some fixed modules besides the BIM. Even if these modules do not appear in the configuration dialog or are not represented as Module DTMs, they have to be considered in the BCMP if they have parameters.

NOTE 2 Empty slots have to be considered in the configuration data. Refer to the GSD file as to which configuration string has to be used.

6.5.2.2.4 Part 4: Address table and slave user parameters

In Part 4 of the BMCP, the address table and the slave user parameter section is provided.

Table 6 – Part 4: Address table and slave user parameters

Byte	Name	Source	Hints
16 + Prm_Data_Len + Cfg_Data_Len	Add_Tab_Len	[5] 6.2.12.15	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len	Slave_User_Data_Len	[6] 6.2.12.21	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 2	Slave_User_Data		Slave user parameter (if available)
...			

After Add_Tab_Len the address table data is inserted into the BMCP. If the master does not support this table the Add_Tab_Len is set to 2 and no data is inserted.

After Slave_UserDataLen the slave user data is inserted into the BMCP. If the master does not support this data the Slave_User_Data_Len is set to 2 and no data is inserted.

Table 7 – Part 4: Extended Prm data

Byte	Name	Source	Hints
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len	Ext_Prm_Data_Len	[5] 6.2.12.23	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len + 1			

After Ext_Prm_Data_Len, the extended prm data is inserted into the BMCP. If the master does not support this data the Ext_Prm_Data_Len is set to 2 and no data is inserted.

6.5.2.2.5 Complete slave bus parameter set

All values are provided from the Slave DTM. It is responsibility of the Slave DTM to be compatible with the Slave GSD. The Master DTM could change some of these initial values sent by Slave DTM if they depend on the capabilities of the master.

Example

Within the GSD file, it is stated that the device supports the Freeze Mode by the keyword "Freeze_Mode_supp". The master sets the value "Freeze_Mode_Req" within the Slave Bus Parameter Set because only the master knows whether it supports this mode.

The following table explains which component is the source of the parameter values ("Parameter provided by"). Some of the values can be changed by the system or by user interaction. For those values it is indicated which component is allowed to change the values ("Configuration by"). If possible, the default values for the parameters are defined ("Default Value").

Table 8 – Complete BMCP

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
0	Slave_Para_Len	Length of the BMCP including this value	Calculated by Slave DTM	-	-
1					
2	SI_Flag	Bit coded. The meaning of the bits:	-	-	-
		Bit 0: reserved	-	-	0
		Bit 1: Extra_Alarm_SAP	Slave DTM shows with this bit that the feature is supported	Master DTM	Extra_Alarm_SAP_sup p within GSD
		Bit 2: DPV1_Data_Types	Slave DTM shows what data types are used by the slave DTM	Master DTM	DPV1_Data_Types within Slave GSD
		Bit 3: DPV1_Supported	Slave DTM shows with this bit that the feature is supported	Master DTM	DPV1_Slave within GSD
		Bit 4: Publisher_enable	Slave DTM shows with this bit that the feature is supported	Master DTM	Publisher_sup within GSD
		Bit 5: Fail_Safe	Slave DTM can show by this flag that the slave needs fail safe (Fail_Safe or Fail_Safe_Required of the GSD)	Master DTM	Value within Slave GSD
		Bit 6: New_Prm	If the Slave DTM set this value, it requests that the MS0 master should set the changed slave parameter. (see [5] 6.2.12.2)	-	0
		Bit 7: Active	If the Slave DTM has finished the module configuration set it to 1.	-	0
3	Slave Type	0 = DP-Slave	Slave DTM shows that it is a DP-Slave	-	0
4	Max_Diag_Data_Len		Slave DTM provides this information by Max_Diag_Data_Len in the GSD	-	Value within Slave GSD
5	Max_Alarm_Len	Length of the alarm structure see [5] table 396	Slave DTM, Conditions: One of GSD keywords:- Diagnostic_Alarm_sup p - Process_Alarm_sup - Pull_Plug_Alarm_sup - Status_Alarm_sup - Updata_Alarm_sup - Manufacturer_Specific_Alarm_sup OR - Alarm_Type_Mode_sup	-	Is calculated on the base of the different GSD values.
6	Max_Channel_Data_Len		Slave DTM: This field defines how much data can be transferred between slave and master. In this case the maximum of these values must be calculated and set by the slave DTM. Rule for calculation: Max_Data_Len C1_Max_Data_Len plus 4 Bytes (Function Num, Slot_Number, Length)	-	Value within Slave GSD
7	Diag_Update_Delay		Slave DTM	-	Slave GSD [6] Diag_Update_Delay

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
8	Alarm_Mode		Slave DTM	-	Slave GSD [6] Alarm_Sequence_Mode_Count
9	Add_SI_Flag	Bit coded. Following the meaning of the bits			
		Bit 0: NA_To_Abort		Master DTM	0
		Bit 1: Ignore_Aclr		Master DTM	0
		Bit 2 to Bit 7: reserved			0
10	C1_Timeout	The following GSD value is used [6]: C1_Response_Timeout	Slave DTM		Slave GSD [6] C1_Response_Timeout
11					
12	Reserved				0
13	Reserved				0
14	Reserved				0
15	Reserved				0
16	Prm_Data_Len	Length of the Parameter Data including this value	Calculated (or maybe fixed) by Slave DTM		
17					
	PRM_DATA				
18	Prm_Data (Station Status)	Bit 0-2: reserved		0	
		Bit 3: WD_On	MS0 Master defines that watchdog is used or not. If watchdog is enable the master has also to set WD_Fact_1 and WD_Fact_2	Master DTM	0
		Bit 4: Freeze_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	Freeze_Mode_supp within GSD
		Bit 5: Sync_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	(Sync_Mode_supp within GSD)
		Bit 6,7: Lock/Unlock Request [5] Table 402	Master DTM	0	
19	Prm_Data (WatchDog1) WD_Fact_1	These values depend on the baud rate. A master should set these values and slaves should handle new values. Consider within Usr_Prm_Data (DPV1_Status_1, Bit 2 WD_Base_1ms)		Master DTM	1

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
20	Prm_Data (WatchDog2) WD_Fact_2	See Byte 19, Watchdog_Time = 10 ms * WD_Fact_1 * WD_Fact_2 OR Watchdog_Time = 1 ms * WD_Fact_1 * WD_Fact_2		Master DTM	1
21	Prm_Data (Min Tsdr)	Default value is 11 bit times [1].		Master DTM	11
22	Prm_Data (Ident_Number)		SlaveDTM	-	Slave GSD [6] Ident_Number
24	Prm_Data (Group_Ident)	This means the slave is not assigned to a group for global control. It is set by the master		Master DTM	0
25	Prm_Data (DPV1_Status_1)	Bit 0,1: reserved			0
		Bit 2: WD_Base_1ms (See Bytes 19 and 20)	Slave DTM shows with this bit that the feature is supported	Master DTM In some rare case there is a need to set the value to 1	WD_Base_1ms_supp within GSD
		Bit 3 – 5: reserved			0
		Bit 6: Fail_Safe	Slave DTM shows with this bit that the feature is supported	Master DTM	(Fail_safe within GSD)
		Bit 7: DPV1 Enable	Slave DTM	Master DTM	Slave GSD (If the GSD minimum include one of these setting : C1_Read_Write_supp = 1 or Diagnostic_Alarm_sup p = 1 or Process_Alarm_supp = 1 or Pull_Plug_Alarm_supp = 1 or Status_Alarm_supp = 1 or Update_Alarm_supp = 1 or Manufacturer_Specific _Alarm_supp = 1 . The slave supports DPV1 and then the DPV1_Enable should be 1

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
26	Prm_Data (DPV1_Status_2)	Bit 0 : Check_Cfg_Mode			
	Slave DTM shows with this bit that the feature is supported.	Master DTM	Check_Cfg_Mode within GSD		
		Bit 1 : reserved	-	-	0
		Bit 2 : Enable_Update_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Update_Alarm_supp within GSD
		Bit 3 : Enable_Status_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Status_Alarm_supp within GSD
		Bit 4: Enable_Manufacturer_Specific_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Manufacturer_Specific_Alarm_supp within GSD
		Bit 5: Enable_Diagnostic_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Diagnostic_Alarm_supp within GSD
		Bit 6: Enable_Process_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Process_Alarm_supp within GSD
		Bit 7: Enable_Pull_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Pull_Plug_Alarm_supp within Slave GSD
27	Prm_Data (DPV1_Status_3)	Bit 0-2: Alarm_Mode	Slave DTM	Master DTM	Data within Slave GSD
		Bit 3: Prm_Structure.	Slave DTM shows with this bit that the feature is supported		Prm_Block_Structure_supp within GSD
		Bit 4: IsoM_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	Isochron_Mode_supp within GSD
		Bit 5-6: reserved	-	-	0
		Bit 7: Prm_Cmd	Slave DTM shows with this bit that the feature is supported	Master DTM	PrmCmd_supp within GSD
28 to ...	Prm_Data(Usr_Prm_Data)		Calculated by Slave DTM	Slave DTM	Data within GSD
X1	Cfg_Data_Len	Length including this value	Calculated by Slave DTM	-	Data within GSD
X1+1					
X1+2 to ...	Cfg_Data	Cfg_Data	Slave DTM, Depending on Module Configuration	Slave DTM	Data within Slave GSD (Module, EndModule)
X2	Add_Tab_Len	Length including this value	Calculated by the Communication DTM		
X2 + 1					

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
X2 + 2 to	Add_Tab_Data	Address assignment Table (only for DPV0 Masters)	Calculated by the Communication DTM		
X3	Slave_User_Data_Len	Length including this value	Calculated by the Communication DTM		
X3 + 1					
X3 + 2 to ...	Slave_User_Data	Slave_User_Data	Calculated by the Communication DTM		Data within GSD
X4 + 1	Ext_Prm_Data_Len	Length including this value	Calculated by Slave DTM	-	
X4 + 2					
X4	Ext_Prm_Data	Extended Parameter Data	Slave DTM	-	Data within GSD
NOTE X1 = 16+Prm_Data_Len X2 = 16 + Prm_Data_Len + Cfg_Data_Len = 16 + x1 + Cfg_Data_Len X3 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len = X2 + Add_Tab_Len X4 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len = X3 + Slave_User_Data_Len					

6.5.2.3 Modification of the BMCP

6.5.2.3.1 Propagation of changes

The BMCP includes parameter and configuration data. The slave DTM or the FDT Frame Application may modify the BMCP.

The system has to ensure that the Communication DTM representing the PROFIBUS master is aware of these changes. This is achieved by sending the event `InstanceDataChanged` from the Slave DTM to report the change of the BMCP. All changes should be reported as soon as possible, but not before the changes are persistent. The Frame Application informs the Parent DTM via `OnChildInstanceDataChanged`, that parameters of a child have been changed. Then the Communication DTM can get the new BCMP of the Slave DTM.

6.5.2.3.2 When it is allowed to change the BMCP

According to the FDT specification is allowed to change the parameters of a DTM starting from “configured” state (see IEC 62453-2).

The BCMP can be changed multiple times, but only if the DTM is in offline mode and if the data set can be locked.

If a Slave DTM wants to change parameters in online mode it must use DPV0 or DPV1 transaction requests. If there is no way of changing the parameters by transaction requests, the DTM must disable configuration and parameterization in the online state.

6.5.2.3.3 Parameter data

If the user changes parameters of the BIM or one module (for example via user interface) and this affects the BMCP; the DTM has to update the BMCP. In addition to this, it must request a save and inform the Frame Application via `InstanceDataChanged`. The Frame Application must distribute the information to all relevant components.

6.5.2.3.4 Configuration data

Configuration data will change every time if the user adds/removes modules or changes the module type, etc.

In the case of the modular DTM, the BIM will be informed when the user adds or removes modules via the service `ChildAdded` and service `ChildRemoved`. Changes of the parameters in module will be reported by service `OnChildInstanceDataChanged`.

The monolithic DTM or the BIM DTM update their BMCP, request a save and inform the Frame Application via `InstanceDataChanged`.

NOTE 1 Changes that affect the BMCP often take effect on the internal topology and the Process Channels. This information must be updated by the DTMs too.

NOTE 2 The BMCP may be changed by the Slave DTM and by the Communication DTM. The Communication DTM must not change the configuration data and the user parameters.

6.5.2.4 Special cases related to the BMCP

6.5.2.4.1 DPV1 support

In the GSD file there are two flags regarding DPV1. At first, the “DPV1_Slave” value: This means that the slave has the possibility to work as a DPV1 slave. If this value exists and the value is “1” then the respective bit within the `SI_Flag` of Slave Bus Parameter Set must be set. For older systems, there should also be the possibility to work as a DPV0 slave.

Only the Communication DTM knows that its master device is able to provide acyclic services.

After building the Slave Bus Parameter Set, the Communication DTM will receive the slave's initial BMCP. If, within the SI_Flag, the respective bit is set, the Master DTM must set the highest bit in the first byte of Extended DPV1 Status. Now the slave works as a DPV1 slave.

6.5.2.4.2 Extended DPV1 status

All the bits in the 3 bytes Extended DPV1 Status are set by the Communication DTM for the master. A Slave DTM must accept these settings and adapt its functionality if necessary.

7 Protocol specific usage of general data types

The following table (Table 9) explains how data types, defined in IEC 62453–2 within the namespace 'fdt', are used with CP 3/1 AND CP 3/2 protocols.

Table 9 – Protocol specific usage of general data types

Data type	Description for use in IEC 61784 CPF 3
fdt:address	Profibus Parameter Address: The attribute 'address' follows the different device models that are defined for PROFIBUS devices. FDT currently supports the following models: PROFIBUS DP / DPV1, PROFIBUS PA, PROFIdrive (greater or equal profile version 3) PROFIBUS DP / DPV1 The device model is based on devices that are composed of slots, whereas slots do not have to represent physical objects. The data that is contained in the slots, are addressable via Indexes. This data may be variables or composed blocks of data. The address attribute is APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0' PROFIBUS PA The device is represented by a device management structure and a number of blocks that provide different functionality (physical block, function block, transducer block). The blocks are mapped to slot addresses, but this mapping may vary depending on the device type. The address attribute is APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0' PROFIdrive According to the PROFIdrive profile, a device (drive unit) may be composed by a number (1-many) of drive objects (DOs). The DOs may have different type. Each DO is uniquely identifiable and manages its own parameters. Each parameter can be uniquely identified by its number (PNU). Each DO has its own number space. A parameter may contain simple data or composed data (e.g. arrays). The data of the device are accessible via a parameter channel (normally slot 0 index 47). The address attribute is APIxxSLOTyyINDEXzz.DODO-id.pnu xx API yy Slot zz Index

Data type	Description for use in IEC 61784 CPF 3
	do-id Drive Object ID pnu ParameterNumber xx, yy, zz, do-id, pnu are based on decimal format without leading '0'
fdt:protocolId	See Clause 4
fdt:deviceTypeId	The element "fdt:deviceTypeId" shall contain the IDENT_NUMBER of the supported physical device. The IDENT_NUMBER shall be entered in decimal format; however, the value should be displayed as hex to the user
fdt:deviceTypeInfo	A PROFIBUS device has to provide its GSD information as human readable string at this attribute. NOTE The GSD information is accessible via the services NetworkManagementInfoRead GetTypeInfo
fdt:deviceTypeInfoPath	Path to the file containing the information which is provided via the attribute 'deviceTypeInfo'. In case of PROFIBUS the attribute contains the full path to the GSD file including the file name. The file name depends on the current locale according to the usage of service SetLanguage. For PROFIBUS devices it is mandatory to provide this attribute. Examples English: 'C:\MyFolder\ABCD.GSE' German: 'C:\MyFolder\ABCD.GSG'
fdt:manufacturerId	Manufacturer according to Profile specification. For example in Profibus PA : Physical Block Index 10 : DEVICE_MAN_ID
fdt:semanticId fdt:applicationDomain	The SemanticIDs for PROFIBUS follow the different device models that are defined for PROFIBUS devices. FDT currently supports the following models: PROFIBUS DP, PROFIBUS PA, PROFIdrive. PROFIBUS PA The applicationDomain is: FDT_PROFIBUS_PA The device is represented by a device management structure and a number of blocks that provide different functionality (physical block, function block, transducer block). The blocks are mapped to slot addresses, but this mapping may vary depending on the device type. Since the device model is based on blocks, the SemanticIds also are based on the block model. Within each block, the data is identifiable by names of parameters. The semanticId for PROFIBUS profile related parameter follows the following rules: - the semanticId shall be built based on the names defined in the profiles - structured parameters shall be combined with a '.'; - spaces within the profile definition shall be exchanged with an underscore; - blocks shall be counted according to the Object Dictionary; - the block number shall be part of the semanticId. The semanticId is: BlockType.BlockIndex.NameOfParameter.AttributeOfParameter Example AnalogInputFB.3.OUT.Unit PROFIdrive The applicationDomain is: FDT_PROFIBUS_PROFIDRIVE According to the PROFIdrive profile, a device (drive unit) may be composed by a number (1-many) of drive objects (DOs). The DOs may have different types. Each DO is uniquely identifiable and manages its own parameters. Each parameter can be uniquely identified by its number (PNU). Each DO has its own number space. A parameter may contain simple data or composed data (e.g. arrays). The data of the device are accessible via a parameter channel (normally slot

Data type	Description for use in IEC 61784 CPF 3
	0 index 47). The semanticId is: DOdo-id.PNUpnu do-id Drive Object ID pnu ParameterNumber do-id, pnu are based on decimal format without leading '0' Example DO3.PNU64 PROFIBUS DPV1, The applicationDomain is: FDT_PROFIBUS_DPV1 The device model is based on devices that are composed of slots, whereas slots do not have to represent physical objects. The data that is contained in the slots are addressable via Indexes. This data may be variables or composed blocks of data. The semanticID for devices not based on a profile is directly based on the PROFIBUS address information: The semanticId is: APIxx.SLOTyy.INDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0'
fdt:subDeviceType	Manufacturer specific value

8 Protocol specific common data types

Not applicable.

9 Network management data types

9.1 General

The data types specified in this subclause are used at following services:

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

The data type net:DeviceAddress (defined in IEC 62453–2) is used for defining the network address of a device.

The protocol specific data types are based on definitions given in the IEC 61784 and IEC 61158 specifications. Furthermore, they contain additional information about the device that is needed by systems to configure CPF 3 links and to establish communication between the CPF 3 master device and the CPF 3 slave devices.

9.1.1 Configuration

The configuration of the device itself is done with the aid of the DTM's GUI. Downloading the configuration into the slave device is performed via the CPF 3 master device. To do that and in order to set up the bus communication, the master needs information from the DTM as there is:

- GSD file
The GSD information is type-specific information and not instance-specific. It is not stored with single slave instances or in a global accessible file. It is provided by the DTM at the

service `GetTypeInformation`. On this service, a DTM of a PROFIBUS device provides the GSD information within its XML document.

The master device can use the general type-specific information from the slave's GSD information like bus timing parameters, supported baud rates, etc.

- **CFG-String (Cfg_Data)**

The CFG-String provides the instance-specific information about the current configuration of the device. It defines the structure of the data frames that will be transmitted on the PROFIBUS. This structure depends on the modules that are actually configured.

The DTM provides the CFG-String within the attribute `busMasterConfigurationPart` that is part of the XML document available via service `InstanceDataRead`. The structure of the `busMasterConfigurationPart` is defined according to the PROFIBUS-DP-Slave-Bus-Parameter-Set (see IEC 62453-2 and also IEC 61158 series).

The master device uses this information to set up communication with the slave device.

9.1.2 Process Channel

In case of CPF 3 protocols, an FDT-Channel is a representative for a single date or a process value that can be accessed from a Frame Application via the master device. The information available at services for IO related information describes how to access a channel via a PROFIBUS DPV1 command or how to address a channel within a PROFIBUS DP frame for cyclic I/O. Besides all mandatory elements (which include `id` and `dpAddress`) it is highly recommended that the DPV1 address information is provided. This information (DPV1 Slot) is used by some frames to manage the PROFIBUS device module information.

In a DPV0 environment, depending on the situation, the underlying master device may have either Master Class 1 functionality or Master Class 2 functionality. A Class 1 master can write output data to a device and control data exchange, where a Class 2 master can only read the output data. Generally, it is assumed that parameterization as described here is performed as a master Class 2 station.

9.1.3 Parameterization

There are two options to write parameters set from the DTM's GUI to the CPF 3 slave device in the field:

- **User Parameters**

User Parameters are part of the PROFIBUS-DP-Slave-Bus-Parameter-Set. They contain manufacturer-specific data to characterize the DP-Slave. The DTM writes the User Parameters to the `busMasterConfigurationPart`. The User Parameters are stored with the master device during PROFIBUS master configuration and are automatically sent to the slave during set up of bus communication. (This is PROFIBUS-specific; for details, see IEC 61158 series.) When changing User Parameters on runtime, the DTM shall use a DP V0 connection and the appropriate DP V0 commands for parameter exchange as described in the data types.

- **Writing Parameters with DP-V1 services (MSAC2 services)**

The DTM may use DP-V1 transport services to send its parameters to the slave device. For that, it has to use a DP-V1 connection and the corresponding communication commands. During setup of communication, DP-V1 services are not sent automatically. The Frame Application or a DTM shall invoke a download of parameters via DP-V1.

For details on the different behavior of slaves depending on the kind of parameterization, refer to the IEC 61158 series.

DP-V1 connections and communication commands can also be used to execute commands at the slave. For details on the use of DP-V1, see also the IEC 61158 series.

9.2 Master-bus parameter set

The following parameter set represents the contents of the data type busMasterConfigurationPart within the device instance data for Profibus master device (Table 10). This attribute has to be set for each Profibus master device according to the IEC 61158 series. For further details, please refer to the sequence chart “Configuration of a Fieldbus Master” and the IEC 61158 series.

Table 10 – Bus parameter set for master device

Name	Type	Comment
Bus_Para_Len	UINT	Length of the bus parameter set inclusive, the length parameter (range 34 to 216-1)
FDL_Add	USINT	Mandatory part of the Bus Parameter Set according to PROFIBUS-DP Specification; not used in this context as station address is transferred in a separate variable; FDL-Add may differ from real station address used by EE and DTMs
Baud_rate	USINT	Code number for the baud rate
TSL	UINT	Slot time
min TSDR	UINT	Minimum station delay response
max TSDR	UINT	Maximum station delay response
TQUI	USINT	Quit-time
TSET	USINT	Setup-time
TTR	UDINT	Target rotation time
G	USINT	GAP update factor
HSA	USINT	Highest station address
max_retry_limit	USINT	Maximum retry limit
Bp_Flag	USINT	Flags for the user interface, e.g. error action flag
Min_Slave_Interval	UINT	Smallest allowed time period between two slave poll cycles
Poll_Timeout	UINT	Master-master timeout
Data_Control_Time	UINT	Guaranteed time period between two Data_transfer_list updates
Octet 1 (reserved)	ARRAY[6] of USINT	
...		
Octet 6 (reserved)		
Master_User_Data_Len	UINT	Length of Master_User_Data inclusive length parameter
Master_Class2_Name	ARRAY (32) of CHAR	Name of DP class 2 master the parameter set was created with
Master_User_Data	ARRAY of USINT	Manufacturer specific

9.3 Slave bus parameter set

The following parameter set represents the contents of the data type busMasterConfigurationPart within the instance data set for Profibus slave devices (Table 11). This attribute has to be set for each Profibus slave device according to the IEC 61158 series. For further details, please refer to the sequence chart “Configuration of a Fieldbus Master” and the IEC 61158 series.

Table 11 – Bus parameter set for slave device

Name	Type	Comment
Slave_Para_Len	UINT	Length of the slave parameter set inclusive the length parameter
Sl_Flag	USINT	Slave specific flags like New_Prm, Active, Fail_Safe,
Slave_Type	USINT	Manufacturer specific slave type denotation (0 by default for DP-Slaves)
Octet 1 (reserved)	ARRAY[12] of USINT	
...		
Octet 12 (reserved)		
Prm_Data_Len	UINT	Length of Prm_Data inclusive the length parameter (range 9 to 246)
Prm_Data	ARRAY of USINT	
Cfg_Data_Len	UINT	Length of Cfg_Data inclusive the length parameter (range 3 to 246)
Cfg_Data	ARRAY of USINT	
Add_Tab_Len	UINT	Length of Add_Tab inclusive the length parameter (range 2 to 216-31)
Add_Tab	ARRAY of USINT	Address assignment table
Slave_User_Data_Len	UINT	Length of Slave_User_Data inclusive the length parameter (range 2 to 216-31)
Slave_User_Data	ARRAY of USINT	Manufacturer specific data to characterize the DP-Slave for the master

9.4 Module and channel data

A slave's current module configuration including the belonging channels has to be available by FdtChannel objects. That is required as within the environment process variables have to be assigned to single channels and the slave configuration has to be displayed in the environment's system overview without using the DTM's user interface.

The addressing of channels within the PROFIBUS data frame is bit oriented and data type independent. A bit position and a bit length determine each channel. The data type of a parameter determines how to convert the bit field.

- Channels within PROFIBUS data frames (Figure 3):

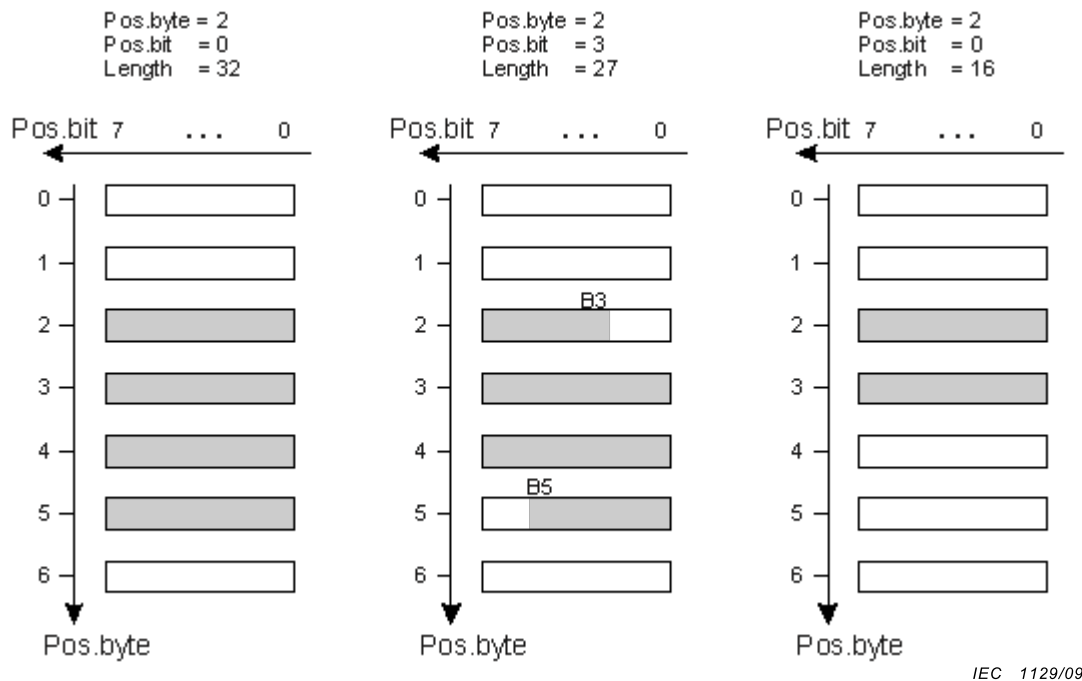


Figure 3 – Example for IO data within datagrams

- Representation in FDT Notation:

bitPosition="16"

bitPosition="19"

bitPosition="16"

bitLength="32"

bitLength="27"

bitLength="16"

The following example shows the expected structure description of a modular slave in a DTM parameter set using FDT-Parameters with the object types defined in the FDT-Specification.

- Representation of the data frame using GSD information:

BeginSlave;;

RIOLB8101.GSD;6;

BeginModules;;

1X03 FrequencyCount;0x51;

Empty ;0x00;

Empty ;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

2XXX ValveBlock.;0x30;

EndModules;;

EndSlave;;

Description of the position of signal channels within the data frame ():

Table 12 – Signal channels within the data frame

# Frame description Remote I/O											
Vendor:	"RIO manufacturer"										
Devicetype :	"LB 8101"		Device ident:			0x8101	# hex				
Frameident:	"B1t1"										
Input data											
Channel	Channel type	Data	Data	Data	Invalid Bit	Invalid Bit	Subst. Value Bit	Subst. Value Bit	Sim. Bit	Sim. Bit	State-Channel
		Pos. byte	Pos. bit	Length	Pos. byte	Pos. bit	Pos. byte	Pos. bit	Pos. byte	Pos. bit	
Module:	"EP01"	"Digital Input (Counter) LB/FB 1X03 (4 Byte)"									
"Count_1_0"	READ INT	0	0	32	-/-	-/-	-/-	-/-	-/-	-/-	0
Module:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input and Output)"									
"DI_8_LFOUT0"	READ BOOL	4	1	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF1"	READ BOOL	4	3	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF2"	READ BOOL	4	5	1	-/-	-/-	-/-	-/-	-/-	-/-	1
Output data											
Channel	Channel type	Data	Data	Data	Invalid Bit	Invalid Bit	Subst. Value Bit	Subst. Value Bit	Sim. Bit	Sim. Bit	State-Channel
		Pos. byte	Pos. bit	Length	Pos. byte	Pos. bit	Pos. byte	Pos. bit			
Modul:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input u. Output)"									
"DO_8_0"	WRITE BOOL	0	0	1	4	1	-/-	-/-	-/-	-/-	0

9.5 GSD information

9.5.1 General

The GSD information is not stored with single slave instances or in a global accessible file. It is provided by the DTM via the service `GetTypeInformation`. A DTM of a PROFIBUS device provides the GSD information within the `fdt:DtmDeviceType` argument.

The existing DCS uses the GSD information to obtain information about the possible configuration and parameters of a DP Slave.

Besides the information about modules and its parameters, a GSD file contains additional information about the slave, such as the supported baud rates.

This information is useful for a DCS system to configure an entire network according to the capabilities of different slaves.

9.5.2 GSD for gateway devices

9.5.2.1 Types of Profibus gateway devices

There are two types of gateway devices.

The visible gateway devices work as a PROFIBUS slave (with a PROFIBUS address) and to the underlying network they act as a master.

The invisible gateway devices just transform the PROFIBUS network to the underlying network.

For both types of devices there is a need for special GSDs.

9.5.2.2 Visible gateway devices

These visible gateway devices are shipped without a GSD file. Instead they have a proprietary software suite that configures and parameterizes it or they are shipped with a tool that creates a GSD for parameterization software. The GSD tool creates a GSD for the slave depending on the underlying network configuration or bus settings (for example baud rates).

In these cases the DTM should provide the functionality of the GSD tool. If the GSD is configuration-dependent, it could call service `NetworkManagementInfoRead` for each of its children, extract the GSD information and create configuration-dependent GSD information in the same way the tool does. After initialization of the DTM, it should deliver a BMCP according to the linking device itself. Every time a child is added or removed, this leads to a change in its own parameter document.

If the GSD depends on bus settings, a DTM's configuration or parameterization dialog could be used to change bus settings. Based on these settings, updated GSD information can be inserted in the information document. Here too the DTM has to request a save and call service `OnInstanceDataChanged`.

Note that the internal structure with its modules and channels has to be updated as well.

9.5.2.3 Transparent gateway devices

There are transparent linking devices on the market (PROFIBUS FMS/DP and PROFIBUS PA) performing a baudrate transformation. This requires a special handling of the slave specific GSD files. There are tools available which are able to adapt existing GSD files according to the higher baudrate (so called 'GSD Converter').

The GSD information must be delivered by the DTM for the device. In order to support this kind of linking devices, a slave DTM must expose the GSD file on hard drive using the attribute 'deviceTypeInformationPath' (refer to DtmDeviceType defined in IEC 62453-2).

A DTM developer should be aware, that the referenced GSD file might be converted in order to support specific system architecture.

It is expected that a Profibus DTM in the attribute 'deviceTypeInformation' is exposing exactly the GSD file which is referenced by the attribute 'deviceTypeInformationPath'.

10 Communication data types

10.1 General

The data types defined here are used at:

- Connect service;
- Disconnect service;
- Transaction service.

10.2 Error information provided by Communication Channel

In every Response data type of FDT Profibus specification an element 'errorCode' is provided. According to PROFIBUS, the error code is standardized to consist of 3 bytes, where each byte carries a meaning.

Since the error code is exchanged between different DTMs (e.g. Communication-DTM and Device-DTM) and since the receiver of the error code will try to understand the error code, the provider shall use the standard format:

- standard length 3 bytes;
- if the device provides error codes, these error codes are provided (and not local error codes from the Master).

If no error occurred, the element 'errorCode' shall be filled with zeros ("000000").

10.3 DPV0 communication

The supported services depend on the type of PROFIBUS master functionality that is provided by the communication infrastructure. Master Class 1 devices typically control the slaves and provide cyclic communication, Master Class 2 devices typically are used to configure the slaves and provide acyclic communication.

Not all defined services are supported if the Master is not in cyclic data exchange with the slaves. In such cases, the following behavior is expected:

If a Communication Channel receives a request that can not be supported, the Transaction service responds with a result="false".

The data types described in this clause are defined for the following namespace:

Namespace: dpv0

Table 13 – Simple DPV0 communication data types

Data type	Definition	Description
busAddress	USINT	Address information according to the IEC 61158 series (see also IEC 62453-2, attribute busAddress)
communicationReference	UUID	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during creation of the connection. The address information has to be used for all following communication calls
connectStatus	enumeration (masterConnectedOnly deviceAtLifeList deviceInDataExchange)	Describes the connection status established by the communication component. The status “masterConnectedOnly” means that the communication component has established a connection to the Profibus master device and will accept an online access to the user parameters, independent of whether the device is available or not. The status “deviceAtLifeList” means that the communication component has established a connection to the Profibus master device and has checked that the device is in the life list of the master stack. In this state, the master will accept an online access to the user parameters and will send the user parameter to the device, independent of whether the device is in data-exchange or not. The status “deviceInDataExchange” means that the communication component has established a connection to the Profibus master device and has checked that the device is in data-exchange. In this state, the master will accept an online access to the user parameters and will send the user parameter to the device, so that the new data will directly influence the process
delayTime	UDINT	Delay time in [ms] between two communication calls
errorCode	ARRAY OF USINT	Status information according to the IEC 61158 series. For description of error code see: DIN 19245 Part 3, PROFIBUS (P. 40ff., 83ff., 39)
schemaVersion	INT	Defines the version of the schema
sequenceTime	UDINT	Period of time in [ms] for the whole sequence

Table 14 – Structured DPV0 Communication data types

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
Abort	STRUCT			Describes the abort	-
	communicationReference	O	[0..1]		
ConnectRequest	STRUCT			Describes the communication request to establish a connection to a Profibus master device. Depending on the used network infrastructure it is possible that this service is not mapped to a field bus request, but is used to manage the software. It is also possible that this service is used as a trigger to set the state of a device	-
	busAddress	M	[1..1]		
	fdt:systemTag	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ConnectResponse	STRUCT			Describes the communication response to the connect request and provides the information how the following WriteUserParameter commands will be sent to the device (see connectStatus). The resulting connection depends on the used communication device and on the configuration of the bus master. The DeviceDTM shall expect that the resulting connection cannot be used to access all services. ConnectResponse may return true even if there is no communication with the device possible. For example if ConnectRequest returns true with connectionStatus = "masterConnectedOnly" this means that there is a connection to the master device, but the device is not at the life list and can therefore not be accessed. Depending on the connectStatus some services may not be available (e.g. in status "masterConnectedOnly" only services provided by the master are accessible (ReadUserParameter, WriteUserParameter, ...) whereas services provided by the device are not accessible. NOTE In this example the DTM is in state "online" but it cannot access the device. The provided services also depend on the type of master device. For an overview of availability of services depending on the type of connection, see the following tables.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
	connectStatus	M	[1..1]		
DisconnectRequest	STRUCT			Describes the communication request to release a connection to a Profibus master device	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
DisconnectResponse	STRUCT			Describes the communication response	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ReadDiagnosisDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification For Master Cl.1 refer to IEC 61158-5, 8.2.3.3.2 “Get slave diag” For Master Cl.2 refer to IEC 61158-5, 8.2.3.3.3 “Read slave diag”	1, 2
	communicationReference	M	[1..1]		
ReadDiagnosisDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadInputDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. Depending on whether the underlying FDTChannel is a Master Cl. 1 or a Master Cl. 2 this service will be local or result in a read access to the slave device. It is possible to specify which data is read by providing an fdt:ChannelReference. For Master Cl.1 refer to IEC 61158-5, 8.2.2.3.3 “Get Input” For Master Cl.2 refer to IEC 61158-5, 8.2.2.3.2 “Read Input”	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadInputDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ReadOutputDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. Depending on whether the underlying FDTChannel is a Master Cl. 1 or a Master Cl. 2 this service will be local or result in a read access to the slave device. It is possible to specify which data will be read by providing an fdt:ChannelReference. For Master Cl. 1 this is a proprietary service (or unavailable). For Master Cl. 2 refer to [1], section 8.2.2.3.6. "Read Output" (an optional service)	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadOutputDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadUserParameterRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The request retrieves only the information that is available from the master device (local service). It may differ from the actual data of the device. The implementation on how to provide the data is proprietary. This is based on a local service or not available	1
	communicationReference	M	[1..1]		
ReadUserParameterResponse	STRUCT			Describes the communication response, read from the Profibus master device, according to the Profibus DPV0 specification. The returned data may reflect only the information that is available from the master device. It may differ from the actual data of the device	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
SequenceBegin	STRUCT			Describes the sequence begin	-
	sequenceTime	O	[0..1]		
	delayTime	O	[0..1]		
	communicationReference	M	[1..1]		
SequenceEnd	STRUCT			Describes the sequence end	-
	communicationReference	M	[1..1]		
SequenceStart	STRUCT			Describes the sequence start	-
	communicationReference	M	[1..1]		
WriteOutputDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The output data will be sent according to the established connection. It is necessary to specify which data will be written by providing an fdt:ChannelReference. If the underlying FDTChannel is provided as Master CI. 2, writing Output Data will be not possible. Refer to IEC 61158-5, 8.2.2.3.5 "Set Output". This functionality depends on the used type of master. In terms of PLC this functionality is named "forcing" of values	1
	communicationReference	M	[1..1]		
	fdt:ChannelReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteOutputDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
WriteUserParameterRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The user parameter will be sent according to the established connection (see connectStatus). For Master Cl. 1 this is based on the service "DDL_M_Set_Prm". For Master Cl. 2 this is based on the service "DDL_M_Set_Prm" (an optional service)	1, 2
	communicationReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteUserParameterResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Depending on the bus master type and on the returned connectStatus following services are available.

Table 15 – Availability of services for Master Class1 (C1)

Slave DTM Service Request	connectStatus		
	masterConnectedOnly	DeviceAtLifeList	DeviceInDataExchange
Connect	✓	✓	✓
ReadUserParameter	O	O	O
WriteUserParameter	✓	✓	✓
ReadOutputData			✓
WriteOutputData			O
ReadInputData			✓
ReadDiagnosisData		✓	✓
NOTE ✓ : the service is available, O : the service is optional and may be available, depending on the capabilities of the underlying master device.			

For Master Class2 (C2), not all connect states are available:

Table 16 – Availability of services for Master Class2 (C2)

Slave DTM Action	connectStatus	
	DeviceAtLifeList (no DPV1 connection to device)	DeviceInDataExchange (DPV1 connection to device)
Connect	✓	✓
ReadUserParameter		
WriteUserParameter	O	O
ReadOutputData		O

Slave DTM Action	connectStatus	
	DeviceAtLifeList (no DPV1 connection to device)	DeviceInDataExchange (DPV1 connection to device)
WriteOutputData		
ReadInputData		✓
ReadDiagnosisData	✓	✓
NOTE ✓ : the service is available, O : the service is optional and may be available, depending on the capabilities of the underlying master device.		

If the Master Class 2 communication component supports DPV1 and DPV0 and has established a DPV1 connection to the device, a call to service Connect for DPV0 shall return the status “DeviceInDataExchange” even if the device is not in status DataExchange.

If no DPV1 connection is established, the Master Class 2 communication component shall verify the availability of the device (at least by service LifeList).

10.4 DPV1 communication

The data types are defined to provide the address information and the communication data required for DPV1 communication (see Table 17 and Table 18).

The data types described in this subclause are defined for the following namespace:

Namespace: dpv1

Table 17 – Simple DPV1 communication data types

Data type	Definition	Description
api	USINT	Address information according to the IEC 61158 series. If the device needs special values for the DPV1-Initiate, the DeviceDTM is responsible for providing the values in the ConnectRequest. If the values are not provided, the communication component will use default values (0)
busAddress	USINT	Address information according to the IEC 61158 series (see also DTMPParameterSchema, attribute busAddress)
errorCode	UUID	Status information according to the IEC 61158 series. For description of error code see: PROFIBUS Nutzerorganisation e.V., PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998. (P.. 120-121); .For description of abort information see: DIN 19245 Part 3, PROFIBUS, (P..54ff., 11)
index	UDINT	Address information according to the IEC 61158 series
communicationReference	ARRAY OF USINT	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during the connect. The address information has to be used for all following communication calls
delayTime	USINT	Minimum delay time in [ms] between two communication calls

Data type	Definition	Description
maxLenDataUnit	USINT	Optional attribute, to describe the amount of data, which can be transferred via the established connection. If this attribute is not available, no special length restriction is announced. Each communication component within the chain of interfaces concerning nested communication could introduce this attribute. Each communication component should change the contents of the attribute based on the following rule: The new value is the minimum of the current value and the restriction of its own implementation. If a communication component has no restriction, it should hand over the given value. If a communication component is able to reuse an established connection concerning a new connect request, it should take into account the data length determined for the existing connection. If the data length is not applicable, the DTM should send an error message via service ErrorMessage
networkMACAddress	ARRAY OF USINT	Network or MAC address as described in the IEC 61158 series. [5] If there is a networkMACAddress, (i.e. networkMACAddress <> ""), the Profibus-DPV1-Master-DTM shall set the corresponding "address type" in the Initiate request telegram "1", else "0". Example Device with address 20 (14hex) addressed via a linking device: networkMACAddress="00000000000014"
scl	INT	Access level as described in the IEC 61158 series. [5] If the device needs special values for the DPV1-Initiate, the DeviceDTM is responsible for providing the values in the ConnectRequest. If the values are not provided, the communication component will use default values (0)
sequenceTime	USINT	Period of time in [ms] for the whole sequence
slot	UDINT	Address information according to the IEC 61158 series

Table 18 – Structured DPV1 communication data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
Abort	STRUCT			Describes the abort
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Describes the communication request
	api	M	[1..1]	
	busAddress	M	[1..1]	
	fdt:systemTag	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Mu l t i p l i c i t y	
ConnectResponse	STRUCT			Describes the communication response
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	maxLenDataUnit	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
DestNetworkAddress	STRUCT			Describes the extended address of the destination [5]
	NetworkAddress	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
NetworkAddress	STRUCT			Describes the extended address format
	api	M	[1..1]	
	scl	M	[1..1]	
	networkMACAddress	M	[1..1]	
ReadRequest	STRUCT			Describes the communication request
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
ReadResponse	STRUCT			Describes the communication response
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
RedundantAddresses	STRUCT			Describes the redundant addresses
	fdtparam:SlaveAddress	M	[1..*]	
SequenceBegin	STRUCT			Describes the sequence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
SequenceEnd	STRUCT			Describes the sequence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Describes the sequence start
	communicationReference	M	[1..1]	
SrcNetworkAddress	STRUCT			Describes the extended address of the source
	NetworkAddress	M	[1..1]	
WriteRequest	STRUCT			Describes the communication request
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
WriteResponse	STRUCT			Describes the communication response
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	

Within a connect request, a DTM of a Profibus redundant slave can provide additional redundant slave addresses, set within its parameter document. The busAddress attribute is to be used as the preferred address; the addresses in the redundantAddresses element should be used in this order if an alternative address is used to connect to the redundant slave.

11 Channel parameter data types

The following table (Table 19) describes the mapping of DPV1 Data types to FDT data types.

Table 19 – Mapping of DPV1 data types to FDT data types

PROFIBUS data types	FDT data type
Boolean	BOOL
Integer8	SINT
Integer16	INT
Integer32	DINT
Unsigned8	USINT
Unsigned16	UINT
Unsigned32	UDINT
Floating Point	FLOAT
Visible String	STRING
Octet String	ARRAY of USINT
Date	DATE
Time of Day	TIME
Time Difference	duration
DS-32	structured

PROFIBUS data types	FDT data type
DS-33	structured
DS-34	structured
DS-35	structured
DS-36	structured
...	structured
DS-52	structured
..	structured

The format of all data types is defined in the IEC 61158 series.

If only a part of the retrieved data is process data (e.g. for DS-33 the data consist of the float-value and a byte status), the access information (DPAAddress or DPV1Address) also contains information that is necessary to retrieve the information from the data structure.

Used at ReadChannelInformation service and WriteChannelInformation service.

The Channel **P**parameter data types describe a how to access a channel via a Profibus DPV1 command or how to address a channel within a Profibus DP frame for cyclic I/O. The following tables (Table 20 and Table 21) provide a description of the data types.

The data types described in this clause are defined for the following namespace:

Namespace: profichannel

Table 20 – Simple Channel Parameter data types

Data type	Definition	Description
api	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
bitLength	UDINT	Additional data type information especially for fieldbus specific data types like 12 bit integer
bitPosition	UDINT	Address information according to the IEC 61158 series for channels accessible via Profibus DP
frameApplicationTag	STRING	Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces
gatewayBusCategory	UUID	Unique identifier for a supported bus type like Profibus or HART according to the FDT specific CATID
index	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
invalidBit	UDINT	Bit position of the invalid status channel accessible via Profibus DP
logic	UDINT	Additional data type information: positive 0=FALSE 1=TRUE
number	BOOL	Address information for diagnosis according to the IEC 61158 series for channels accessible via Profibus DP. See [5], 9.3.1 Read DP – Slave Diagnostic Information
protectedByChannelAssignment	INT	TRUE if the channel is set to read only by the Frame Application. Usually set to TRUE if a channel assignment exists
schemaVersion	UDINT	Defines the version of the schema
simulationBit	USINT	Bit position of the simulation status channel accessible via Profibus DP
slotNumber	BOOL	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
statusChannel	UDINT	TRUE if the channel is for status information only
substituteValueBit	STRING	Bit position of the substitute status channel accessible via Profibus DP

Table 21 – Structured Channel Parameter data types

Data type	Definition			Description
	Elementary data type	U s a g e	Multi- plicity	
DpAddress	STRUCT			Address information according to the IEC 61158 series for channels accessible via Profibus DP
	bitPosition	M	[1..1]	
	bitLength	M	[1..1]	
DpV1Address	STRUCT			Address information according to the IEC 61158 series for channels accessible via Profibus DPV1. It is highly recommended to provide this address information to enable assignment of logical addresses to a profibus module
	api	M	[1..1]	
	slotNumber	M	[1..1]	
	index	M	[1..1]	
	bitPosition	O	[0..1]	
	bitLength	O	[0..1]	
FDTChannel	STRUCT			Description of the channel
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannel- Assignment	M	[1..1]	
	number	M	[1..1]	
	fdt:dataType	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	applId:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	DpAddress	O	[0..1]	
	DpV1Address	O	[0..1]	
	StatusInformation	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:StructuredElements	O	[0..1]	

Data type	Definition			Description
	Elementary data type	U s a g e	Multi- plicity	
FDTChannelType	STRUCT			Description of the channel component in case of channels with gateway functionality
	fdt:VersionInformation	M	[1..1]	
	gatewayBusCategory	O	[0..1]	
	statusChannel	O	[0..1]	
StatusInformation	STRUCT			Description of additional status information for channels accessible via Profibus DP
	logic	M	[1..1]	
	invalidBit	O	[0..1]	
	simulationBit	O	[0..1]	

12 Device identification

12.1 General

There are different Profibus specific identification elements.

A Profibus Scan may detect different device types: I&M devices, Profile devices or pure DP devices.

Following rule shall be applied for Profibus Communication Channels:

- if I&M is available, create I&M identification,
- otherwise check Profile PA,
- otherwise create pure DP info = IDENT_NUMBER.

The data types described in this subclause are defined for the following namespace:

Namespace: profiident

12.2 Protocol specific handling of the data type STRING

Profibus char array rules:

- in all strings based on char ranges defined in the field bus protocol specification, the leading spaces are left trimmed. The char array is to be filled with 0x20h (blank);
- in VisibleStrings, invisible characters provided by a device have to be replaced by '?'.

12.3 Common device type identification data types

The data types described in this clause are reused as defined by 12.4 and 12.6 .

Table 22 lists the identification relevant data, source and format for Profibus DP.

Table 22 – Identification data types with Profibus DP specific mapping

Profibus - Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT data type (display format)	Specification reference
busProtocol	IdBusProtocol	For all DP devices : "protocol_DP"	Protocol	-	Enum	-
slaveAddress	IdAddress	Bus address is provided as part of live list by a PROFIBUS master. Service: FMA1/2_LIVE_LIST	Slave Address	Unsigned8	STRING (dec)	[5] Part 2, Section 4.2.3.6
identNumber	IdTypeID	IDENT_NUMBER can be requested by: DP Service DDLM_SLAVE_DIAG Allowed values are: Profile IDENT_NUMBER : 0x9700 (0x9700 to 0x9742) or Manufacturer specific IDENT_NUMBER	IDENT_NUMBER	Unsigned16	ARRAY of USINT (hex)	[5] Part 8, Section 9.3.1
manufacturerSpecificExtension		Can be used by DTM for a vendor specific device identification information, e.g. by combining a number of device parameter values into one string value. This can be used to identify a specific device variant			STRING	

Table 23 shows the mapping for PROFIBUS I&M data.

Table 23 – Identification data types with Profibus I&M specific mapping

Profibus-Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT format (display format)	Specification reference
identNumber, manufacturerSpecificExtension and slaveAddress as defined in DP table (Table 22).						
busProtocol	IdBusProtocol	For all I&M devices : "protocol_IM"	Protocol	-	Enum	-
manufacturerId	IdManufacturer	I&M 0 Element 1 Mapped to PB.DEVICE_MAN_ID	MANUFACTURER_ID	Unsigned16	UINT	[6] Section 3.2.2
orderId	-	I&M 0 Element 2 Mapped to PB.DEVICE_ID	ORDER_ID	20 Octets Visible String	STRING[20]	[6] Section 3.2.3
serialNumber	IdSerialNumber	I&M 0 Element 3 unique serial number Mapped to PB.DEVICE_SER_NUM	SERIAL_NUMBER	16 Octets VisibleString	STRING[16]	[6] Section 3.2.4
hardwareRevision	IdHardwareRevision	I&M 0 Element 4 HARDWARE_REVISION NOTE: If the device is a PA device,(PROFILE_ID is set to 0x9700), HARDWARE_REVISION shall be read from PA_I&M0 [6]	HARDWARE_REVISION	2 Octets - unsigned int 16 Converted to string to cover also: PA_I&M0 : VisibleString 16	STRING	[6] Section 3.2.5, Amendment I&M
softwareRevision	IdSoftwareRevision	I&M 0 Element 5 SOFTWARE_REVISION NOTE: If the device is a PA device (PROFILE_ID is set to 0x9700), SOFTWARE_REVISION shall be read from PA_I&M0 String format is used in FDT to harmonize version handling for all fieldbus protocols.	SOFTWARE_REVISION	4 Octets – 1 char + 3 unsigned 8, e.g.: V1.2.3 PA_I&M0 : VisibleString 16	STRING	[6] Section 3.2.6, Amendment I&M
profileID	-	I&M 0 Element 7	PROFILE_ID	2 Octets – unsigned 16	UINT	[6] Section 3.2.8

Profibus-Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT format (display format)	Specification reference
profileSpecificType	-	I&M 0 Element 8 MSB = BLOCK_OBJECT, LSB = PARENT_CLASS	PROFILE_SPECIFIC_TYPE	2 Octets – unsigned 16	UINT	[6] Section 3.2.9
imVersion	-	I&M 0 Element 9	IM_VERSION	2 Octets – unsigned 16; MSB major version xxx, LSB minor version yyy	FLOAT xxx.yyy	[6] Section 3.2.10
imSupported	-	I&M 0 Element 10	IM_SUPPORTED	2 Octets -	UINT	[6] Section 3.2.11
tagFunction	IdTag	I&M 1 Element 1 TAG_FUNCTION	TAG_FUNCTION	32 Octets – visible string	STRING[32]	[6] Section 3.2.12
tagLocation	-	I&M 1 Element 2 TAG_LOCATION	TAG_LOCATION	22 Octets – visible string	STRING[22]	[6] Section 3.2.13
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM ProfileDTM, BlockSpecificProfileDTM	DTM Support Level	-	Enum	

Table 24 lists the identification relevant data, source and format for Profibus PA.

Table 24 – Identification data types with Profibus PA specific mapping

Profibus-Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT format (display format)	Specification reference
identNumber, manufacturerSpecificExtension and slaveAddress as defined in DP table (Table 22).						
busProtocol	IdBusProtocol	For all PA devices : "protocol_PA"	Protocol	-	enum	-
manufacturerId	IdManufacturer	Physical Block - Index 10 (Unsigned16 – 2 decimal)	DEVICE_MAN_ID	Unsigned16	UINT	[2] Part 1, Section 3.11
device_id	-	Physical Block - Index 11 - VisibleString 16	DEVICE_ID	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
profile	-	Block structure of physical block – element 7 : OctetString - (Index 10 – size 2)	Profile	OctetString2	UINT	[2]0 Part 1, Section 3.7.2
profileRevision	IdBusProtocolVersion	Block structure of physical block – element 8 Unsigned 16 - (Index 6)	Profile Revision	Unsigned16	UINT	[2] Part 1, Section 3.7.2
softwareRevision	IdSoftwareRevision	Physical Block - Index 8 (VisibleString 16)	SOFTWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
hardwareRevision	IdHardwareRevision	Physical Block - Index 9 (VisibleString 16)	HARDWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
serialNumber	IdSerialNumber	Physical Block - Index 12 (VisibleString 16)	DEVICE_SER_NUM	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
deviceTAG	IdTag	BO of PB - OctetString 32	TAG_DESC	OctetString32	STRING[16]	
profileSpecificType	-	FDT attribute mapped to Block Object PARENT_CLASS sort of first TransducerBlock	BO PARENT_CLASS	Unsigned16	UINT	[2] Part 1, Section 3.7.2
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	DTM Support Level		enum	

Table 25 and Table 26 show identification data types with protocol independent semantics.

Table 25 – Simple identification data types with protocol independent semantics

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	enumeration - genericSupport - profileSupport - blockspecificProfileSupport - specificSupport - identSupport
match	STRING	Used by Device DTM to define a regular expression which shall match to scanned physical define identification information
nomatch	STRING	Used by Device DTM to define a regular expression which shall not match to scanned physical define identification information. Used by Device DTM to indicate if identification information may not match

Table 26 – Structured identification data types with protocol independent semantics

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
RegExpr	STRUCT			Includes regular expression string – either for match or for nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.4 Topology scan data types

These data types are used at Scan service response (Table 27 and Table 28).

Table 27 – Simple topology scan data types

Data type	Definition	Description
busAddress	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1

The data type ProfibusDevice describes one entry in the list of scanned PROFIBUS-Devices (see Table 28).

Table 28 – Structured topology scan data types

Data type	Definition			Description
	Elementary data type	Usage	Multiplicity	
ProfibusDevice	STRUCT			Specifies a Profibus device
	schemaVersion	O	[0..1]	
	busAddress	M	[1..1]	
	fdt:deviceTypeId	M	[1..1]	
	fdt:subDeviceType	O	[0..1]	

12.5 Scan identification data types

This subclause defines data types that are used to provide protocol specific scanning (see Table 29 and Table 30).

The data types described in this subclause are used in the following services:

- Scan service

Table 29 – Simple scan identification data types

Data type	Definition	Description
configuredState	enumeration (configuredAndPhysicallyAvailable configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	A communication master shall indicate in this attribute if the scan response is related to a detected physical device which is configured or unconfigured
resultState	enumeration (provisional final error)	Identifies if the result is one of the provisional results or the final result of the split scan results

Table 30 – Structured scan identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdAddress	STRUCT			All data types contain exactly one attribute each including the value of the scanned physical device. All data types with semantic meaning have a prefix “Id” for better identification
	profibusident:slaveAddress	M	[1..1]	
IdBusProtocol	STRUCT			
	profibusident:busProtocol	M	[1..1]	
IdBusProtocolVersion				
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	M	[1..1]	
IdManufacturer				
IdSerialNumber	STRUCT			These data types contain corresponding attributes defined in IEC 62453-2. They are transformed to name value pairs without semantic meaning for Frame Application
	profibusident:serialNumber	M	[1..1]	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	M	[1..1]	
IdTypeID	STRUCT			
	profibusident:identNumber	M	[1..1]	
Manufacturer_id	STRUCT			
	profibusident:manufacturer_id	M	[1..1]	
Order_id	STRUCT			
	profibusident:order_id	M	[1..1]	
ProfileID	STRUCT			
	profibusident:profileID	M	[1..1]	
ProfileSpecificType	STRUCT			
	profibusident:profileSpecificType	M	[1..1]	
ImVersion	STRUCT			
	profibusident:imVersion	M	[1..1]	
TagFunction	STRUCT			

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	profibusident:tagFunction	M	[1..1]	
TagLocation	STRUCT			
	profibusident:tagLocation	M	[1..1]	
ImSupported	STRUCT			
	profibusident:imSupported	M	[1..1]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:software↗RevisionP A	M	[1..1]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardware↗Revision PA	M	[1..1]	
ProfileRevisionPA	STRUCT			
	profibusident:profileRevisionPA	M	[1..1]	
Device_man_id	STRUCT			
	profibusident:device_man_id	M	[1..1]	
Device_id	STRUCT			
	profibusident:device_id	M	[1..1]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profileSpecific↗Type PA	M	[1..1]	
ManufacturerSpecific ↗Extension				These elements contain all elements for the appropriate protocol variant
	STRUCT			
	profibusident:manufacturer↗Speci ficExtension	M	[1..1]	
ScanIdentification_IM	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	ImVersion	M	[1..1]	
	TagFunction	M	[1..1]	
	TagLocation	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	ImSupported	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_PA	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	
	ProfileSpecificTypePA	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	IdTag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_DP	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentifications	STRUCT			Collection of ScanIdentification elements
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	choice of	M	[1..*]	
	ScanIdentification_DP	S	[0..*]	
	ScanIdentification_IM	S	[0..*]	
	ScanIdentification_PA	S	[0..*]	

12.6 Device type identification data types – provided by DTM

This subclause defines data types that are used to provide protocol specific information for device types (see Table 31).

The data types described in this clause are used in the following services:

- GetIdentificationInformation service

Table 31 – Structured device identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdBusProtocol	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix “Id” for better identification
	profibusident:busProtocol	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdBusProtocolVersion				
IdManufacturer				
IdTypeID	STRUCT			
	profibusident:identNumber	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Manufacturer_id	STRUCT			These elements contain corresponding attributes defined in IEC 62453-2. They are transformed to name value pairs without semantic meaning for Frame Application
	profibusident:manufacturer_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Order_id	STRUCT			
	profibusident:order_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileID	STRUCT			
	profibusident:profileID	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificType	STRUCT			
	profibusident:profileSpecificType	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImVersion	STRUCT			
	profibusident:imVersion	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImSupported	STRUCT			
	profibusident:imSupported	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:softwareRevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardwareRevisionPA	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	profibusident:RegExpr	O	[0..*]	
ProfileRevisionPA	STRUCT			
	profibusident:profile↗RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_man_id	STRUCT			
	profibusident:device_man_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_id	STRUCT			
	profibusident:device_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profile↗SpecificTypePA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ManufacturerSpecific↗Extension	STRUCT			
	profibusident:manufacturer↗SpecificExtension	M	[1..1]	
DeviceIdentification_IM	STRUCT			These elements contain all elements for the appropriate protocol variant
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	ImVersion	M	[1..1]	
	ImSupported	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification_PA	STRUCT			
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	ProfileSpecificTypePA	M	[1..1]	
	ManufacturerSpecific¬Extension	O	[0..1]	
DeviceIdentification_DP	STRUCT			
	profibusident:idDTM¬SupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdTypeID	M	[1..1]	
	ManufacturerSpecific¬Extension	O	[0..1]	
DeviceIdentification s	STRUCT			Collection of DeviceIdentification elements
	fdt:protocolId	M	[1..1]	
	choice of	M	[1..*]	
	DeviceIdentification_DP	S	[0..*]	
	DeviceIdentification_IM	S	[0..*]	
	DeviceIdentification_PA	S	[0..*]	

12.7 Identification information in GUI

It is mandatory to display I&M parameter labels in DTMs Presentation Object or Frame Application user interface from IM-Runtime.xml, in order to achieve consistency and acceptance for the user.

13 ProfiSafe

13.1 Motivation

The purpose of the ProfiSafe-UseCase paper is to compile a collection of typical existing failsafe automation examples from the discrete and continuous manufacturing areas and to derive a common systematic approach. This proposed solution ought to be based on

- Profibus DPV1 and ProfiSafe communication principles;
- new possibilities of combined failsafe controllers (standard and failsafe programs within one PLC);
- and current activities of other Profibus working groups like FDT/DTM and Proxy FB.

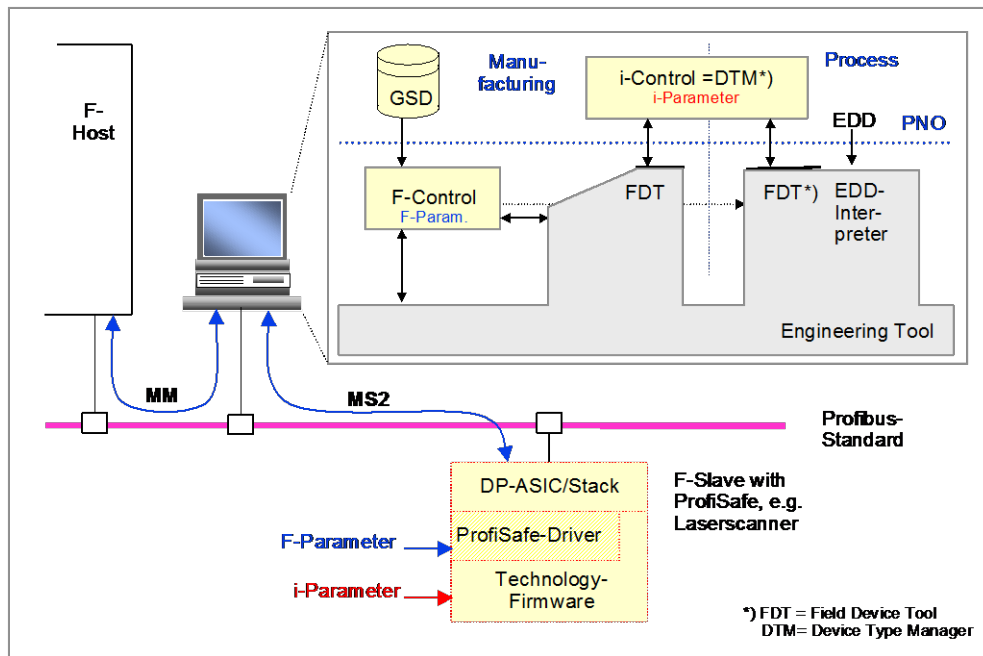
Within the context of the ProfiSafe-UseCase paper this PAS covers the partial aspect "FDT programmers interface for individual device parameter verification".

13.2 General parameter handling

In order to communicate in a safe way, each ProfiSafe device (F-Slave) requires so-called F-Parameters for the adjustment of the operational mode of its ProfiSafe driver. Those parameters comprise watch dog time, safety integrity level, container size, etc. In case of simple F-slaves no further parameters are required. F-Parameters will therefore be defined within GSD Revision 4.0.

In contrast, a complex F-slave requires additional individual and safety relevant parameters (individual device parameter) that, due to its size (>240 Bytes), often cannot be transported by the initial parameterization telegram (Prm-Telegram). As an amendment, ProfiSafe

guidelines are suggesting an additional method via proxy function blocks in a safety PLC, thus providing additional functionality like program-controlled reparameterization, etc.



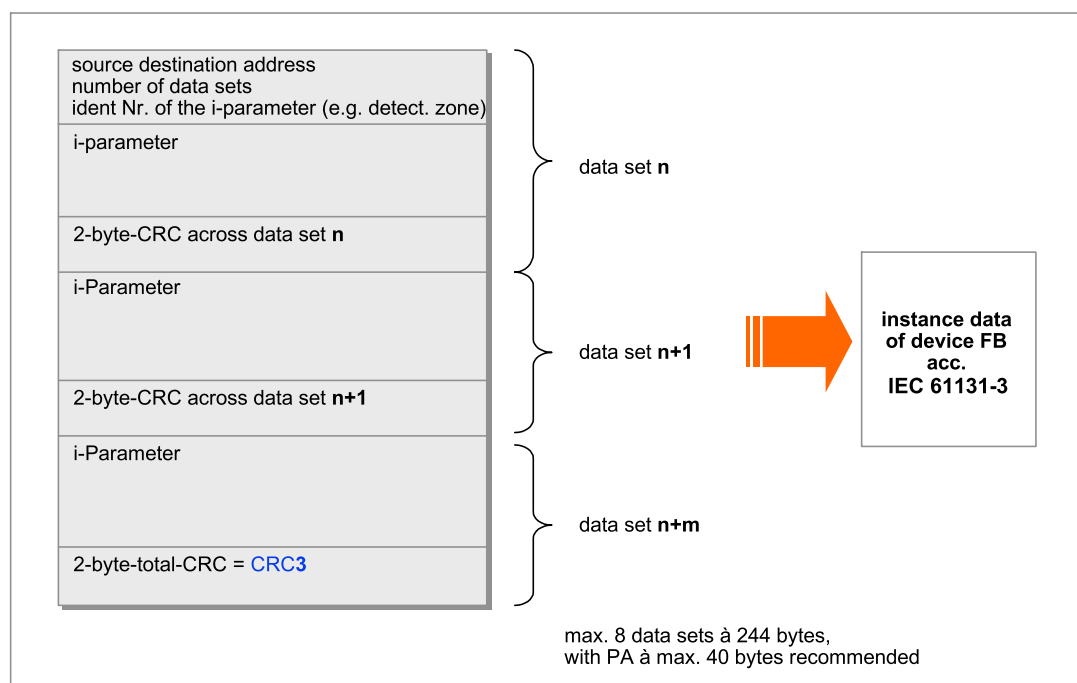
IEC 1130/09

Figure 4 – F-Parameter and individual device parameter

Individual device parameters, being F-device specific by nature, ought to be handled by Device-Type-Manager programs (DTM) coming from the F-device manufacturers. The engineering tool operates as a Frame Application for such a DTM of a F-slave. It is routing the communication requests of a DTM to its device (MS2) and provides data persistence. Otherwise, it is covering all traditional tasks like network configuration, parameterization, commissioning and diagnosis. It is communicating with the PLC via the Master-Master-Protocol (MM). In this respect, a PLC on the Profibus is a device of its own class (Master Class 1). The engineering tool itself is defined to be "a Master Class 2" device.

13.3 ProfiSafe individual device parameter

The structure of the ProfiSafe individual device parameter set is defined within the ProfiSafe guidelines. This set is wrapped up in XML-format for transportation across the FDT interface.



IEC 1131/09

Figure 5 – Data structure of ProfiSafe individual device parameters

The data structures and the data formats of the inner individual device parameters are proprietary. The whole set is divided into parts that fit into a Profibus data set ready for transportation. Each data set comprises a 2-bytes CRC for data consistency check. The polynomial is fixed and defined within the ProfiSafe guidelines. The CRC of the last set checks the total block.

Bibliography

- [1] PROFIBUS-PA Profile for Process Control Devices Version 3.0
 - [2] PROFIBUS Profile Guidelines, Part 1, Identification & Maintenance Functions, Version 1.1, May 2003
 - [3] PROFIBUS Guideline (Order No. 2.082): Chapter “Services on communication relationships MSAC_C2”: MSAC2_Initiate
 - [4] PROFIBUS Profile Amendment 3 to the Profibus Profile for Process Control Devices V3.0
 - [5] IEC 61158-6:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 6: Application layer protocol specification*
 - [6] PROFIBUS Specification Specification for PROFIBUS Device Description and Device Integration, Volume 1: GSD Version 5.04, July 2005
 - [7] IEC 61158-5:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 5: Application layer service specification*
 - [8] PROFIBUS Nutzerorganisation e.V., *PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998*
-

SOMMAIRE

AVANT-PROPOS.....	65
INTRODUCTION.....	67
1 Domaine d'application.....	69
2 Références normatives	69
3 Termes, définitions, symboles, abréviations et conventions	70
3.1 Termes et définitions	70
3.2 Symboles et abréviations	70
3.3 Conventions	71
3.3.1 Noms de type de données et références aux types de données.....	71
3.3.2 Vocabulaire pour les exigences	71
3.3.3 Utilisation de l'UML	71
4 Catégorie de bus	71
5 Accès aux données d'instances et de dispositifs.....	72
5.1 Objets "Process Channel" fournis par le DTM.....	72
5.2 Services DTM pour accéder aux données d'instances et de dispositifs.....	72
6 Comportement spécifique à un protocole.....	72
6.1 Modèle de dispositif PROFIBUS.....	72
6.2 Configuration et paramétrisation des dispositifs PROFIBUS.....	74
6.2.1 Généralités	74
6.2.2 DTM monolithique pour un dispositif PROFIBUS modulaire	74
6.2.3 DTM modulaire pour un dispositif PROFIBUS modulaire	75
6.3 Support pour la configuration DPV0	75
6.4 Esclaves PROFIBUS fonctionnant sans un maître PROFIBUS cyclique.....	75
6.5 Informations relatives à PROFIBUS d'un DTM esclave	76
6.5.1 Généralités	76
6.5.2 Bus Master Configuration Part (partie configuration principale de bus (BMCP))	76
7 Usage spécifique à un protocole des types de données généraux.....	88
8 Types de données communs spécifiques à un protocole	90
9 Types de données de gestion de réseau	90
9.1 Généralités.....	90
9.1.1 Configuration	91
9.1.2 Process Channel (voie de processus).....	91
9.1.3 Paramétrisation.....	92
9.2 Jeu de paramètres du bus maître	92
9.3 Jeu de paramètres du bus esclave	93
9.4 Données de module et de voie	94
9.5 Informations de GSD	97
9.5.1 Généralités	97
9.5.2 GSD pour dispositifs passerelles	97
10 Types de données de communication.....	98
10.1 Généralités.....	98
10.2 Informations d'erreur fournies par la voie de communication Communication Channel	98
10.3 Communication DPV0	98

10.4 Communication DPV1	105
11 Types de données paramètres de voie (Channel)	108
12 Identification de dispositif	112
12.1 Généralités.....	112
12.2 Traitement, spécifique à un protocole, du type de données STRING	112
12.3 Types de données d'identification de types de dispositif communs.....	112
12.4 Types de données de balayage de topologie	117
12.5 Types de données d'identification de balayage (Scan).....	118
12.6 Types de données d'identification de type de dispositif – fournis par DTM.....	121
12.7 Informations d'identification dans l'interface GUI	123
13 ProfiSafe	124
13.1 Motivation	124
13.2 Traitement général des paramètres	124
13.3 Paramètre de dispositif individuel ProfiSafe.....	126
Bibliographie	127
Figure 1 – Partie 303-1 de la série IEC 62453	68
Figure 2 – Modèle de dispositif PROFIBUS FDT	74
Figure 3 – Exemple pour données E/S dans des datagrammes	94
Figure 4 – F-Parameter (paramètre F) et paramètre de dispositif individuel	125
Figure 5 – Structure de données des paramètres de dispositifs individuels ProfiSafe.....	126
Tableau 1 – Identificateurs de protocole	71
Tableau 2 – Identificateurs de couche physique.....	71
Tableau 3 – BMCP Partie 1 – Configuration générale	78
Tableau 4 – BMCP Partie 2 – Données de paramètres.....	78
Tableau 5 – BMCP Partie 3 – Données de configuration	79
Tableau 6 – Partie 4: Table d'adresses et paramètres utilisateur d'esclave.....	80
Tableau 7 – Partie 4: Extended Prm data (Données de paramètres étendues).....	80
Tableau 8 – BMCP complète	81
Tableau 9 – Usage spécifique à un protocole des types de données généraux	88
Tableau 10 – Jeu de paramètres de bus pour dispositif maître	92
Tableau 11 – Jeu de paramètres de bus pour dispositif esclave	93
Tableau 12 – Voies de signal au sein de la trame de données	96
Tableau 13 – Types de données simples de communication DPV0.....	99
Tableau 14 – Types structurés de données de communication DPV0	100
Tableau 15 – Disponibilité des services pour la classe 1 de maître (C1).....	105
Tableau 16 – Disponibilité des services pour la classe 2 de maître (C2).....	105
Tableau 17 – Types de données simples de communication DPV1.....	106
Tableau 18 – Types de données structurés de communication DPV1	107
Tableau 19 – Mise en correspondance des types de données DPV1 aux types de données FDT	109
Tableau 20 – Types simples de données ChannelParameter pour les paramètres de voies	110

Tableau 21 – Types structurés de données ChannelParameter pour les paramètres de voies	111
Tableau 22 – Types de données d'identification avec mise en correspondance spécifique à Profibus DP	113
Tableau 23 – Types de données d'identification avec mise en correspondance spécifique à Profibus I&M	114
Tableau 24 – Types de données d'identification avec mise en correspondance spécifique à Profibus PA	116
Tableau 25 – Types de données d'identification simples avec sémantique indépendante vis-à-vis du protocole	117
Tableau 26 – Types de données d'identification structurés avec sémantique indépendante vis-à-vis du protocole	117
Tableau 27 – Types de données simples de balayage de topologie	117
Tableau 28 – Types de données structurés de balayage de topologie	118
Tableau 29 – Types de données simples d'identification de balayage	118
Tableau 30 – Types de données structurés d'identification de balayage	118
Tableau 31 – Types de données structurés d'identification de dispositif	121
Tableau 32 – Identificateurs de couche DataLink	72

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

SPÉCIFICATION DES INTERFACES DES OUTILS DES DISPOSITIFS DE TERRAIN (FDT) –

Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1 et CP 3/2

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

DÉGAGEMENT DE RESPONSABILITÉ

Cette version consolidée n'est pas une Norme IEC officielle, elle a été préparée par commodité pour l'utilisateur. Seules les versions courantes de cette norme et de son(ses) amendement(s) doivent être considérées comme les documents officiels.

Cette version consolidée de l'IEC 62453-303-1 porte le numéro d'édition 1.1. Elle comprend la première édition (2009-06) [documents 65E/127/FDIS et 65E/140/RVD] et son amendement 1 (2016-06) [documents 65E/336/CDV et 65E/395A/RVC]. Le contenu technique est identique à celui de l'édition de base et à son amendement.

Dans cette version Redline, une ligne verticale dans la marge indique où le contenu technique est modifié par l'amendement 1. Les ajouts sont en vert, les suppressions sont en rouge, barrées. Une version Finale avec toutes les modifications acceptées est disponible dans cette publication.

La Norme internationale IEC 62453-303-1 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Chaque partie de la série IEC 62453-3xy est destinée à être lue conjointement à l'IEC 62453-2.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62453, sous le titre général *Spécification des interfaces des outils des dispositifs de terrain (FDT)*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de la publication de base et de son amendement ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "<i>colour inside</i>" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

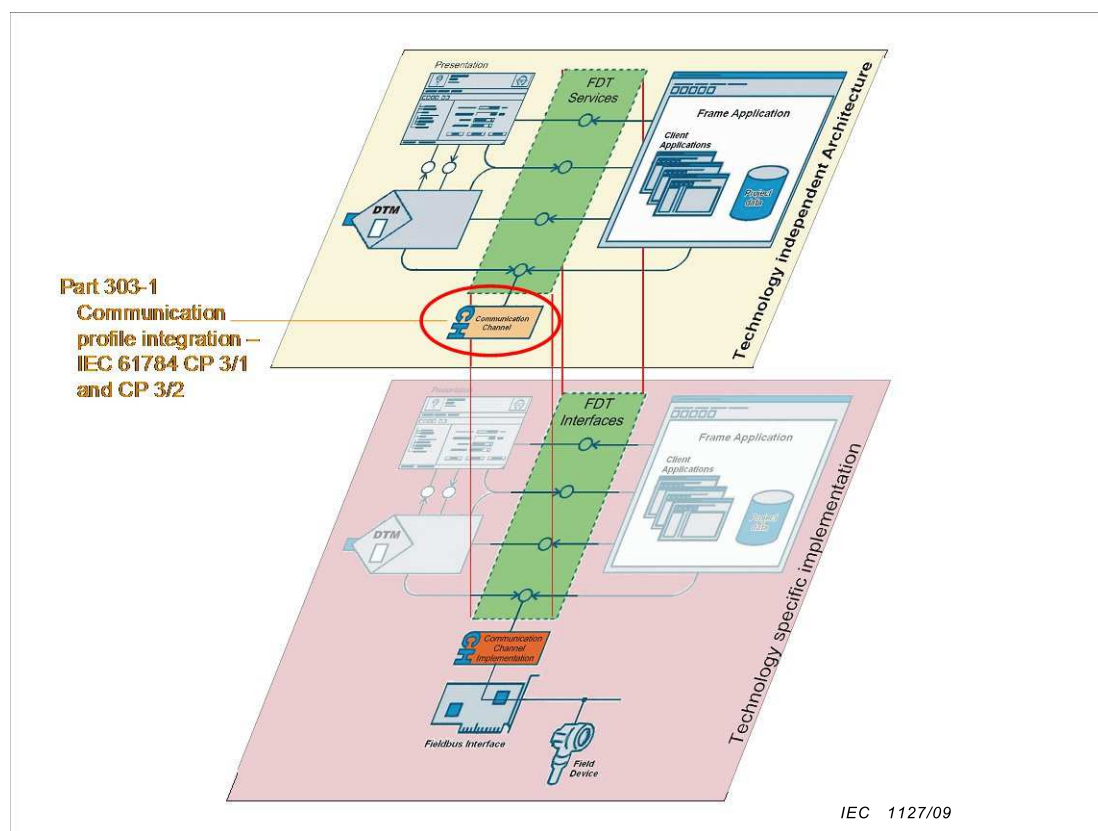
INTRODUCTION

La présente partie de l'IEC 62453 est une spécification d'interface à l'attention des développeurs de composants FDT (Field Device Tool) pour le contrôle de fonctions et l'accès aux données au sein d'une architecture client/serveur. La spécification résulte d'un processus d'analyse et de conception pour développer des interfaces normalisées et faciliter le développement de serveurs et de clients par plusieurs fournisseurs qui ont besoin d'interfonctionner sans couture.

Avec l'intégration des bus de terrain dans des systèmes de commande, il existe quelques autres tâches qu'il est nécessaire d'accomplir. Outre les outils spécifiques à un bus de terrain et à un dispositif, il existe la nécessité d'intégrer ces outils dans des outils d'études ou de planification de plus haut niveau à l'échelle d'un système. En particulier, pour l'utilisation dans des systèmes de commande étendus et hétérogènes, généralement dans le secteur de l'industrie de transformation, la définition sans ambiguïté d'interfaces d'étude qui soient faciles à utiliser pour tous ceux qui sont impliqués a une grande importance.

Un composant logiciel spécifique à un dispositif, appelé DTM (Device Type Manager, «gestionnaire de type de dispositif»), est livré par le fabricant de dispositif de terrain avec son dispositif. Le DTM est intégré dans des outils d'études par l'intermédiaire des interfaces FDT définies dans la présente spécification. L'approche à l'intégration est en général ouverte pour toutes les sortes de bus de terrain et, donc, satisfait aux exigences pour intégrer des types différents de dispositifs dans des systèmes de commande hétérogènes.

La Figure 1 montre comment l'IEC 62453-303-1 est alignée dans la structure de la série IEC 62453.



Légende

Anglais	Français
Part 303-1 Communication profile integration – IEC 61784 CP 3/1 and CP 3/2	Part 303-1 Intégration des profils de communication – IEC 61784 CP 3/1 et CP 3/2
Presentation	Présentation
Technology independent architecture	Architecture indépendante vis-à-vis de toute technologie
Technology specific implementation	Mise en œuvre spécifique à une technologie
Frame application	Application cadre
Client application	Application Client
Communication channel	Voie de communication
FDT Service	Services de FDT
FDT Interfaces	Interfaces des FDT
Fieldbus interface	Interface Fieldbus

Figure 1 – Partie 303-1 de la série IEC 62453

SPÉCIFICATION DES INTERFACES DES OUTILS DES DISPOSITIFS DE TERRAIN (FDT) –

Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1 et CP 3/2

1 Domaine d'application

Le Profil de communication 3/1 et le Profil de communication 3/2 (communément appelés PROFIBUS™¹) définissent des profils de communication basés sur l'IEC 61158-2 Type 3, l'IEC 61158-3-3, l'IEC 61158-4-3, l'IEC 61158-5-3 et l'IEC 61158-6-3. Les profils de base CP 3/1 (PROFIBUS DP) et CP 3/2 (PROFIBUS PA) sont définis dans l'IEC 61784-1.

La présente partie de la IEC 62453 donne des informations pour intégrer le protocole PROFIBUS dans la spécification d'interface FDT (IEC 62453-2).

La présente partie de l'IEC 62453 spécifie les services de communication et autres services.

La présente spécification ne contient la spécification FDT ni ne la modifie.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61131-3:2003, *Automates programmables – Partie 3: Langages de programmation*

IEC 61158 (toutes les parties), *Réseaux de communications industriels – Spécifications des bus de terrain*

IEC 61158-2 :2014, *Réseaux de communications industriels – Spécifications des bus de terrain – Partie 2: Spécification et définition des services de couche physique*

IEC 61158-3-3, *Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3 *Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-5-3: *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements*

IEC 61158-6-3, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

¹ PROFIBUS™ est un nom commercial de l'organisation sans but lucratif PROFIBUS Nutzerorganisation e.V. (PNO). Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. La conformité à la présente norme n'exige pas l'utilisation des logos enregistrés pour PROFIBUS™. L'utilisation des logos enregistrés pour PROFIBUS™ exige la permission de PNO.

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

3 Termes, définitions, symboles, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 62453-1 et l'IEC 62453-2 s'appliquent.

3.1.1

module d'interface de bus

module d'un dispositif de terrain qui fournit la connexion au bus de terrain

3.1.2

CP 3/1

profil de communication de CPF3, caractérisé par une émission asynchrone; RS 485 (ANSI TIA/EIA RS-485-A); RS 485-IS en option; fibre plastique; fibre de verre multimode ou fibre de verre monomode; fibre PCF

3.1.3

CP 3/2

profil de communication de CPF3, caractérisé par une émission synchrone; codage Manchester et alimentation par bus (manchester coded and bus powered (MBP)); sécurité intrinsèque (MBP-IS) et faible puissance (MBP-LP) en option

3.2 Symboles et abréviations

Pour les besoins du présent document, les symboles et abréviations donnés dans l'IEC 62453-1 et l'IEC 62453-2 ainsi que les suivants s'appliquent.

ANSI	American National Standards Institute (Institut national américain des normes (http://www.ansi.org))
BIM	Bus Interface Module (Module d'interface de bus)
BMCP	Bus Master Configuration Part (Partie configuration principale de bus)
CFG	Données de configuration utilisées au cours de l'initialisation de l'équipement esclave PROFIBUS
DCS	Distributed Control System (Système de commande distribué)
DP	Decentralized Peripherals (Périphériques décentralisés)
EIA	Electronic Industries Alliance (Alliance des industries électroniques)
FDL	Fieldbus Data Link layer (Couche liaison de données des bus de terrain)
FMA	Fieldbus Management layer (Couche gestion des bus de terrain)
FMS	Fieldbus Message Specification (Spécification de message des bus de terrain)
GSD	General Station Description (Description de station générale)
MBP	Manchester coded Bus Powered (Alimenté par bus et à codage Manchester)
I&M	Identification and maintenance functions (Fonctions d'identification et maintenance)
PA	Process Automation (Automatisation de processus)
PCF	Polymer Clad Fibre (Fibre à gaine polymère)
PROFIBUS	Process Field Bus (Bus de terrain de processus)
RS	Radio Sector (Secteur radio)/ Recommended Standard (Norme recommandée)

3.3 Conventions

3.3.1 Noms de type de données et références aux types de données

Les conventions de dénomination et de référencement des types de données sont expliquées dans l'IEC 62453-2, Article A.1

3.3.2 Vocabulaire pour les exigences

Les expressions suivantes sont utilisées pour spécifier des exigences.

Utilisation de "doit" ou "obligatoire"	Aucune exception tolérée.
Utilisation de "il convient de" ou "recommandé"	Forte recommandation. Il peut être légitime, dans des cas particuliers exceptionnels, de s'écarter du comportement décrit.
Utilisation de "conditionnel"	La fonction ou le comportement doit être donné(e), selon des conditions définies.
Utilisation de "peut" ou "facultatif"	La fonction ou le comportement peut être donné(e), selon des conditions définies.

3.3.3 Utilisation de l'UML

Les figures dans le présent document utilise la notation UML telle que définie dans l'Annexe A de l'IEC 62453-1.

4 Catégorie de bus

Les protocoles CP 3/1 et CP 3/2 sont identifiés dans l'élément protocolld du type de données structuré 'fdt:BusCategory' par les identificateurs uniques suivants (Tableau 1):

Tableau 1 – Identificateurs de protocole

Valeur d'identificateur	Nom de Protocolld	Description
036D1497-387B-11D4-86E1-00E0987270B9	'Profibus DP/V0'	Prise en charge du protocole Profibus DP V0
036D1499-387B-11D4-86E1-00E0987270B9	'Profibus DP/V1'	Prise en charge du protocole Profibus DP V1

Les protocoles CP 3/1 et CP 3/2 utilisent les identificateurs uniques suivants dans les membres physicalLayer au sein du type de données PhysicalLayer (Tableau 2):

Tableau 2 – Identificateurs de couche physique

Valeur d'identificateur	Nom	Description
036D1590-387B-11D4-86E1-00E0987270B9	MBP	IEC 61158-2 (MBP, Profibus PA)
036D1591-387B-11D4-86E1-00E0987270B9	RS485	IEC 61158-2:2014, Article 22 (RS485, PROFIBUS DP)
036D1592-387B-11D4-86E1-00E0987270B9	Fiber Optic (Fibres optiques)	IEC 61158-2:2014, Article 23 (Câble à fibres optiques, PROFIBUS DP)
036D1593-387B-11D4-86E1-00E0987270B9		Ethernet (déconseillé, ne pas utiliser)

Le Tableau 32 définit quelle DataLinkLayer doit être utilisée en combinaison avec les valeurs de BusCategory définies dans le Tableau 32.

Tableau 32 – Identificateurs de couche DataLink

Valeur d'identificateur	Nom	Description
50A21B35-7EE7-4999-8174-70396929C0B4	PROFIBUS FDL	PROFIBUS FDL
CDF338DC-E9A3-4D13-91AC-60A43DCB2904	PROFIBUS FMA1/2	PROFIBUS FMA1/2

5 Accès aux données d'instances et de dispositifs

5.1 Objets "Process Channel" fournis par le DTM

Il convient que l'ensemble minimal de données fournies soit: les valeurs de processus modélisées comme des objets ~~channel~~ (de voies), y compris les plages et les changements d'échelle.

5.2 Services DTM pour accéder aux données d'instances et de dispositifs

Le services InstanceDataInformation et DeviceDataInformation doivent donner accès à au moins tous les paramètres du Physical Block et le statut et la valeur Out des blocs fonctionnels doivent être exposés.

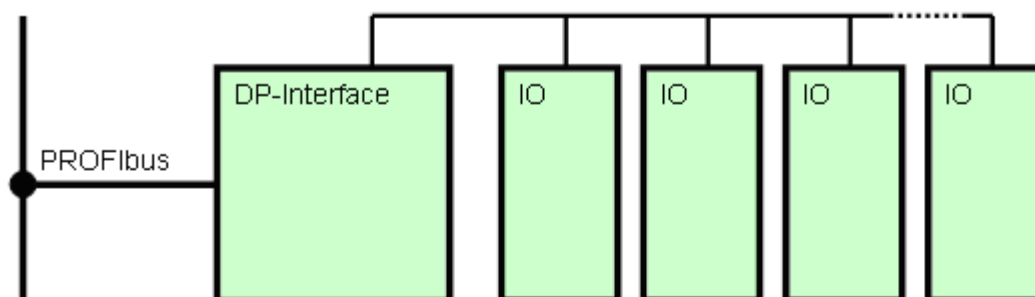
Conformément à l'IEC 62453-2, au moins un jeu d'informations sémantiques (un par protocole de bus de terrain pris en charge) doit être fourni pour chaque objet de données accessibles, en utilisant le type de données général 'SemanticInformation'. Le type de données 'applicationDomain' correspondant doit avoir une valeur définie pour Profibus et le type de données 'semanticId' doit avoir une valeur appropriée, telle que spécifiée dans le Tableau 9.

6 Comportement spécifique à un protocole

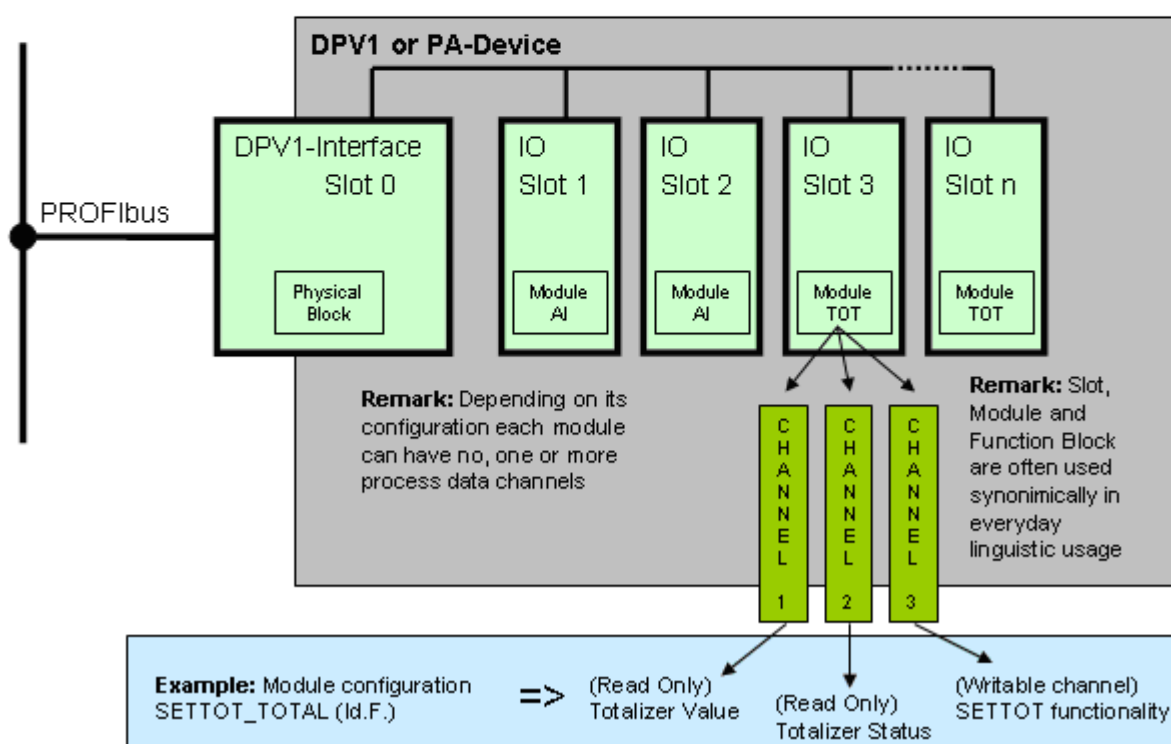
6.1 Modèle de dispositif PROFIBUS

FDT étend le modèle de dispositif PROFIBUS en utilisant des Process Channel (voies de processus) pour la description des valeurs E/S (voir Figure 2).

Classical View of PROFIBUS device



PROFIBUS notations from a monolithic DPV1 or PA device DTMs point of view



IEC 1128/09

Légende

Anglais	Français
Classical View of PROFIBUS device	Vue classique d'un dispositif PROFIBUS
PROFIBus	PROFIBus
DP-Interface	Interface DP
PROFIBUS notation from a monolithic DPV1 or PA device DTMs point of view	Notation PROFIBUS, du point de vue des DTM de dispositif DPV1 ou PA monolithiques
DPV1 or PA device	Dispositif DPV1 ou PA
IO Slot 0	Baie 0 E/S
Physical block	Bloc physique
Module	Module
Remark: Depending on its configuration each module can have no, one or more process data channels.	Remarque: Selon sa configuration, chaque module peut avoir zéro, une ou plusieurs voies de données de processus.
Channel	Voie
Remark: Slot, Module and Function Block are often used synonymically in everyday linguistic usage	Remarque: Slot («baie»), Module et Fonction («Bloc fonctionnel») sont souvent utilisés comme synonymes dans le langage de tous les jours.

Anglais	Français
Example: Module configuration	Exemple: Configuration de module
(Read Only) Totalizer Value	(Lecture seule) Valeur de totaliseur
(Read Only) Totalizer Status	(Lecture seule) Statut de totaliseur
(Writable Channel) SETTOT functionality	(Voie inscriptible) Fonctionnalité SETTOT
IO	E/S (entrée/sortie)

Figure 2 – Modèle de dispositif PROFIBUS FDT

6.2 Configuration et paramétrisation des dispositifs PROFIBUS

6.2.1 Généralités

Dans un outil de configuration GSD, l'utilisateur définit la configuration et établit les paramètres appropriés pour les modules. L'outil de configuration crée la chaîne de configuration et la chaîne de paramètres qui sont utilisées pour établir correctement l'esclave.

Avec FDT, la configuration et la paramétrisation des dispositifs ne sont plus exécutées seulement par un composant central; elles se sont partiellement déplacées dans les DTM. Un DTM est chargé de donner des informations de configuration et de paramétrisation pour le maître cyclique qui met les esclaves PROFIBUS en fonctionnement.

Un DTM est utilisé pour réajuster un dispositif de terrain à son application spécifique. Au sein de PROFIBUS, il y a trois différents aspects du réajustement:

- paramétrisation: usr prm data (utilisée dans le service SET_PRM de PROFIBUS pour établir la communication cyclique et le comportement spécifique du dispositif);
- paramétrisation d'application: paramètres spécifiques à une application (émis au moyen des services read/write (lecture/écriture) acycliques de PROFIBUS);
- configuration: données de configuration (utilisées dans le service CHK_CFG de PROFIBUS pour la définition du format et de la longueur des données d'entrée/sortie qui sont émises dans le cadre de la communication cyclique).

La paramétrisation d'application émise par la communication acyclique ne relève pas du domaine d'application du présent document. Dans le présent document, le terme "paramétrisation" représente la paramétrisation des communications (SET_PRM).

6.2.2 DTM monolithique pour un dispositif PROFIBUS modulaire

Un DTM monolithique est un DTM seul qui représente le dispositif complet avec son module d'interface de bus (BIM pour «Bus Interface Module») et ses modules E/S (d'entrée/sortie). En général, un tel DTM propose un dialogue de configuration (objet "presentation") qui permet la définition des BIM et modules utilisés. Le dialogue de configuration peut être disponible par le biais de la fonction FDT normalisée "Configure" (voir [1] 4.3 Configuration de fonctionnement).

Ce ne sont pas tous les dispositifs PROFIBUS qui exigent un dialogue de configuration. C'est pourquoi tous les DTM ne fournissent pas nécessairement la fonction "Configure". Cela ne vaut que pour les dispositifs PROFIBUS non modulaires si les Usr_Prm-Data ne peuvent pas être modifiées.

Le dialogue de configuration permet de changer les données seulement en mode hors ligne si le jeu de données peut être verrouillé.

6.2.3 DTM modulaire pour un dispositif PROFIBUS modulaire

Des DTM distincts représentent le BIM (DTM de dispositif) et les modules E/S particuliers (DTM de modules). L'effort pour développer un tel DTM modulaire est normalement plus grand que dans le cas d'un DTM monolithique, car:

- un protocole privé doit être mis en œuvre entre le BIM et les modules E/S pour assurer que seul un DTM de module peut être ajouté au DTM de BIM. Cela nécessite un ID de protocole propre et l'adaptation/création d'une communication;
- dans certains cas, des interfaces privées complémentaires sont nécessaires pour échanger des informations entre le DTM de dispositif pour BIM et les DTM de modules.

La mise en œuvre d'un DTM modulaire procure les avantages suivants:

- le projet représente la structure des dispositifs;
- l'utilisateur est à même d'accéder à des informations relatives au module, et ce, directement en fonction du DTM de module ;
- la spécification FDT définit un mécanisme pour identifier les DTM. Avec ces mécanismes, il est possible d'assurer la prise en charge pour le balayage des modules en dessous du BIM et de générer automatiquement la topologie;
- la prise en charge d'un nouveau type de BIM ou de module E/S exige un DTM supplémentaire "seulement" et n'affecte pas les composants existants. Cela peut se traduire par un effort d'essais réduit.

Les données de configuration pour établir la configuration PROFIBUS doivent être fournies par le DTM de dispositif (représentant le BIM). Ces données de configuration peuvent être générées à partir d'informations de DTM enfants instanciés et en utilisant un dialogue de configuration.

Il convient que des DTM modulaires soient fournis pour des dispositifs modulaires (par exemple: un opérateur d'installation peut ajouter/retirer des modules). Les DTM monolithiques sont utilisés pour représenter des dispositifs qui ne présentent aucune modularité (les dispositifs PA, par exemple).

6.3 Support pour la configuration DPV0

Un esclave PROFIBUS est configuré par un maître cyclique et communique par PROFIBUS DP. En plus de cela, l'esclave peut prendre en charge la communication DPV1.

Un DTM de passerelle pour un esclave PROFIBUS n'est pas tenu de fournir la communication pour le schéma de communication DPV0. Par exemple, il existe un système E/S distant avec des modules HART. Il peut avoir un DTM de passerelle qui nécessite le protocole DPV1 et fournit le protocole HART (défini dans le document d'informations et dans le document de paramètres). Cela permet aux DTM de dispositifs HART de communiquer avec leurs dispositifs par le truchement du DTM de passerelle et par le DTM de communication pour DPV1. Respectant la spécification, le DTM de passerelle délivre les documents de paramètres de voies pour les protocoles DPV1 et HART à la fois. Le protocoleId est un membre du type de données NetworkManagementInfo.

Les voies de processus (Process Channels) doivent fournir les documents de Paramètres de Voies (ChannelParameter) pour DPV1, y compris toutes les informations permettant l'intégration dans le système de commande (par exemple: DPAAddress de la valeur E/S, si disponible).

6.4 Esclaves PROFIBUS fonctionnant sans un maître PROFIBUS cyclique

Dans la plupart des cas, un esclave PROFIBUS est configuré et paramétré par un dispositif maître PROFIBUS cyclique. Donc, un dispositif maître en fonctionnement dans le réseau est requis.

Certains esclaves sont capables de permettre une communication acyclique sans un maître cyclique en fonctionnement. Notamment dans le cas de la fonctionnalité de passerelle, ceci constitue un insigne avantage car ils permettent la paramétrisation de dispositifs de terrain qui leur sont connectés en utilisant un maître bus acyclique. Ainsi, les spécialistes des instruments peuvent également travailler avec des dispositifs de terrain même si le dispositif de commande ne fonctionne pas encore.

Si un maître lance la communication, ces dispositifs commencent à détecter la vitesse et les réglages de bus pour réagir de façon appropriée. Cela peut prendre un certain temps.

Ci-après, il est décrit deux cas qu'un utilisateur peut garder à l'esprit lorsqu'il travaille avec de tels dispositifs.

Cas d'utilisation 1:

L'utilisateur accomplit un balayage de réseau. Le DTM de communication tente de lire des données de diagnostic au moyen d'une demande GetDiagnose mais ne reçoit pas de réponse. Le dispositif n'est pas détecté par le DTM de communication. Cela se produit surtout lorsque le dispositif a une basse adresse PROFIBUS. La raison en est que le dispositif n'a pas achevé la détection de la vitesse du bus/ des réglages du bus comme cela a été demandé pour les données de diagnostic. La solution est de donner à ces dispositifs une adresse PROFIBUS plus haute.

Cas d'utilisation 2:

L'utilisateur tente de connecter un dispositif de terrain relié à la passerelle qui prend en charge DPV1 sans un maître cyclique en fonctionnement. Cela peut conduire à un message d'erreur car le dispositif passerelle n'a pas achevé la détection de la vitesse du bus/des réglages du bus comme cela a été demandé pour une connexion. L'utilisateur doit donc tenter la connexion de nouveau. Cela se produit seulement dans de très rares cas.

6.5 Informations relatives à PROFIBUS d'un DTM esclave

6.5.1 Généralités

Les informations utilisées par un dispositif maître cyclique pour établir correctement un réseau PROFIBUS et permettre une communication cyclique entre le système de commande et les dispositifs esclaves sont fournies par un DTM dans

- la partie configuration principale de bus (BMCP);
- les informations GSD;
- la topologie interne ;
- les voies de processus.

Un DTM d'un esclave PROFIBUS doit délivrer ces parties d'informations relatives à PROFIBUS pour être intégré dans un système d'ingénierie FDT. Dans les prochains paragraphes, il est donné une description plus détaillée de la manière de générer et de la manière de fournir ces informations. Cela dépend de la sorte de DTM (voir 6.2 Configuration et paramétrisation des dispositifs PROFIBUS).

6.5.2 Bus Master Configuration Part (partie configuration principale de bus (BMCP))

6.5.2.1 Introduction à la BMCP

La BMCP d'une seule instance de DTM prise séparément décrit les données réelles de paramètres et de configuration de l'esclave PROFIBUS correspondant. Chaque DTM représentant un dispositif d'esclave PROFIBUS doit fournir une Bus Master Configuration Part. La BMCP est fournie dans le membre busMasterConfigurationPart du type de données

NetworkManagementInfo. Ces informations sont obtenues en appelant le service NetworkManagementInfoRead.

La BMCP inclut des informations relatives à la configuration et aux paramètres pour l'esclave. La BMCP est fournie par le DTM et est indispensable pour générer la configuration de maître.

La BMCP contient des données susceptibles de changer durant la configuration de maître. Cela signifie que la BMCP peut être modifiée et retournée au DTM d'esclave en appelant NetworkManagementInfoWrite. Un DTM d'esclave doit accepter les nouvelles informations et recalculer les paramètres de configuration/internes pour s'accorder avec la nouvelle BMCP.

DTM doit vérifier si, oui ou non, les nouvelles valeurs sont conformes aux capacités du dispositif. L'appel de NetworkManagementInfoWrite sera refusé si le dispositif ne peut pas traiter les nouvelles valeurs.

6.5.2.2 Création de la BMCP

Le présent paragraphe explique dans le détail la signification des éléments individuels de la BMCP.

La BMCP peut être créée à partir des informations GSD d'un dispositif PROFIBUS.

La BMCP est divisée en quatre parties qui sont expliquées dans les paragraphes suivants.

Les explications utilisent les mots clés de GSD et font référence à la spécification PROFIBUS. Voir aussi le tableau avec la BMCP complète en 6.5.2.2.1.

6.5.2.2.1 Partie 1: De Slave_Para_Len à l'Octet 15

La première partie consiste en un jeu fixe d'octets décrits dans le tableau ci-dessous.

Tableau 3 – BMCP Partie 1 – Configuration générale

Octet	Nom	Défini dans	Remarque
0	Slave_Para_Len	[5] 6.2.12.1	Longueur de la BMCP, y compris cette valeur
1			
2	SI_Flag	[5] 6.2.12.2	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0: réservé, • Bit 1: Extra_Alarm_SAP, • Bit 2: DPV1_Data_Types, • Bit 3: DPV1_Supported, • Bit 4: Publisher_Enable, • Bit 5: Fail_Safe. Les bits suivants ne sont pas basés sur des valeurs GSD: • Bit 6: New_Prm, • Bit 7: Active
3	Slave Type	[5] 6.2.12.3	La valeur est 0 (= DP-Slave, c'est-à-dire esclave DP)
4	Max_Diag_Data_Len	[5] 6.2.12.4	La valeur GSD suivante est utilisée [6]: Max_Diag_Data_Len
5	Max_Alarm_Len	[5] 6.2.12.5	
6	Max_Channel_Data_Len	[5] 6.2.12.6	Ce champ définit la quantité de données pouvant être transférées entre esclave et maître. Dans ce cas, le maximum de ces valeurs GSD [6] doit être calculé: • Max_Data_Len, • C1_Max_Data_Len plus 4 octets (Function Num, Slot_Number, Length)
7	Diag_Update_Delay	[5] 6.2.12.7	La valeur GSD suivante est utilisée [6]:Diag_Update_Delay
8	Alarm_Mode	[5] 6.2.12.8	
9	Add_SI_Flag	[5] 6.2.12.9	
10	C1_Timeout	[5] 6.2.12.10	La valeur GSD suivante est utilisée [6]:C1_Response_Timeout
11			
12	Reserved		
13			
14			
15			

6.5.2.2.2 Partie 2: Données de paramètres

La Partie 2 de la BMCP est constituée des données de paramètres d'esclave.

Tableau 4 – BMCP Partie 2 – Données de paramètres

Octet	Nom	Défini dans	Remarque
16	Prm_Data_Len	[5] 6.2.12.11	Longueur des Données de paramètres, y compris cette valeur
17			
18	Station Status	[5] 6.2.4.1	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0 à 2: réservés, • Bit 3: WD_On, • Bit 4: Freeze_Req, • Bit 5: Freeze_Req, • Bit 6, 7(Lock/Unlock Request c'est-à-dire Demande de verrouillage/déverrouillage): [5] Tableau 402
19	WatchDog1	[5] 6.2.4.2	Ces valeurs (WD_Fact_1 et WD_Fact_2) dépendent du débit en bauds. Il convient qu'un maître établisse ces valeurs et que des esclaves traitent de nouvelles valeurs.
20	WatchDog2	[5] 6.2.4.3	
21	Min Tsdr	[5] 6.2.4.4	La valeur par défaut est un temps de 11 bits [1]

Octet	Nom	Défini dans	Remarque
22	Ident_Number	[5] 6.2.3.5	La valeur GSD suivante est utilisée [6]: Ident_Number
23			
24	Group_Ident	[5] 6.2.4.5	Indique l'affectation de groupe de l'esclave sous une forme codée au niveau du bit
25	DPV1_Status_1	[5] 6.2.4.7	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0, 1: réservés, • Bit 2: WD_Base_1ms, • Bit 3 à 5: réservés, • Bit 6: Fail_Safe, • Bit 7: DPV1_Enable, Ces bits rendent compte des capacités des esclaves au maître et sont modifiés par le maître MS0 suivant ses capacités.
26	DPV1_Status_2	[5] 6.2.4.8	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0: Check_Cfg_Mode, • Bit 1: réservé, • Bit 2: Enable_Update_Alarm, • Bit 3: Enable_Status_Alarm, • Bit 4: Enable_Manufacturer_Specific_Alarm, • Bit 5: Enable_Diagnostic_Alarm, • Bit 6: Enable_Process_Alarm, • Bit 7: Enable_Pull_Plug_Alarm, Ces bits rendent compte des capacités des esclaves au maître et sont modifiés par le maître MS0 suivant ses capacités.
27	DPV1_Status_3	[5] 6.2.4.9	• Bit 0 à 2: - , • Bit 3: Prm Structure, • Bit 4: IsoM_Req, • Bit 5-6: réservés, • Bit 7: Prm_Cmd
28	User_Prm_Data		À partir de l'Octet 28 de la BMCP, un DTM doit insérer des données de paramètres utilisateur supplémentaires. Les informations sont données dans le fichier GSD au moyen de la valeur de User_Prm_Data
...			
NOTE 1 Il est possible qu'un module n'ait pas de données de paramètres utilisateur. Dans ce cas, aucune chaîne de paramètres n'est insérée dans la BMCP pour ce module.			
NOTE 2 Certains esclaves ont un certain nombre de modules fixes en plus de la BIM. Même si ces modules n'apparaissent pas dans le dialogue de configuration ou ne sont pas représentés comme des DTM de modules, ils doivent être pris en considération dans la BMCP s'ils ont des paramètres.			
NOTE 3 Les trois octets de statut DPV1 sont définis par le maître MS0.			

6.5.2.2.3 Partie 3: Données de configuration

Les données de configuration sont fournies comme partie 3:

Tableau 5 – BMCP Partie 3 – Données de configuration

Octet	Nom	Source	Remarque
16+Prm_Data_Len	Cfg_Data_Len	[5] 6.2.12.13	Longueur, y compris cette valeur
16+Prm_Data_Len + 1			
16+Prm_Data_Len +2	Cfg_Data		Données de configuration (si disponibles)
...			

À partir de la position d'octets (16+Prm_Data_Len +2), les chaînes de configuration pour le BIM et les modules sont données dans l'ordre croissant. Les informations sont données dans le fichier GSD par le truchement du module de valeur.

NOTE 1 Certains esclaves ont un certain nombre de modules fixes en plus de la BIM. Même si ces modules n'apparaissent pas dans le dialogue de configuration ou ne sont pas représentés comme des DTM de modules, ils doivent être pris en considération dans la BMCP s'ils ont des paramètres.

NOTE 2 Des baies vides doivent être prises en considération dans les données de configuration. Se référer au fichier GSD pour savoir quelle chaîne de configuration est à utiliser.

6.5.2.2.4 Partie 4: Table d'adresses et paramètres utilisateur d'esclave

Dans la Partie 4 de la BMCP, il est fourni la section table d'adresses et paramètres utilisateur d'esclave.

Tableau 6 – Partie 4: Table d'adresses et paramètres utilisateur d'esclave

Octet	Nom	Source	Conseils
16 + Prm_Data_Len + Cfg_Data_Len	Add_Tab_Len	[5] 6.2.12.15	Longueur, y compris cette valeur
16 + Prm_Data_Len + Cfg_Data_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len	Slave_User_Data_Len	[6] 6.2.12.21	Longueur, y compris cette valeur
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 2	Slave_User_Data		Paramètre utilisateur d'esclave (si disponible)
...			

Après Add_Tab_Len, les données de la table d'adresses sont insérées dans la BMCP. Si le maître ne prend pas en charge cette table, le paramètre Add_Tab_Len est mis à 2 et aucune donnée n'est insérée.

Après Slave_UserDataLen, les données utilisateur d'esclave sont insérées dans la BMCP. Si le maître ne prend pas en charge ces données, le paramètre Slave_User_Data_Len est mis à 2 et aucune donnée n'est insérée.

Tableau 7 – Partie 4: Extended Prm data (Données de paramètres étendues)

Octet	Nom	Source	Conseils
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len	Ext_Prm_Data_Len	[5] 6.2.12.23	Longueur, y compris cette valeur
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len + 1			

Après Ext_Prm_Data_Len, les données de paramètres étendus sont insérées dans la BMCP. Si le maître ne prend pas en charge ces données, le paramètre Ext_Prm_Data_Len est mis à 2 et aucune donnée n'est insérée.

6.5.2.2.5 Jeu complet de paramètres de bus esclave

Toutes les valeurs sont fournies à partir du DTM d'esclave. Il est de la responsabilité du DTM d'esclave d'être compatible avec la GSD d'esclave. Le DTM de maître pourrait changer certaines de ces valeurs initiales envoyées par le DTM d'esclave si elles dépendent des capacités du maître.

Exemple

Au sein du fichier GSD, il est énoncé que le dispositif prend en charge le mode Freeze ("gel") par le mot-clé "Freeze_Mode_supp". Le maître établit la valeur "Freeze_Mode_Req" au sein du jeu de paramètres de bus esclave ("Slave Bus Parameter Set"), car seul le maître sait s'il prend en charge ce mode ou non.

Le tableau suivant explique quel composant est la source des valeurs de paramètres ("Paramètres fournis par"). Certaines de ces valeurs peuvent être modifiées par le système ou par interaction avec l'utilisateur. Pour ces valeurs, il est indiqué quel composant est autorisé à changer les valeurs ("Configuration par"). Si possible, les valeurs par défaut pour les paramètres sont définies ("Valeur par défaut").

Tableau 8 – BMCP complète

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
0	Slave_Para_Len	Longueur de la BMCP, y compris cette valeur	Calculé par le DTM d'esclave	-	-
1					
2	SI_Flag	Bit codé. La signification des bits:	-	-	-
		Bit 0: réservé	-	-	0
		Bit 1: Extra_Alarm_SAP	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Extra_Alarm_SAP_sup p au sein de GSD
		Bit 2: DPV1_Data_Types	Le DTM d'esclave montre quels types de données sont utilisés par le DTM d'esclave.	DTM de maître	DPV1_Data_Types au sien de la GSD d'esclave
		Bit 3: DPV1_Supported	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	DPV1_Slave au sein de la GSD
		Bit 4: Publisher_enable	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Publisher_supp au sein de GSD
		Bit 5: Fail_Safe	Le DTM d'esclave peut montrer par ce fanion que l'esclave a besoin de sécurité intégrée (Fail_Safe ou Fail_Safe_Required de la GSD)	DTM de maître	Valeur au sein de la GSD d'esclave
		Bit 6: New_Prm	Si le DTM d'esclave établit cette valeur, il indique qu'il convient que le maître MS0 établisse le paramètre d'esclave modifié. (voir [5] 6.2.12.2)	-	0
		Bit 7: Active	Si le DTM esclave a fini la configuration de module, le mettre à 1.	-	0
3	Slave Type	0 = Esclave DP	L'esclave de DTM montre qu'il s'agit d'un esclave DP.	-	0
4	Max_Diag_Data_Len		L'esclave DTM fournit ces informations par le paramètre Max_Diag_Data_Len dans la GSD	-	Valeur au sein de la GSD d'esclave
5	Max_Alarm_Len	Longueur de la structure d'alarmes (voir [5] Tableau 396)	DTM d'esclaves, conditions: Un des mots-clés de GSD:- Diagnostic_Alarm_supp - Process_Alarm_supp - Pull_Plug_Alarm_supp - Status_Alarm_supp - Updata_Alarm_supp - Manufacturer_Specific_Alarm_supp OR - Alarm_Type_Mode_supp	-	Est calculée sur la base des différentes valeurs GSD.
6	Max_Channel_Data_Len		DTM d'esclave: Ce champ définit la quantité de données pouvant être transférées entre esclave et maître. Dans ce cas, le maximum de ces valeurs doit être calculé et établi par le DTM d'esclave. Règle pour le calcul: Max_Data_Len C1_Max_Data_Len plus 4 octets (Function Num, Slot_Number, Length)	-	Valeur au sein de la GSD d'esclave
7	Diag_Update_Delay		DTM d'esclave:	-	GSD d'esclave [6] Diag_Update_Delay

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
8	Alarm_Mode		DTM d'esclave:	-	GSD d'esclave [6] Alarm_Sequence_Mode_Count
9	Add_SI_Flag	Bit codé. Suivant la signification des bits:			
		Bit 0: NA_To_Abort		DTM de maître	0
		Bit 1: Ignore_Aclr		DTM de maître	0
		Bit 2 à Bit 7: réservés			0
10	C1_Timeout	La valeur GSD suivante est utilisée [6]: C1_Response_Timeout	DTM d'esclave:		GSD d'esclave [6] C1_Response_Timeout
11					
12	Réservé				0
13	Réservé				0
14	Réservé				0
15	Réservé				0
16	Prm_Data_Len	Longueur des Données de paramètres, y compris cette valeur	Calculé (ou peut-être fixé) par le DTM d'esclave		
17					
	PRM_DATA				
18	Prm_Data (Station Status)	Bits 0 à 2: réservés		0	
		Bit 3: WD_On	Le maître MS0 définit que le chien de garde (watchdog) est utilisé ou non. Si le chien de garde est activé, le maître doit aussi établir WD_Fact_1 et WD_Fact_2	DTM de maître	0
		Bit 4: Freeze_Req	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Freeze_Mode_supp au sein de GSD
		Bit 5: Sync_Req	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	(Sync_Mode_supp au sein de GSD)
		Bits 6, 7: Demande de verrouillage/déverrouillage [5] Tableau 402	DTM de maître	0	

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
19	Prm_Data (WatchDog1) WD_Fact_1	Ces valeurs dépendent du débit en bauds. Il convient qu'un maître établisse ces valeurs et que des esclaves traitent de nouvelles valeurs. Prendre en considération au sein de Usr_Prm_Data (DPV1_Status_1, Bit 2 WD_Base_1ms)		DTM de maître	1
20	Prm_Data (WatchDog2) WD_Fact_2	Voir Octet 19, Watchdog_Time = 10 ms * WD_Fact_1 * WD_Fact_2 OR Watchdog_Time = 1 ms * WD_Fact_1 * WD_Fact_2		DTM de maître	1
21	Prm_Data (Min Tsdr)	La valeur par défaut est un temps de 11 bits [1].		DTM de maître	11
22	Prm_Data (Ident_Number)		SlaveDTM	-	GSD d'esclave [6] Ident_Number
24	Prm_Data (Group_Ident)	Cela signifie que l'esclave n'est pas affecté à un groupe pour commande globale. Il est établi par le maître		DTM de maître	0
25	Prm_Data (DPV1_Status_1)	Bits 0, 1: réservés			0
		Bit 2: WD_Base_1ms (Voir Octets 19 et 20)	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître Dans quelques rares cas, il est nécessaire de mettre la valeur à 1.	WD_Base_1ms_supp au sein de GSD
		Bits 3 à 5: réservés			0
		Bit 6: Fail_Safe	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	(Fail_safe au sein de GSD)

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
		Bit 7: DPV1 Enable	DTM d'esclave:	DTM de maître	GSD d'esclave (Si le minimum de GSD comprend l'une des valeurs établies suivantes: C1_Read_Write_supp = 1 ou Diagnostic_Alarm_supp = 1 ou Process_Alarm_supp = 1 ou Pull_Plug_Alarm_supp = 1 ou Status_Alarm_supp = 1 ou Update_Alarm_supp = 1 ou Manufacturer_Specific_Alarm_supp = 1. L'esclave prend en charge DPV1 et il convient alors que DPV1_Enable soit 1

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
26	Prm_Data (DPV1_Status_2)	Bit 0: Check_Cfg_Mode			
	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Check_Cfg_Mode au sein de GSD		
		Bit 1: réservé	-	-	0
		Bit 2: Enable_Update_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Update_Alarm_supp au sein de GSD
		Bit 3: Enable_Status_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Status_Alarm_supp au sein de GSD
		Bit 4: Enable_Manufacturer_Specific_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Manufacturer_Specific_Alarm_supp au sein de GSD
		Bit 5: Enable_Diagnostic_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Diagnostic_Alarm_supp au sein de GSD
		Bit 6: Enable_Process_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Process_Alarm_supp au sein de GSD
		Bit 7: Enable_Pull_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Pull_Plug_Alarm_supp au sein de Slave GSD
27	Prm_Data (DPV1_Status_3)	Bit 0-2: Alarm_Mode	DTM d'esclave:	DTM de maître	Données au sein de la GSD d'esclave
		Bit 3: Prm_Structure.	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.		Prm_Block_Structure_supp au sein de GSD
		Bit 4: IsoM_Req	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Isochron_Mode_supp au sein de GSD
		Bit 5-6: réservé	-	-	0
		Bit 7: Prm_Cmd	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	PrmCmd_supp au sein de GSD
28 à ...	Prm_Data(Usr_Prm_Data)		Calculé par le DTM d'esclave	DTM d'esclave:	Données au sein de GSD
X1	Cfg_Data_Len	Longueur, y compris cette valeur	Calculé par le DTM d'esclave	-	Données au sein de GSD
X1+1					
X1+2 à ...	Cfg_Data	Cfg_Data	DTM d'esclave, Dépend de la configuration des modules	DTM d'esclave:	Données au sein de la GSD d'esclave (Module, EndModule)
X2	Add_Tab_Len	Longueur, y compris cette valeur	Calculée par le DTM de communication		

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
X2 + 1					
X2 + 2 à ...	Add_Tab_Data	Table d'affectation d'adresses (seulement pour les maîtres DPV0)	Calculée par le DTM de communication		
X3	Slave_User_Data_Len	Longueur, y compris cette valeur	Calculée par le DTM de communication		
X3 + 1					
X3 + 2 à ...	Slave_User_Data	Slave_User_Data	Calculée par le DTM de communication		Données au sein de GSD
X4 + 1	Ext_Prm_Data_Len	Longueur, y compris cette valeur	Calculé par le DTM d'esclave	-	
X4 + 2					
X4	Ext_Prm_Data	Données de paramètres étendues	DTM d'esclave:	-	Données au sein de GSD
NOTE X1 = 16+Prm_Data_Len X2 = 16 + Prm_Data_Len + Cfg_Data_Len = 16 + x1 + Cfg_Data_Len X3 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len = X2 + Add_Tab_Len X4 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len = X3 + Slave_User_Data_Len					

6.5.2.3 Modification de la BMCP

6.5.2.3.1 Propagation des modifications

La BMCP inclut des données de paramètres et de configuration. Le DTM d'esclave ou la FDT Frame Application ("application-cadre FDT") peut modifier la BMCP.

Le système doit assurer que le DTM de communication représentant le maître PROFIBUS est conscient de ces modifications. Cela est accompli en envoyant l'événement InstanceDataChanged depuis le DTM esclave afin de rapporter le changement de la BMCP. Il convient que tous les changements soient rapportés dès que possible, mais pas avant que les changements ne soient persistants. La Frame Application informe le DTM parent, par OnChildInstanceDataChanged, que des paramètres d'un enfant ont été modifiés. Le DTM de communication peut alors récupérer la nouvelle BMCP du DTM d'esclave.

6.5.2.3.2 Quand il est autorisé à modifier la BMCP

Conformément à la spécification, il est autorisé de changer les paramètres d'un DTM en commençant à partir de l'état "configuré" (voir IEC 62453-2).

La BMCP peut être modifiée plusieurs fois, mais seulement si le DTM est en mode hors ligne et si le jeu de données peut être verrouillé.

Si un DTM d'esclave souhaite modifier des paramètres en mode en ligne, il doit utiliser les demandes de transaction DPV0 ou DPV1. S'il n'y a aucun moyen de changer les paramètres par des demandes de transaction, le DTM doit désactiver la configuration et la paramétrisation dans l'état en ligne.

6.5.2.3.3 Données de paramètres

Si l'utilisateur modifie des paramètres du BIM ou d'un module (Par exemple par l'intermédiaire de l'interface utilisateur) et cela affecte la BMCP, le DTM doit mettre à jour la BMCP. En plus de cela, il doit demander une sauvegarde et informer l'application-cadre par le biais de l'InstanceDataChanged. La Frame Application doit distribuer les informations à tous les composants pertinents.

6.5.2.3.4 Données de configuration

Les données de configuration changeront chaque fois que l'utilisateur ajoute/retire des modules ou change le type de module, etc.

Dans le cas d'un DTM modulaire, le BIM sera informé lorsque l'utilisateur ajoute ou retire des modules au moyen du service ChildAdded et du service ChildRemoved. Les changements des paramètres dans un module seront rapportés par le service OnChildInstanceDataChanged.

Le DTM monolithique ou le DTM de BIM met à jour sa BMCP, demande une sauvegarde et informe la Frame Application par le biais de l'InstanceDataChanged.

NOTE 1 Les changements qui affectent la BMCP prennent souvent effet sur la topologie interne et les voies de processus (Process Channel). Il faut que ces informations soient mises à jour par les DTM aussi.

NOTE 2 La BMCP peut être modifiée par le DTM d'esclave et par le DTM de communication. Il ne faut pas que le DTM de communication change les données de configuration et les paramètres utilisateur.

6.5.2.4 Cas particuliers relatifs à la BMCP

6.5.2.4.1 Prise en charge de DPV1

Dans le fichier GSD, il y a deux fanions concernant DPV1. En premier lieu, la valeur "DPV1_Slave": Cela signifie que l'esclave a la possibilité de fonctionner comme esclave

DPV1. Si la valeur existe et la valeur est "1", le bit respectif dans le SI_Flag du Slave Bus Parameter Set doit être mis. Pour des systèmes plus anciens, il convient que la possibilité de fonctionner comme esclave DPV0 existe.

Seul le DTM de communication sait que son dispositif maître est capable de fournir des services acycliques.

Après avoir bâti le Slave Bus Parameter Set, le DTM de communication recevra la BMCP initiale de l'esclave. Si, dans le SI_Flag, le bit respectif est mis, le DTM de maître doit mettre le bit de poids fort dans le premier octet de Extended DPV1 Status. Maintenant, l'esclave fonctionne comme esclave DPV1.

6.5.2.4.2 Extended DPV1 status

Tous les bits dans les trois octets Extended DPV1 Status sont mis par le DTM de communication pour le maître. Un DTM esclave doit accepter ces valeurs de réglage et adapter sa fonctionnalité si besoin est.

7 Usage spécifique à un protocole des types de données généraux

La table ci-après (Tableau 9) explique comment des types de données, définis dans l'IEC 62453–2 au sein de l'espace de noms 'fdt', sont utilisés avec les protocoles CP 3/1 et CP 3/2.

Tableau 9 – Usage spécifique à un protocole des types de données généraux

Type de données	Description à utiliser dans l'IEC 61784 CPF 3
fdt:address	Adresse de paramètre Profibus: L'attribut 'address' suit les différents modèles de dispositifs qui sont définis pour les dispositifs PROFIBUS. FDT prend actuellement en charge les modèles suivants: PROFIBUS DP / DPV1, PROFIBUS PA, PROFIdrive (version supérieure ou égale à la version de profil 3) PROFIBUS DP / DPV1 Le modèle de dispositif est basé sur des dispositifs qui sont constitués de baies, attendu que les baies ne représentent pas des objets physiques. Les données qui sont contenues dans les baies sont adressables par de indices. Ces données peuvent être variables ou des blocs composés de données. L'attribut d'adresses est APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz sont basés sur le format décimal sans zéro ("0") de tête. PROFIBUS PA Le dispositif est représenté par une structure de gestion de dispositif et un nombre de blocs qui fournissent une fonctionnalité différente (bloc physique, bloc fonctionnel, bloc transducteur). Les blocs sont mis en correspondance avec des adresses de baies, mais cette mise en correspondance peut varier en fonction du type de dispositif. L'attribut d'adresses est APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz sont basés sur le format décimal sans zéro ("0") de tête. PROFIdrive Conformément au profil PROFIdrive, un dispositif (unité d'entraînement) peut être composé d'un certain nombre (1 à plusieurs) objets d'entraînement (DO

Type de données	Description à utiliser dans l'IEC 61784 CPF 3
	pour "Drive objects"). Les DO peuvent avoir des types différents. Chaque DO est identifiable de façon univoque et gère ses propres paramètres. Chaque paramètre peut être identifié de façon univoque par son numéro (PNU). Chaque DO a son propre espace de numéros. Un paramètre peut contenir des données simples ou des données composées (des matrices, par exemple). Les données du dispositif sont accessibles par une voie de paramètres (normalement baie 0 indice 47). L'attribut d'adresse est APIxxSLOTyyINDEXzz.Dodo-id.pnu xx API yy Slot zz Index do-id Drive Object ID pnu ParameterNumber xx, yy, zz, do-id, pnu sont basés sur le format décimal sans zéro ("0") de tête
fdt:protocolId	Voir Article 4
fdt:deviceTypeId	L'élément "fdt:deviceTypeId" doit contenir l'IDENT_NUMBER du dispositif physique pris en charge. L'IDENT_NUMBER doit être saisi au format décimal; toutefois, il convient que la valeur soit affichée sous forme hexadécimale pour l'utilisateur.
fdt:deviceTypeInfo	Un dispositif PROFIBUS doit fournir ses informations de GSD sous la forme d'une chaîne lisible par l'homme en cet attribut. NOTE Les informations de GSD sont accessibles au moyen des services NetworkManagementInfoRead GetTypeInfo
fdt:deviceTypeInfoPath	Chemin conduisant au fichier contenant les informations qui sont fournies par l'intermédiaire de l'attribut 'deviceTypeInfo'. Dans le cas de PROFIBUS, l'attribut contient le chemin complet du fichier GSD, y compris le nom du fichier. Le nom de fichier dépend du paramètre de lieu actuel conformément à l'utilisation du service SetLanguage. Pour les dispositifs PROFIBUS, il est obligatoire de fournir cet attribut. Exemples Anglais: 'C:\MyFolder\ABCD.GSE' Allemand: 'C:\MyFolder\ABCD.GSG'
fdt:manufacturerId	Fabricant selon la spécification de profils. Par exemple, dans Profibus PA: Physical Block Index 10: DEVICE_MAN_ID
fdt:semanticId fdt:applicationDomain	Les SemanticID pour PROFIBUS suivent les différents modèles de dispositifs qui sont définis pour les dispositifs PROFIBUS. FDT prend actuellement en charge les modèles suivants: PROFIBUS DP, PROFIBUS PA, PROFIdrive. PROFIBUS PA L' applicationDomain est: FDT_PROFIBUS_PA Le dispositif est représenté par une structure de gestion de dispositif et un nombre de blocs qui fournissent une fonctionnalité différente (bloc physique, bloc fonctionnel, bloc transducteur). Les blocs sont mis en correspondance avec des adresses de baies, mais cette mise en correspondance peut varier en fonction du type de dispositif. Le modèle de dispositifs étant basé sur des blocs, les SemanticIds sont également basés sur le modèle de blocs. Au sein de chaque bloc, les données sont identifiables par des noms de paramètres. Le semanticId pour le paramètre relatif au profil PROFIBUS respecte les règles suivantes: - le semanticId doit être construit sur la base des noms définis dans les profils; - les paramètres structurés doivent être combinés avec un '.'; - les espaces dans la définition du profil doivent être remplacés par un trait de soulignement ("underscore");

Type de données	Description à utiliser dans l'IEC 61784 CPF 3
	- les blocs doivent être comptés conformément au dictionnaire d'objets (Object Dictionary); - le numéro de bloc doit être partie intégrante du semanticID. Le semanticId est: BlockType.BlockIndex.NameOfParameter.AttributeOfParameter Exemple AnalogInputFB.3.OUT.Unit PROFIdrive L' applicationDomain est: FDT_PROFIBUS_PROFIDRIVE Conformément au profil PROFIdrive, un dispositif (unité d'entraînement) peut être composé d'un certain nombre (1 à plusieurs) objets d'entraînement (DO pour "Drive objects"). Les DO peuvent avoir des types différents. Chaque DO est identifiable de façon univoque et gère ses propres paramètres. Chaque paramètre peut être identifié de façon univoque par son numéro (PNU). Chaque DO a son propre espace de numéros. Un paramètre peut contenir des données simples ou des données composées (des matrices, par exemple). Les données du dispositif sont accessibles par une voie de paramètres (normalement baie 0 indice 47). Le semanticId est: DODO-id.PNUpnu do-id Drive Object ID pnu ParameterNumber do-id, pnu sont basés sur le format décimal sans zéro ("0") de tête. Exemple DO3.PNU64 PROFIBUS DPV1, L' applicationDomain est: FDT_PROFIBUS_DPV1 Le modèle de dispositif est basé sur des dispositifs qui sont constitués de baies, attendu que les baies ne représentent pas des objets physiques. Les données qui sont contenues dans les baies sont adressables par des indices. Ces données peuvent être variables ou des blocs composés de données. Le semanticID pour des dispositifs qui n'est pas basé sur un profil est directement basé sur les informations d'adresse PROFIBUS: Le semanticId est: APIxx.SLOTyy.INDEXzz xx API yy Slot zz Index xx, yy, zz sont basés sur le format décimal sans zéro ("0") de tête.
fdt:subDeviceType	Valeur spécifique à un fabricant

8 Types de données communs spécifiques à un protocole

Non applicable.

9 Types de données de gestion de réseau

9.1 Généralités

Les types de données spécifiés dans le présent paragraphe sont utilisés aux services suivants:

- service NetworkManagementInfoRead;
- service NetworkManagementInfoWrite.

Le type de données `net:DeviceAddress` (défini dans l'IEC 62453-2) est utilisé pour définir l'adresse réseau d'un dispositif.

Les types de données spécifiques à un protocole sont basés sur des définitions données dans les spécifications IEC 61784 et IEC 61158. En outre, celles-ci contiennent des informations complémentaires concernant le dispositif dont les systèmes ont besoin pour configurer les liaisons CPF 3 et établir la communication entre le dispositif maître CPF 3 et les dispositifs esclaves CPF 3.

9.1.1 Configuration

La configuration du dispositif en soi est effectuée avec l'aide de l'interface graphique utilisateur (GUI) du DTM. Le téléchargement de la configuration vers le dispositif esclave est accompli par l'intermédiaire du dispositif maître CPF 3. Pour ce faire et afin d'établir la communication de bus, le maître a besoin d'informations de la part du DTM comme elles existent:

- **Fichier GSD**
Les informations de GSD sont des informations spécifiques à un type et pas spécifiques à une instance. Elles ne sont pas stockées avec des instances d'esclaves individuelles ou dans un fichier accessible au niveau global. Elles sont fournies par le DTM au niveau du service `GetTypeInformation`. Sur ce service, un DTM d'un dispositif PROFIBUS fournit les informations de GSD au sein de son document XML.
Le dispositif maître peut utiliser les informations générales spécifiques à un type issues des informations de GSD de l'esclave comme les paramètres de temporisations de bus, les débits en bauds pris en charge, etc.
- **CFG-String (Cfg_Data)**
La CFG-String fournit des informations spécifiques à une instance relatives à la configuration courante du dispositif. Elle définit la structure des trames de données qui seront émises sur le PROFIBUS. Cette structure dépend des modules qui sont réellement configurés.
Le DTM fournit la CFG-String au sein de l'attribut `busMasterConfigurationPart` qui est partie intégrante du document XML disponible par le biais du service `InstanceDataRead`. La structure de l'attribut `busMasterConfigurationPart` est définie conformément au PROFIBUS-DP-Slave-Bus-Parameter-Set (voir l'IEC 62453-2 et aussi la série IEC 61158).

Le dispositif maître utilise ces informations pour établir la communication avec le dispositif esclave.

9.1.2 Process Channel (voie de processus)

Dans le cas des protocoles CPF 3, une FDT-Channel (voie FDT) est une représentation pour une date simple ou une valeur de processus qui peut être atteinte à partir de la Frame Application par le biais du dispositif maître. Les informations disponibles en des services pour des informations relatives aux entrées-sorties (E/S) décrivent comment accéder à une voie par une commande PROFIBUS DPV1 ou comment adresser une voie avec une trame PROFIBUS DP pour E/S cycliques. Outre tous les éléments obligatoires (qui incluent `id` et `dpAddress`), il est fortement recommandé que des informations d'adresse DPV1 soient fournies. Ces informations (DPV1 Slot) sont utilisées par certaines trames pour gérer les informations de modules de dispositifs PROFIBUS.

Dans un environnement DPV0, en fonction de la situation, le dispositif maître sous-jacent peut avoir soit la fonctionnalité de maître de classe 1, soit la fonctionnalité de maître de classe 2. Un maître de classe 1 peut écrire des données de sortie dans un dispositif et commander l'échange de données, là où un maître de classe 2 peut seulement lire les données de sortie. En général, il est présumé que la paramétrisation telle que décrite ici est accomplie en tant que station de maître de classe 2.

9.1.3 Paramétrisation

Il existe deux options pour écrire des paramètres établis à partir de la GUI du DTM dans le dispositif esclave CPF 3 sur le terrain:

- Paramètres utilisateur
Les User Parameters (paramètres utilisateur) sont partie intégrante du PROFIBUS-DP-Slave-Bus-Parameter-Set. Ils contiennent certaines données spécifiques à un fabricant pour caractériser l'esclave DP. Le DTM écrit les paramètres utilisateur dans la busMasterConfigurationPart. Les paramètres utilisateur sont stockés avec le dispositif maître pendant la configuration du maître PROFIBUS et sont automatiquement envoyés à l'esclave au cours de l'établissement de la communication de bus. (Ceci est spécifique à PROFIBUS; pour les détails, voir la série IEC 61158.) Lors du changement des paramètres utilisateur pendant le temps d'exécution, le DTM doit utiliser une connexion DP V0 et les commandes DP V0 appropriées pour l'échange de paramètres tel que décrit dans les types de données.
- Écriture de paramètres avec les services DP-V1 (services MSAC2)

Le DTM peut utiliser des services de transport DP-V1 pour envoyer ses paramètres vers le dispositif esclave. Pour cela, il doit utiliser une connexion DP-V1 et les commandes de communication correspondantes. Au cours de l'établissement de la communication, les services DP-V1 ne sont pas envoyés automatiquement. La Frame Application ou un DTM doit invoquer le téléchargement de paramètres par l'intermédiaire de DP-V1.

Pour les détails sur le comportement différent des esclaves en fonction du type de paramétrisation, se référer à l'IEC 61158.

Les connexions et les commandes de communication DP-V1 peuvent également être utilisées pour exécuter des commandes au niveau de l'esclave. Pour les détails sur l'utilisation de DP-V1, voir aussi la série IEC 61158.

9.2 Jeu de paramètres du bus maître

Le jeu de paramètres suivants représente le contenu du type de données busMasterConfigurationPart au sein des données d'instances de dispositifs pour le dispositif maître de Profibus (Tableau 10). Cet attribut doit être mis pour chaque dispositif maître de Profibus conforme à la série IEC 61158. Pour plus de détails, se référer au diagramme de séquence "Configuration d'un maître Fieldbus" et à la série IEC 61158.

Tableau 10 – Jeu de paramètres de bus pour dispositif maître

Nom	Type	Commentaire
Bus_Para_Len	UINT	Longueur du jeu de paramètres de bus inclusive, le paramètre de longueur (page 34 à 216-1)
FDL_Add	USINT	Partie obligatoire du jeu de paramètres de bus selon la Spécification DP PROFIBUS; non utilisée dans le présent contexte, car l'adresse est transférée dans une variable distincte; FDL-Add peut différer de l'adresse de station réelle utilisée par EE et les DTM
Baud_rate	USINT	Numéro de code pour le débit en bauds
TSL	UINT	Durée d'intervalle de temps
min TSDR	UINT	Réponse de retard de station minimal
max TSDR	UINT	Réponse de retard de station maximal
TQUI	USINT	Temps de cessation
TSET	USINT	Temps d'établissement
TTR	UDINT	Temps de rotation cible
G	USINT	Facteur mise à jour GAP
HSA	USINT	Plus haute adresse de station

Nom	Type	Commentaire
max_retry_limit	USINT	Limite maximale des répétitions de tentative
Bp_Flag	USINT	Fanions pour l'interface utilisateur, par exemple fanion d'action erronée
Min_Slave_Interval	UINT	Plus petite durée permise entre deux cycles de sondage d'esclaves
Poll_Timeout	UINT	Temporisation maître-maître
Data_Control_Time	UINT	Durée garantie entre deux mises à jour de Data_transfer_list
Octet 1 (réservé)	ARRAY[6] of USINT	
...		
Octet 6 (réservé)		
Master_User_Data_Len	UINT	Longueur de Master_User_Data, y compris le paramètre de longueur
Master_Class2_Name	ARRAY (32) of CHAR	Nom du maître de classe 2 DP avec lequel le jeu de paramètre a été créé
Master_User_Data	ARRAY OF USINT	Spécifique à un fabricant

9.3 Jeu de paramètres du bus esclave

Le jeu de paramètres suivants représente le contenu du type de données busMasterConfigurationPart au sein du jeu de données d'instances pour les dispositifs esclaves de Profibus (Tableau 11). Cet attribut doit être mis pour chaque dispositif esclave de Profibus conforme à la série IEC 61158. Pour plus de détails, se référer au diagramme de séquence "Configuration d'un maître Fieldbus" et à la série IEC 61158.

Tableau 11 – Jeu de paramètres de bus pour dispositif esclave

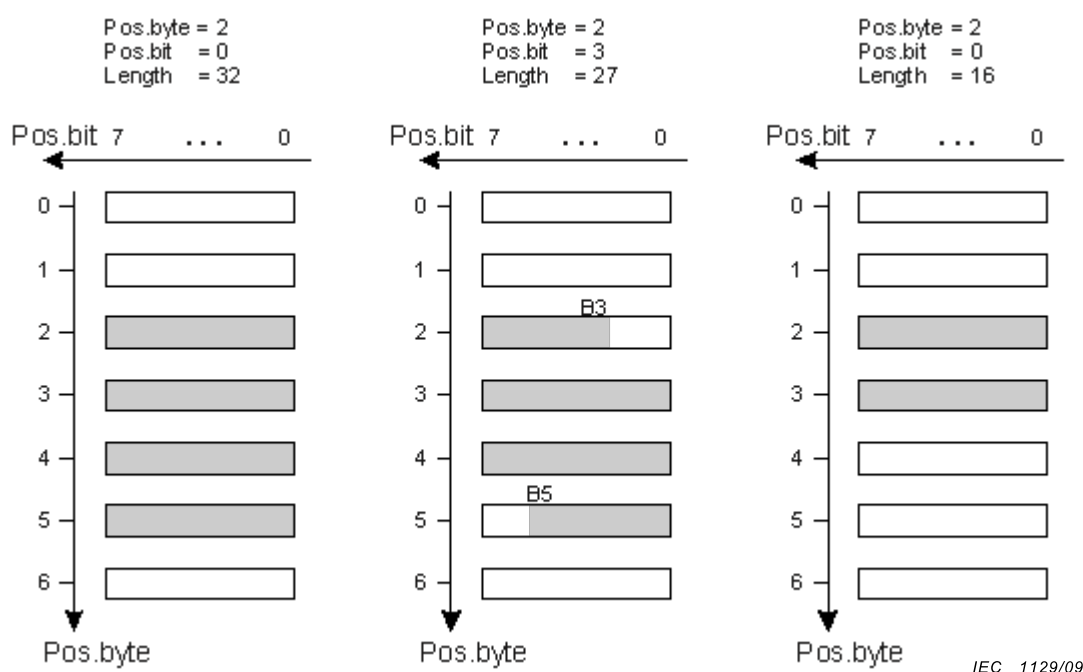
Nom	Type	Commentaire
Slave_Para_Len	UINT	Longueur du jeu de paramètres d'esclave, y compris le paramètre de longueur
SI_Flag	USINT	Fanions spécifiques à un esclave tels que New_Prm, Active, Fail_Safe,
Slave_Type	USINT	Désignation de type d'esclave spécifique à un fabricant (0 par défaut pour les esclaves DP)
Octet 1 (réservé)	ARRAY[12] of USINT	
...		
Octet 12 (réservé)		
Prm_Data_Len	UINT	Longueur de Prm_Data, y compris le paramètre de longueur (page 9 à 246)
Prm_Data	ARRAY OF USINT	
Cfg_Data_Len	UINT	Longueur de Cfg_Data, y compris le paramètre de longueur (page 3 à 246)
Cfg_Data	ARRAY OF USINT	
Add_Tab_Len	UINT	Longueur de Add_Tab, y compris le paramètre de longueur (page 2 à 216-31)
Add_Tab	ARRAY OF USINT	Table d'affectation d'adresses
Slave_User_Data_Len	UINT	Longueur de Slave_User_Data, y compris le paramètre de longueur (page 2 à 216-31)
Slave_User_Data	ARRAY OF USINT	Données spécifiques à un fabricant pour caractériser l'esclave DP pour le maître.

9.4 Données de module et de voie

Une configuration de module courante d'un esclave, y compris les voies connexes, doit être mise à disposition par les objets FdtChannel. Cela est indispensable, car au sein de l'environnement, les variables de processus doivent être affectées à des voies seules et la configuration d'esclave doit être affichée dans la vue d'ensemble du système de l'environnement sans utiliser l'interface utilisateur du DTM.

L'adressage de voies au sein de la trame de données PROFIBUS est orienté bit et indépendant du type de données. Une position de bits et une longueur en bits déterminent chaque voie. Le type de données d'un paramètre détermine comment convertir le champ de bits.

- Voies au sein des trames de données PROFIBUS (Figure 3):



Légende

Anglais	Français
Pos. byte	Octet Pos.
Pos Bit	Bit Pos.
Lenght	Longueur

Figure 3 – Exemple pour données E/S dans des datagrammes

- Représentation dans la notation FDT:

bitPosition="16"

bitPosition="19"

bitPosition="16"

bitLength="32"

bitLength="27"

bitLength="16"

L'exemple suivant montre la description de la structure attendue d'un esclave modulaire dans un jeu de paramètres DTM utilisant des paramètres FDT avec des types d'objets définis dans la Spécification FDT.

- Représentation de la trame de données utilisant des informations de GSD:

```
BeginSlave;;
```

```
RIOLB8101.GSD;6;
```

BeginModules;;

1X03 FrequencyCount;0x51;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

2XXX ValveBlock.;0x30;

EndModules;;

EndSlave;;

Description de la position de voies de signal au sein de la trame de données ():

Tableau 12 – Voies de signal au sein de la trame de données

# Frame description Remote I/O (Description de trames - E/S distantes)											
Vendeur:	"RIO manufacturer"										
Devicetype:	"LB 8101"		Device ident:			0x8101	# hex				
Frameident:	"B1t1"										
Données d'entrée											
Voie	Type de voie	Données	Données	Données	Bit non valide	Bit non valide	Bit de valeur de substitution	Bit de valeur de substitution	Bit de simulation	Bit de simulation	État-voie
		Position Octet.	Position Bit.	Longueur	Position Octet	Position Bit	Position Octet	Position Bit	Position Octet	Position bit	
Module:	"EP01"	"Digital Input (Counter) LB/FB 1X03 (4 Byte)"									
"Count_1_0"	READ INT	0	0	32	-/-	-/-	-/-	-/-	-/-	-/-	0
Module:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input and Output)"									
"DI_8_LFOUT0"	READ BOOL	4	1	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF1"	READ BOOL	4	3	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF2"	READ BOOL	4	5	1	-/-	-/-	-/-	-/-	-/-	-/-	1
Données de sortie											
Voie	Type de voie	Données	Données	Données	Bit non valide	Bit non valide	Bit de valeur de substitution	Bit de valeur de substitution	Bit de simulation	Bit de simulation.	État-voie
		Position Octet.	Position Bit.	Longueur	Position Octet.	Position Bit.	Position Octet.	Position Bit.			
Modul:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input u. Output)"									
"DO_8_0"	WRITE BOOL	0	0	1	4	1	-/-	-/-	-/-	-/-	0

9.5 Informations de GSD

9.5.1 Généralités

Les informations de GSD ne sont pas stockées avec des instances d'esclaves individuelles ou dans un fichier accessible au niveau global. Elles sont fournies par le DTM au niveau du service GetTypeInformation. Un DTM d'un dispositif PROFIBUS fournit les informations de GSD au sein de l'argument fdt:DtmDeviceType.

Le DCS existant utilise les informations de GSD pour obtenir des informations relatives à la configuration et aux paramètres possibles d'un esclave DP.

Outre les informations relatives aux modules et à ses paramètres, un fichier GSD contient des informations complémentaires concernant l'esclave, telles que les débits en bauds pris en charge.

Ces informations sont utiles pour un système DCS pour configurer un réseau tout entier selon les capacités des différents esclaves.

9.5.2 GSD pour dispositifs passerelles

9.5.2.1 Types de dispositifs passerelles Profibus

Il existe deux types de dispositifs passerelles.

Les dispositifs passerelles visibles fonctionnent comme un esclave PROFIBUS (avec une adresse PROFIBUS) et pour le réseau sous-jacent, ils agissent comme un maître.

Les dispositifs passerelles invisibles transforment juste le réseau PROFIBUS en le réseau sous-jacent.

Pour les deux types de dispositifs, il existe un besoin de GSD particulières.

9.5.2.2 Dispositifs passerelles visibles

Ces dispositifs passerelles visibles sont livrés sans un fichier GSD. À la place, ils ont une suite logicielle propriétaire qui le configure et le paramètre ou bien ils sont livrés avec un outil qui crée une GSD pour logiciel de paramétrisation. L'outil de GSD crée une GSD pour esclave en fonction de la configuration du réseau sous-jacent ou en fonction des valeurs de réglage des bus (les débits en bauds, par exemple).

Dans ces cas, il convient que le DTM fournisse la fonctionnalité de l'outil de GSD. Si la GSD est dépendante de la configuration, il pourrait appeler le service NetworkManagementInfoRead pour chacun de ses enfants, extraire les informations de GSD et créer des informations de GSD dépendantes de la configuration de la même manière que le fait l'outil. Après initialisation du DTM, il convient qu'il délivre une BMCP en fonction du dispositif de liaison lui-même. Chaque fois qu'un enfant est ajouté ou retiré, cela conduit à un changement dans son propre document de paramètres.

Si le GSD dépend des valeurs de réglage du bus, un dialogue de configuration ou de paramétrisation de DTM pourrait être utilisé pour modifier les valeurs de réglage du bus. Sur la base de ces valeurs de réglage, des informations de GSD mises à jour peuvent être insérées dans le document d'informations. Ici aussi, le DTM doit demander une sauvegarde et appeler le service OnInstanceDataChanged.

Noter que la structure interne avec ses modules et ses voies doit être mise à jour également.

9.5.2.3 Dispositifs passerelles transparents

Il existe des dispositifs de liaison transparents sur le marché (PROFIBUS FMS/DP et PROFIBUS PA) qui accomplissent une transformation de débits en bauds. Cela exige un traitement particulier des fichiers GSD spécifiques à un esclave. Il existe des outils disponibles qui sont capables d'adapter des fichiers GSD existants selon le débit en bauds plus élevé (lesdits "Convertisseurs de GSD").

Les informations de GSD doivent être délivrées par le DTM pour le dispositif. Afin de prendre en charge cette sorte de dispositifs de liaison, un DTM d'esclave doit exposer le fichier GSD sur le disque dur en utilisant l'attribut 'deviceTypeInfoPath' (se référer à DtmDeviceType défini dans l'IEC 62453-2).

Il convient qu'un développeur de DTM garde à l'esprit que fichier GSD référencé pourrait être converti afin de prendre en charge l'architecture de système spécifique.

Un DTM Profibus dans l'attribut 'deviceTypeInfo' est censé exposer exactement le fichier GSD qui est référencé par l'attribut 'deviceTypeInfoPath'.

10 Types de données de communication

10.1 Généralités

Les types de données définis ici sont utilisés au:

- service Connect (connexion);
- au service Disconnect (déconnexion);
- au service Transaction (transaction).

10.2 Informations d'erreur fournies par la voie de communication Communication Channel

Dans chaque type de données Response de la Spécification FDT Profibus, il est fourni un élément 'errorCode'. Selon PROFIBUS, le code d'erreur est normalisé pour être constitué de trois octets, chaque octet contenant une signification.

Sachant que le code d'erreur est échangé entre des DTM différents (par exemple: DTM de communication et DTM de dispositif) et que le destinataire du code d'erreur tentera de comprendre le code d'erreur, le fournisseur doit utiliser le format normalisé:

- longueur normalisée de trois octets;
- si le dispositif fournit des codes d'erreur, ces codes d'erreur sont fournis (et non des codes d'erreur locaux issus du maître).

Si aucune erreur ne s'est produite, l'élément 'errorCode' doit être rempli de zéros ("000000").

10.3 Communication DPV0

Les services pris en charge dépendent du type de fonctionnalité de maître PROFIBUS qui est fournie par l'infrastructure de communication. Les dispositifs maîtres de classe 1 commandent typiquement les esclaves et fournissent une communication cyclique, les dispositifs maîtres de classe 2 sont typiquement utilisés pour configurer les esclaves et fournir une communication acyclique.

Tous les services définis ne sont pas pris en charge si le maître n'est pas engagé dans un échange cyclique de données avec les esclaves. Dans de tels cas, le comportement suivant est attendu:

Si une voie de communication reçoit une demande qui ne peut pas être prise en charge, le service Transaction répond par result="false".

Les types de données décrits dans le présent article sont définis pour l'espace de noms suivant:

Espace de noms: dpv0

Tableau 13 – Types de données simples de communication DPV0

Type de données	Définition	Description
busAddress	USINT	Informations d'adresse selon la série IEC 61158 (voir aussi l'IEC 62453-2, attribut busAddress)
communicationReference	UUID	Identificateur obligatoire pour une liaison de communication à un dispositif. Cet identificateur est alloué par le composant de communication pendant la création de la connexion. Les informations d'adresse doivent être utilisées pour tous les appels de communication suivants
connectStatus	enumeration (masterConnectedOnly deviceAtLifeList deviceInDataExchange)	Décrit le statut de connexion établi par le composant de communication. Le statut "masterConnectedOnly" signifie que le composant de communication a établi une connexion au dispositif maître Profibus et acceptera un accès en ligne aux paramètres utilisateur, indépendamment du fait que le dispositif soit disponible ou non. Le statut "deviceAtLifeList" signifie que le composant de communication a établi une connexion au dispositif maître Profibus et a vérifié que le dispositif est dans la liste de vie de la pile du maître. Dans cet état, le maître acceptera un accès en ligne aux paramètres utilisateur et enverra le paramètre utilisateur au dispositif, indépendamment du fait que le dispositif soit engagé dans un échange de données ou non. Le statut "deviceInDataExchange" signifie que le composant de communication a établi une connexion au dispositif maître Profibus et a vérifié que le dispositif est engagé dans un échange de données. Dans cet état, le maître acceptera un accès en ligne aux paramètres utilisateur et enverra le paramètre utilisateur au dispositif, afin que les nouvelles données influencent directement le processus.
delayTime	UDINT	Retard en [ms] entre deux appels de communication
errorCode	ARRAY OF USINT	Informations de statut selon la série IEC 61158. Pour la description de code d'erreur, voir: la DIN 19245 Partie 3, PROFIBUS (P. 40ff., 83ff., 39)
schemaVersion	INT	Définit la version du schéma
sequenceTime	UDINT	Durée en [ms] pour toute la séquence

Tableau 14 – Types structurés de données de communication DPV0

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
Abort	STRUCT			Décrit l'abandon	-
	communicationReference	O	[0..1]		
ConnectRequest	STRUCT			Décrit la demande de communication pour établir une connexion à un dispositif maître Profibus. Selon l'infrastructure de réseau utilisée, il est possible que ce service ne soit pas mis en correspondance avec une demande de bus de terrain, mais soit utilisé pour gérer le logiciel. Il est également possible que ce service soit utilisé comme déclencheur pour établir l'état d'un dispositif.	-
	busAddress	M	[1..1]		
	fdt:systemTag	O	[0..1]		
ConnectResponse	STRUCT			Décrit la réponse de communication à la demande de connexion et donne les informations sur la manière dont les commandes WriteUserParameter suivantes seront envoyées vers le dispositif (voir connectStatus). La connexion résultante dépend du dispositif de communication utilisé et de la configuration du maître bus. Le DeviceDTM doit s'attendre à ce que la connexion résultante ne puisse pas être utilisée pour accéder à tous les services. ConnectResponse peut retourner "true" même s'il n'y a aucune communication possible avec le dispositif. Par exemple, si ConnectRequest retourne "true" avec connectionStatus = "masterConnectedOnly", cela signifie qu'il y a une connexion au dispositif maître, mais le dispositif n'est pas sur la liste de vie et ne peut donc pas être accessible. En fonction du connectStatus, certains services peuvent ne pas être disponibles (par exemple: dans le état "masterConnectedOnly", seuls des services fournis par le maître sont accessibles (ReadUserParameter, WriteUserParameter, ...) attendu que les services fournis par le dispositif ne sont pas accessibles. NOTE Dans cet exemple, le DTM est dans un état "online" (en ligne), mais il ne peut pas accéder au dispositif. Les services fournis dépendent également du type du dispositif maître. Pour une vue d'ensemble de la disponibilité des services en fonction du type de connexion, voir les tableaux suivants.	-

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
	connectStatus	M	[1..1]		
Disconnect ¬Request	STRUCT			Décrit la demande de communication pour libérer une connexion à un dispositif maître Profibus.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
Disconnect ¬Response	STRUCT			Décrit la réponse de communication.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
ReadDiagnosis ¬Data ¬Request	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Pour la Cl.1 de maître, se référer à l'IEC 61158-5, 8.2.3.3.2 "Get slave diag". Pour la Cl.2 de maître, se référer à l'IEC 61158-5, 8.2.3.3.3 "Read slave diag"	1, 2
	communicationReference	M	[1..1]		
ReadDiagnosis ¬Data Response	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
ReadInputData→Request	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Selon que la voie FDTChannel sous-jacente est de la Cl. 1 de maître ou de la Cl. 2 de maître, ce service sera local ou conduira à un accès en lecture au dispositif esclave. Il est possible de spécifier quelles données sont lues en fournissant un fdt:ChannelReference. Pour la Cl.1 de maître, se référer à l'IEC 61158-5, 8.2.2.3.3 "Get Input". Pour la Cl.2 de maître, se référer à l'IEC 61158-5, 8.2.2.3.2 "Read Input"	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadInputData→Response	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadOutputData→Request	STRUCT			Décrit la demande de communication selon la Spécification Profibus DPV0. Selon que la FDTChannel sous-jacente est de la Cl. 1 de maître ou de la Cl. 2 de maître, ce service sera local ou conduira à un accès en lecture au dispositif esclave. Il est possible de spécifier quelles données seront lues en fournissant un fdt:ChannelReference. Pour la Cl. 1 de maître, il s'agit d'un service propriétaire (ou non disponible). Pour la Cl. 2 de maître, se référer à [1], section 8.2.2.3.6. "Read Output" (un service facultatif)	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadOutputData→Response	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
ReadUserParameterRequest	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. La demande récupère uniquement les informations qui sont disponibles auprès du dispositif maître (service local). Elles peuvent différer des données réelles du dispositif. La mise en œuvre de la manière de fournir les données est propriétaire. Elle est basée sur un service local ou n'est pas disponible.	1
	communicationReference	M	[1..1]		
ReadUserParameterResponse	STRUCT			Décrit la réponse de communication, lue dans le dispositif maître Profibus, conformément à la spécification Profibus DPV0. Les données retournées peuvent ne refléter que les informations qui sont mises à disposition par le dispositif maître. Elles peuvent différer des données réelles du dispositif	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
SequenceBegin	STRUCT			Décrit la séquence begin	-
	sequenceTime	O	[0..1]		
	delayTime	O	[0..1]		
	communicationReference	M	[1..1]		
SequenceEnd	STRUCT			Décrit la séquence end	-
	communicationReference	M	[1..1]		
SequenceStart	STRUCT			Décrit la séquence start	-
	communicationReference	M	[1..1]		

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
WriteOutputDataRequest	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Les données de sortie seront envoyées selon la connexion établie. Il est nécessaire de spécifier quelles données seront écrites par fourniture d'un fdt:ChannelReference. Si la FDTChannel sous-jacente est fournie comme Cl. 2 de maître, l'écriture de données de sortie (Output Data) ne sera pas possible. Se référer à l'IEC 61158-5, 8.2.2.3.5 "Set Output". Cette fonctionnalité dépend du type de maître utilisé. En termes de PLC (automate programmable), cette fonctionnalité est appelé "forcing" de valeurs.	1
	communicationReference	M	[1..1]		
	fdt:ChannelReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteOutputDataResponse	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
WriteUserParameterRequest	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Le paramètre utilisateur sera envoyé selon la connexion établie (voir connectStatus). Pour la Cl. 1 de maître, cela est basé sur le service "DDL_M_Set_Prm". Pour la Cl. 2 de maître, cela est basé sur le service "DDL_M_Set_Prm" (un service facultatif).	1, 2
	communicationReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteUserParameterResponse	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Selon le type de maître bus et le connectStatus retourné, les services suivants sont disponibles.

Tableau 15 – Disponibilité des services pour la classe 1 de maître (C1)

Demande de services de DTM d'esclave	connectStatus		
	masterConnectedOnly	DeviceAtLifeList	DeviceInDataExchange
Connect	✓	✓	✓
ReadUserParameter	O	O	O
WriteUserParameter	✓	✓	✓
ReadOutputData			✓
WriteOutputData			O
ReadInputData			✓
ReadDiagnosisData		✓	✓
NOTE ✓: le service est disponible, O: le service est facultatif et peut être disponible, selon les capacités du dispositif maître sous-jacent.			

Pour la Class2 de maître (C2), tous les états "connect" (connexion) ne sont pas nécessairement disponibles:

Tableau 16 – Disponibilité des services pour la classe 2 de maître (C2)

Action de DTM d'esclave	connectStatus	
	DeviceAtLifeList (aucune connexion DPV1 au dispositif)	DeviceInDataExchange (connexion DPV1 au dispositif)
Connect	✓	✓
ReadUserParameter		
WriteUserParameter	O	O
ReadOutputData		O
WriteOutputData		
ReadInputData		✓
ReadDiagnosisData	✓	✓
NOTE ✓: le service est disponible, O: le service est facultatif et peut être disponible, selon les capacités du dispositif maître sous-jacent.		

Si le composant de communication de la classe 2 de maître prend en charge DPV1 et DPV0 et a établi une connexion DPV1 au dispositif, un appel au service Connect pour DPV0 doit retourner le statut "DeviceInDataExchange" même si le dispositif n'est pas dans le statut DataExchange.

Si aucune connexion DPV1 n'est établie, le composant de communication de la classe 2 de maître doit vérifier la disponibilité du dispositif (au moins par le service LifeList).

10.4 Communication DPV1

Les types de données sont définis pour prouver les informations d'adresse et les données de communication requises pour la communication DPV1 (voir Tableau 17 et Tableau 18).

Les types de données décrits dans le présent paragraphe sont définis pour l'espace de noms suivant:

Espace de noms: dpv1

Tableau 17 – Types de données simples de communication DPV1

Type de données	Définition	Description
api	USINT	Informations d'adresse selon la série IEC 61158. Si le dispositif a besoin de valeurs particulières pour DPV1-Initiate, le DeviceDTM a la responsabilité de fournir les valeurs dans la ConnectRequest. Si les valeurs ne sont pas fournies, le composant de communication utiliser des valeurs par défaut (0)
busAddress	USINT	Informations d'adresse selon la série IEC 61158 (voir aussi DTMPParameterSchema, attribut busAddress)
errorCode	UUID	Informations de statut selon la série IEC 61158. Pour la description de code d'erreur, voir: PROFIBUS Nutzerorganisation e.V., PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998. (P.. 120-121); .Pour la description des informations d'abandon, voir: la DIN 19245 Partie 3, PROFIBUS, (P..54ff., 11)
index	UDINT	Informations d'adresse selon la série IEC 61158.
communicationReference	ARRAY OF USINT	Identificateur obligatoire pour une liaison de communication à un dispositif. Cet identificateur est alloué par le composant de communication pendant la connexion. Les informations d'adresse doivent être utilisées pour tous les appels de communication suivants
delayTime	USINT	Retard minimal en [ms] entre deux appels de communication
maxLenDataUnit	USINT	Attribut facultatif, pour décrire la quantité des données qui peuvent être transférées par la connexion établie. Si cet attribut n'est pas valide, aucune restriction de longueur particulière n'est annoncée. Chaque composant de communication au sein de la chaîne d'interfaces concernant la communication imbriquée pourrait introduire cet attribut. Il convient que chaque composant de communication change le contenu de l'attribut en se basant sur la règle suivante: La nouvelle valeur est le minimum de la valeur courante et de la restriction de sa propre mise en œuvre. Si un composant de communication n'a pas de restriction, il convient qu'il transfère la valeur donnée. Si un composant de communication est capable de réutiliser une connexion établie concernant une nouvelle demande de connexion, il convient qu'il prenne en compte la longueur des données déterminée pour la connexion existante. Si la longueur de données n'est pas applicable, il convient que le DTM envoie un message d'erreur au moyen du service ErrorMessage
networkMACAddress	ARRAY OF USINT	Adresse réseau ou adresse MAC telle que décrite dans la série IEC 61158. [5] S'il y a une networkMACAddress, (à savoir networkMACAddress <> ""), le Profibus-DPV1-Master-DTM doit mettre à "1" le "address type" (type d'adresse) correspondant dans le télégramme de demande Initiate, autrement à "0". Exemple Dispositif avec l'adresse 20 (14hex) adressé par l'intermédiaire d'un dispositif de liaison: networkMACAddress="00000000000014"
scl	INT	Le niveau d'accès tel que décrit dans la série IEC 61158. [5] Si le dispositif a besoin de valeurs particulières pour DPV1-Initiate, le DeviceDTM a la responsabilité de fournir les valeurs dans la ConnectRequest. Si les valeurs ne sont pas fournies, le composant de communication utiliser des valeurs par défaut (0)
sequenceTime	USINT	Durée en [ms] pour toute la séquence
slot	UDINT	Informations d'adresse selon la série IEC 61158.

Tableau 18 – Types de données structurés de communication DPV1

Type de données	Définition			Description
	Types de données élémentaires	Usage	Multi-plicité	
Abort	STRUCT			Décrit l'abandon
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Décrit la demande de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	fdt.systemTag	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
ConnectResponse	STRUCT			Décrit la réponse de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	maxLenDataUnit	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
DestNetworkAddress	STRUCT			Décrit l'adresse étendue de la destination [5].
	NetworkAddress	M	[1..1]	
DisconnectRequest	STRUCT			Décrit la demande de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Décrit la réponse de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
NetworkAddress	STRUCT			Décrit le format d'adresse étendue.
	api	M	[1..1]	
	scl	M	[1..1]	
	networkMACAddress	M	[1..1]	
ReadRequest	STRUCT			Décrit la demande de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multi- plicité	
ReadResponse	STRUCT			Décrit la réponse de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
RedundantAddresses	STRUCT			Décrit les adresses redondantes.
	fdtparam:SlaveAddress	M	[1..*]	
SequenceBegin	STRUCT			Décrit la séquence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	
SequenceEnd	STRUCT			Décrit la séquence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Décrit la séquence start
	communicationReference	M	[1..1]	
SrcNetworkAddress	STRUCT			Décrit l'adresse étendue de la source.
	NetworkAddress	M	[1..1]	
WriteRequest	STRUCT			Décrit la demande de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
WriteResponse	STRUCT			Décrit la réponse de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	

Au sein d'une demande de connexion, un DTM d'un esclave redondant Profibus peut fournir des adresses complémentaires d'esclaves redondants, établies au sein de son document de paramètres. L'attribut busAddress doit être utilisé comme l'adresse préférentielle; il convient que les adresses dans l'élément redundantAddresses soient utilisées dans cet ordre si une adresse de remplacement est utilisée pour connecter l'esclave redondant.

11 Types de données paramètres de voie (Channel)

Le tableau suivant (Tableau 19) décrit la mise en correspondance des types de données DPV1 avec les types de données FDT.

**Tableau 19 – Mise en correspondance des types de données DPV1
 aux types de données FDT**

Types de données PROFIBUS	Type de données FDT
Boolean	BOOL
Integer8	SINT
Integer16	INT
Integer32	DINT
Unsigned8	USINT
Unsigned16	UINT
Unsigned32	UDINT
Floating Point ("Virgule flottante")	FLOAT
Visible String	STRING
Octet String	ARRAY OF USINT
Date	DATE
Time of Day	TIME
Time Difference	duration
DS-32	structured (structuré)
DS-33	structured (structuré)
DS-34	structured (structuré)
DS-35	structured (structuré)
DS-36	structured (structuré)
...	structured (structuré)
DS-52	structured (structuré)
..	structured (structuré)

Le format de tous les types de données est défini dans la série IEC 61158.

Si seule une partie des données récupérées est constituée de données de processus (par exemple: pour DS-33, les données sont constituées de la valeur float et du statut d'octet), les informations d'accès (DPAAddress ou DPV1Address) contiennent également des informations qui sont nécessaires pour récupérer les informations issues de la structure de données.

Utilisés au service ReadChannelInformation et au service WriteChannelInformation.

Les types de données ~~ChannelParameter~~ **paramètres de voies** décrivent le mode d'accès à une voie par une commande Profibus DPV1 ou ~~comment adresser~~ **le mode d'adressage d'une** voie au sein d'un cadre Profibus DP pour **une** E/S cyclique. Les tableaux suivants (Tableau 20 et Tableau 21) fournissent une description des types de données.

Les types de données décrits dans le présent article sont définis pour l'espace de noms suivant:

Espace de noms: profichannel

Tableau 20 – Types simples de données-ChannelParameter pour les paramètres de voies

Type de données	Définition	Description
api	USINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1
bitLength	UDINT	Informations complémentaires sur les types de données, notamment pour les types de données spécifiques aux bus de terrain comme entier de 12 bits
bitPosition	UDINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DP
frameApplicationTag	STRING	Étiquette spécifique à l'application-cadre utilisée pour l'identification et la navigation. Il convient que le DTM affiche cette étiquette aux interfaces utilisateur spécifiques à une voie.
gatewayBusCategory	UUID	Identificateur unique pour un type de bus pris en charge comme Profibus ou HART conformément au CATID spécifique à FDT
index	USINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1
invalidBit	UDINT	Position de bit de la voie de statut invalide accessible par le biais de Profibus DP
logic	UDINT	Informations de types de données supplémentaires: positive 0=FALSE 1=TRUE
number	BOOL	Informations d'adresse pour diagnostic selon la série IEC 61158 pour les voies accessibles par Profibus DP Voir [5], 9.3.1 Read DP – Slave Diagnostic Information
protectedByChannelAssignment	INT	TRUE si la voie est mise à lecture seule par la Frame Application (application-cadre). Habituellement mis à TRUE s'il existe une attribution de voie.
schemaVersion	UDINT	Définit la version du schéma.
simulationBit	USINT	Position de bit de la voie de statut de simulation accessible par le biais de Profibus DP
slotNumber	BOOL	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1
statusChannel	UDINT	TRUE si la voie est pour informations de statut seulement.
substituteValueBit	STRING	Position de bit de la voie de statut de substitution accessible par le biais de Profibus DP

Tableau 21 – Types structurés de données ~~Channel~~Parameter pour les paramètres de voies

Type de données	Définition			Description
	Type de données élémentaire	U s a g e	Multi- PLICITÉ	
DpAddress	STRUCT			Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DP
	bitPosition	M	[1..1]	
	bitLength	M	[1..1]	
DpV1Address	STRUCT			Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1. Il est fortement recommandé de fournir des informations d'adresse pour permettre l'affectation d'adresses logiques à un module profibus.
	api	M	[1..1]	
	slotNumber	M	[1..1]	
	index	M	[1..1]	
	bitPosition	O	[0..1]	
	bitLength	O	[0..1]	
FDTChannel	STRUCT			Description de la voie
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannelAssignment	M	[1..1]	
	number	M	[1..1]	
	fdt:dataType	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	applId:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	DpAddress	O	[0..1]	
	DpV1Address	O	[0..1]	
	StatusInformation	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:StructuredElements	O	[0..1]	
FDTChannelType	STRUCT			Description du composant de voie dans le cas des voies avec fonctionnalité de passerelle
	fdt:VersionInformation	M	[1..1]	
	gatewayBusCategory	O	[0..1]	
	statusChannel	O	[0..1]	
StatusInformation	STRUCT			Description d'informations de statut complémentaires pour les voies accessibles par Profibus DP

Type de données	Définition			Description
	Type de données élémentaire	U s a g e	Multi- plicité	
	logic	M	[1..1]	
	invalidBit	O	[0..1]	
	simulationBit	O	[0..1]	

12 Identification de dispositif

12.1 Généralités

Il existe différents éléments d'identification spécifiques à Profibus.

Un Profibus Scan (balayage Profibus) peut détecter différents types de dispositifs: dispositifs I&M, dispositifs Profile ou purs dispositifs DP.

La règle suivante doit être appliquée pour des Profibus Communication Channels (voies de communication Profibus):

- Si I&M est disponible, créer l'identification I&M,
- autrement vérifier Profile PA,
- autrement créer information purement DP = IDENT_NUMBER.

Les types de données décrits dans le présent paragraphe sont définis pour l'espace de noms suivant:

Espace de noms: profiident

12.2 Traitement, spécifique à un protocole, du type de données STRING

Règles pour char array (matrices de caractères) Profibus:

- dans toutes les chaînes basées sur des plages de caractères (char) définies dans la spécification de protocole de bus de terrain, les espaces de têtes sont rognés. La matrice de caractères (char array) doit être remplie de caractères 0x20h (blanc);
- dans les VisibleString, les caractères invisibles fournis par un dispositif doivent être remplacés par '?'

12.3 Types de données d'identification de types de dispositif communs

Les types de données décrits dans cet article sont réutilisés comme défini par 12.4 et 12.6 .

Le Tableau 22 énumère les données, source et format pertinents pour l'identification pour Profibus DP.

Tableau 22 – Types de données d'identification avec mise en correspondance spécifique à Profibus DP

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Type de données FDT (format d'affichage)	Référence de la spécification
busProtocol	IdBusProtocol	Pour tous les dispositifs DP: "protocol_DP"	Protocol	-	Enum	-
slaveAddress	IdAddress	L'adresse de bus est fournie comme partie intégrante de la liste de vie par un maître PROFIBUS. Service: FMA1/2_LIVE_LIST	Slave Address	Unsigned8	STRING (décimal)	[5] Partie 2, Section 4.2.3.6
identNumber	IdTypeID	IDENT_NUMBER peut être demandé par: DP Service DDLM_SLAVE_DIAG Les valeurs permises sont: IDENT_NUMBER de profil: 0x9700 (0x9700 à 0x9742) ou IDENT_NUMBER spécifique à un fabricant	IDENT_NUMBER	Unsigned16	ARRAY of USINT (hexadécimal)	[5] Partie 8, Section 9.3.1
manufacturerSpecificExtension		Peut être utilisé par DTM pour des informations d'identification de dispositif spécifiques à un vendeur, par exemple en combinant un certain nombre de valeurs de paramètres de dispositif en une seule valeur string (chaîne). Cela peut être utilisé pour identifier une variante de dispositif spécifique.			STRING	

Le Tableau 23 montre la mise en correspondance pour les données I&M PROFIBUS.

Tableau 23 – Types de données d'identification avec mise en correspondance spécifique à Profibus I&M

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Format FDT (format d'affichage)	Référence de la spécification
identNumber, manufacturerSpecificExtension et slaveAddress tels que définis dans le tableau DP (Tableau 22).						
busProtocol	IdBusProtocol	Pour tous les dispositifs I&M: "protocol_IM"	Protocol	-	Enum	-
manufacturerId	IdManufacturer	I&M 0 élément 1 mis en correspondance avec PB.DEVICE_MAN_ID	MANUFACTURER_ID	Unsigned16	UINT	[6] Section 3.2.2
orderId	-	I&M 0 élément 2 mis en correspondance avec PB.DEVICE_ID	ORDER_ID	20 Octets Visible String	STRING[20]	[6] Section 3.2.3
serialNumber	IdSerialNumber	I&M 0 élément 3 numéro de série unique mis en correspondance avec PB.DEVICE_SER_NUM	SERIAL_NUMBER	16 Octets VisibleString	STRING[16]	[6] Section 3.2.4
hardwareRevision	IdHardwareRevision	I&M 0 élément 4 HARDWARE_REVISION NOTE: Si le dispositif est un dispositif PA, (PROFILE_ID est mis à 0x9700), HARDWARE_REVISION doit être lu dans PA_I&M0 [6]	HARDWARE_REVISION	2 Octets - entier non signé de 16 bits converti en string (chaîne) pour couvrir également: PA_I&M0: VisibleString 16	STRING	[6] Section 3.2.5, Amendement I&M
softwareRevision	IdSoftwareRevision	I&M 0 élément 5 SOFTWARE_REVISION NOTE: Si le dispositif est un dispositif PA (PROFILE_ID est mis à 0x9700), SOFTWARE_REVISION doit être lu dans PA_I&M0 Le format String est utilisé pour harmoniser la gestion des versions pour tous les protocoles de bus de terrain.	SOFTWARE_REVISION	4 Octets – 1 char + 3 unsigned 8 , par exemple: V1.2.3 PA_I&M0: VisibleString 16	STRING	[6] Section 3.2.6, Amendement I&M

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Format FDT (format d'affichage)	Référence de la spécification
profileID	-	I&M 0 élément 7	PROFILE_ID	2 Octets – unsigned 16	UINT	[6] Section 3.2.8
profileSpecificType	-	I&M 0 élément 8 MSB = BLOCK_OBJECT, LSB = PARENT_CLASS	PROFILE_SPECIFIC_TYPE	2 Octets – unsigned 16	UINT	[6] Section 3.2.9
imVersion	-	I&M 0 élément 9	IM_VERSION	2 Octets – unsigned 16; MSB version majeure xxx, LSB version mineure yyy	FLOAT xxx.yyy	[6] Section 3.2.10
imSupported	-	I&M 0 élément 10	IM_SUPPORTED	2 Octets -	UINT	[6] Section 3.2.11
tagFunction	IdTag	I&M 1 élément 1 TAG_FUNCTION	TAG_FUNCTION	32 Octets – visible string	STRING[32]	[6] Section 3.2.12
tagLocation	-	I&M 1 élément 2 TAG_LOCATION	TAG_LOCATION	22 Octets – visible string	STRING[22]	[6] Section 3.2.13
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM ProfileDTM, BlockSpecificProfileDTM	Niveau de prise en charge de DTM	-	Enum	

Le Tableau 24 énumère les données, source et format pertinents l'identification pour Profibus PA.

Tableau 24 – Types de données d'identification avec mise en correspondance spécifique à Profibus PA

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Format FDT (format d'affichage)	Référence de la spécification
identNumber, manufacturerSpecificExtension et slaveAddress tels que définis dans le tableau DP (Tableau 22).						
busProtocol	IdBusProtocol	Pour tous les dispositifs PA: "protocol_PA"	Protocol	-	enum	-
manufacturerId	IdManufacturer	Physical Block - Index 10 (Unsigned16 – 2 decimal)	DEVICE_MAN_ID	Unsigned16	UINT	[2] Partie 1, Section 3.11
device_id	-	Physical Block - Index 11 - VisibleString 16	DEVICE_ID	VisibleString16	STRING[16]	[2] Partie 1, Section 3.11
profile	-	Structure en blocs du bloc physique – élément 7: OctetString - (Index 10 – size 2)	Profile	OctetString2	UINT	[2]0 Partie 1, Section 3.7.2
profileRevision	IdBusProtocolVersion	Structure en blocs de bloc physique – élément 8 Unsigned 16 - (Index 6)	Profile Revision (révision de profil)	Unsigned16	UINT	[2] Partie 1, Section 3.7.2
softwareRevision	IdSoftwareRevision	Physical Block - Index 8 (VisibleString 16)	SOFTWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
hardwareRevision	IdHardwareRevision	Physical Block - Index 9 (VisibleString 16)	HARDWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
serialNumber	IdSerialNumber	Physical Block - Index 12 (VisibleString 16)	DEVICE_SER_NUM	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
deviceTAG	IdTag	BO of PB - OctetString 32	TAG_DESC	OctetString32	STRING[16]	
profileSpecificType	-	Attribut FDT mis en correspondance avec Block Object PARENT_CLASS sorte de premier TransducerBlock	BO PARENT_CLASS	Unsigned16	UINT	[2] Partie 1, Section 3.7.2
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	Niveau de prise en charge de DTM		enum	

Le Tableau 25 et le Tableau 26 montrent les types de données d'identification avec une sémantique indépendante vis-à-vis du protocole.

Tableau 25 – Types de données d'identification simples avec sémantique indépendante vis-à-vis du protocole

Type de données	Définition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	enumeration • genericSupport • profileSupport • blockspecificProfileSupport • specificSupport • identSupport
match	STRING	Utilisé par le DTM de dispositif pour définir une expression générique qui doit concorder avec les informations d'identification de définition physique balayées
nomatch	STRING	Utilisé par le DTM de dispositif pour définir une expression générique qui ne doit pas concorder avec les informations d'identification de définition physique balayées utilisé par le DTM de dispositif pour indiquer si les informations d'identification ne peuvent pas concorder.

Tableau 26 – Types de données d'identification structurés avec sémantique indépendante vis-à-vis du protocole

Type de données	Définition			Description
	Types de données élémentaires	Usage	Multiplicité	
RegExpr	STRUCT			Inclut une chaîne d'expression générique – soit pour match (concordance), soit pour nomatch (pas de concordance)
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.4 Types de données de balayage de topologie

Ces types de données sont utilisés à la réponse de service Scan (Tableau 27 et Tableau 28).

Tableau 27 – Types de données simples de balayage de topologie

Type de données	Définition	Description
busAddress	USINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1

Le type de données ProfibusDevice décrit une entrée dans la liste de dispositifs PROFIBUS balayés (voir Tableau 28).

Tableau 28 – Types de données structurés de balayage de topologie

Type de données	Définition			Description
	Type de données élémentaire	Usage	Multiplicité	
ProfibusDevice	STRUCT			Spécifie un dispositif Profibus.
	schemaVersion	O	[0..1]	
	busAddress	M	[1..1]	
	fdt:deviceTypeId	M	[1..1]	
	fdt:subDeviceType	O	[0..1]	

12.5 Types de données d'identification de balayage (Scan)

Ce paragraphe définit les types de données qui sont utilisés pour fournir un balayage spécifique à un protocole (voir Tableau 29 et Tableau 30).

Les types de données décrits dans le présent paragraphe sont utilisés dans les services suivants:

- Service "Scan"

Tableau 29 – Types de données simples d'identification de balayage

Type de données	Définition	Description
configuredState	enumeration (configuredAndPhysicallyAvailable configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	Un maître de communication doit indiquer dans cet attribut si la réponse de balayage est relative à un dispositif physique détecté qui est configuré ou non configuré.
resultState	enumeration (provisional final error)	Identifie si le résultat est celui des résultats provisoires ou le résultat final des résultats de balayage divisés.

Tableau 30 – Types de données structurés d'identification de balayage

Type de données	Définition			Description
	Types de données élémentaires	Usage	Multiplicité	
IdAddress	STRUCT			Tous les types de données contiennent exactement un attribut chacun incluant la valeur du dispositif physique balayé. Tous les types de données ayant une signification sémantique ont un préfixe "Id" pour une meilleure identification.
	profibusident:slaveAddress	M	[1..1]	
IdBusProtocol	STRUCT			
	profibusident:busProtocol	M	[1..1]	
IdBusProtocolVersion				
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	M	[1..1]	
IdManufacturer				
IdSerialNumber	STRUCT			
	profibusident:serialNumber	M	[1..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	M	[1..1]	
IdTypeID	STRUCT			
	profibusident:identNumber	M	[1..1]	
Manufacturer_id	STRUCT			Ces types de données contiennent des attributs correspondants définis dans l'IEC 62453-2. Ils sont transformés en paires nom-valeur sans signification sémantique pour Frame Application
	profibusident:manufacturer_id	M	[1..1]	
Order_id	STRUCT			
	profibusident:order_id	M	[1..1]	
ProfileID	STRUCT			
	profibusident:profileID	M	[1..1]	
ProfileSpecificType	STRUCT			
	profibusident:profileSpecificType	M	[1..1]	
ImVersion	STRUCT			
	profibusident:imVersion	M	[1..1]	
TagFunction	STRUCT			
	profibusident:tagFunction	M	[1..1]	
TagLocation	STRUCT			
	profibusident:tagLocation	M	[1..1]	
ImSupported	STRUCT			
	profibusident:imSupported	M	[1..1]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:software↗RevisionPA	M	[1..1]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardware↗RevisionPA	M	[1..1]	
ProfileRevisionPA	STRUCT			
	profibusident:profileRevisionPA	M	[1..1]	
Device_man_id	STRUCT			
	profibusident:device_man_id	M	[1..1]	
Device_id	STRUCT			
	profibusident:device_id	M	[1..1]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profileSpecific↗TypePA	M	[1..1]	
ManufacturerSpecific↗Extension				
	STRUCT			
	profibusident:manufacturer↗SpecificExtension	M	[1..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
ScanIdentification_IM	STRUCT			Ces éléments contiennent tous les éléments pour la variante de protocole appropriée
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	ImVersion	M	[1..1]	
	TagFunction	M	[1..1]	
	TagLocation	M	[1..1]	
	ImSupported	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_PA	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	
	ProfileSpecificTypePA	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	IdTag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_DP	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
ScanIdentifications	STRUCT			Ensemble d'éléments ScanIdentification
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	choice of	M	[1..*]	
	ScanIdentification_DP	S	[0..*]	
	ScanIdentification_IM	S	[0..*]	
	ScanIdentification_PA	S	[0..*]	

12.6 Types de données d'identification de type de dispositif – fournis par DTM

Ce paragraphe définit les types de données qui sont utilisés pour fournir des informations spécifiques à un protocole pour des types de dispositifs (voir Tableau 31).

Les types de données décrits dans le présent paragraphe sont utilisés dans les services suivants:

- service GetIdentificationInformation

Tableau 31 – Types de données structurés d'identification de dispositif

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
IdBusProtocol	STRUCT			Tous les éléments contiennent exactement un attribut chacun incluant la valeur du dispositif physique balayé. Tous les éléments ayant une signification sémantique ont un préfixe "Id" pour une meilleure identification
	profibusident:busProtocol	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdBusProtocolVersion				
IdManufacturer				
IdTypeID	STRUCT			
	profibusident:identNumber	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Manufacturer_id	STRUCT			Ces éléments contiennent des attributs correspondants définis dans l'IEC 62453-2. Ils sont transformés en paires nom-valeur sans signification sémantique pour Frame Application
	profibusident:manufacturer_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
Order_id	STRUCT			
	profibusident:order_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileID	STRUCT			
	profibusident:profileID	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificType	STRUCT			
	profibusident:profile↯SpecificType	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImVersion	STRUCT			
	profibusident:imVersion	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImSupported	STRUCT			
	profibusident:imSupported	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:software↯RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardware↯RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileRevisionPA	STRUCT			
	profibusident:profile↯RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_man_id	STRUCT			
	profibusident:device_man_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_id	STRUCT			
	profibusident:device_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profile↯SpecificTypePA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ManufacturerSpecific↯Extension	STRUCT			
	profibusident:manufacturer↯SpecificExtension	M	[1..1]	
DeviceIdentification_IM	STRUCT			Ces éléments contiennent tous les

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
	profibusident:idDTM↗SupportLevel	M	[1..1]	éléments pour la variante de protocole appropriée
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	ImVersion	M	[1..1]	
	ImSupported	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification_PA	STRUCT			
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	
	ProfileSpecificTypePA	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification_DP	STRUCT			
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdTypeID	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification s	STRUCT			Ensemble d'éléments DeviceIdentification
	fdt:protocolId	M	[1..1]	
	choice of	M	[1..*]	
	DeviceIdentification_DP	S	[0..*]	
	DeviceIdentification_IM	S	[0..*]	
	DeviceIdentification_PA	S	[0..*]	

12.7 Informations d'identification dans l'interface GUI

Il est obligatoire d'afficher les étiquettes des paramètres I&M dans l'objet Presentation des DTM ou l'interface utilisateur de Frame Application à partir de IM-Runtime.xml, afin d'assurer la cohérence et l'acceptation pour l'utilisateur.

13 ProfiSafe

13.1 Motivation

Le but du papier ProfiSafe-UseCase est de compiler un ensemble d'exemples types existants d'automation de sécurité intégrée à partir d'emplacements de fabrication discrets et continus et de dériver une approche systématique commune. Il est recommandé que cette solution proposée soit fondée sur

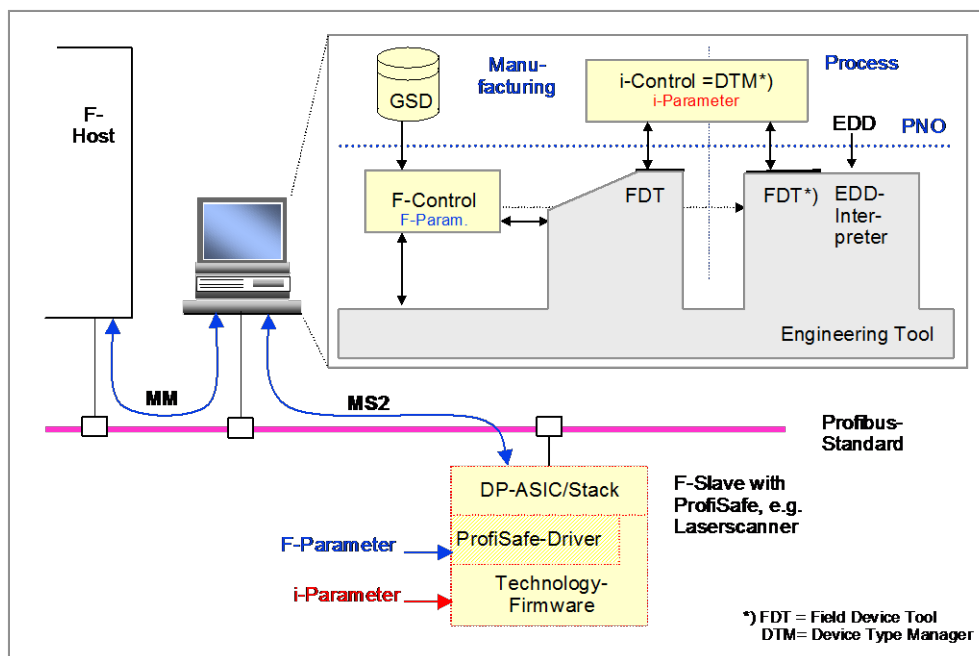
- les principes de communication de Profibus DPV1 et ProfiSafe;
- de nouvelles possibilités d'automates de sécurité intégrée combinés (programmes normalisés et à sécurité intégrée au sein d'un même automate programmable);
- et les activités courantes d'autres groupes de travail Profibus comme FDT/DTM et Proxy FB.

Dans le contexte du papier ProfiSafe-UseCase, la présente spécification accessible au public (PAS) couvre l'aspect partiel "Interface programmeurs FDT pour vérification de paramètres de dispositifs individuels".

13.2 Traitement général des paramètres

Afin de communiquer d'une manière sécurisée, chaque dispositif ProfiSafe (F-Slave) exige les dits F-Parameters (paramètres F) pour le réajustement du mode opérationnel de son pilote ProfiSafe. Ces paramètres comprennent le temps de chien de garde, le niveau d'intégrité de sécurité, la taille de conteneur, etc. Dans le cas des F-slaves (esclaves F) simples, aucun autre paramètre n'est requis. Les F-Parameters seront donc définis dans la GSD Révision 4.0.

En revanche, un F-slave (esclave F) complexe requiert des paramètres pertinents individuels et de sécurité supplémentaires (paramètre de dispositif individuel) qui, en raison de sa taille (>240 octets), ne peuvent pas être transportés par le télégramme de paramétrisation initial (Prm-Telegram). À titre d'amendement, des lignes directrices de ProfiSafe conseillent une méthode supplémentaire par le truchement de blocs fonctionnels proxy dans un automate programmable (PLC) de sécurité, fournissant ainsi une fonctionnalité supplémentaire comme la reparamétrisation commandée par programme, etc.



Légende

IEC 1130/09

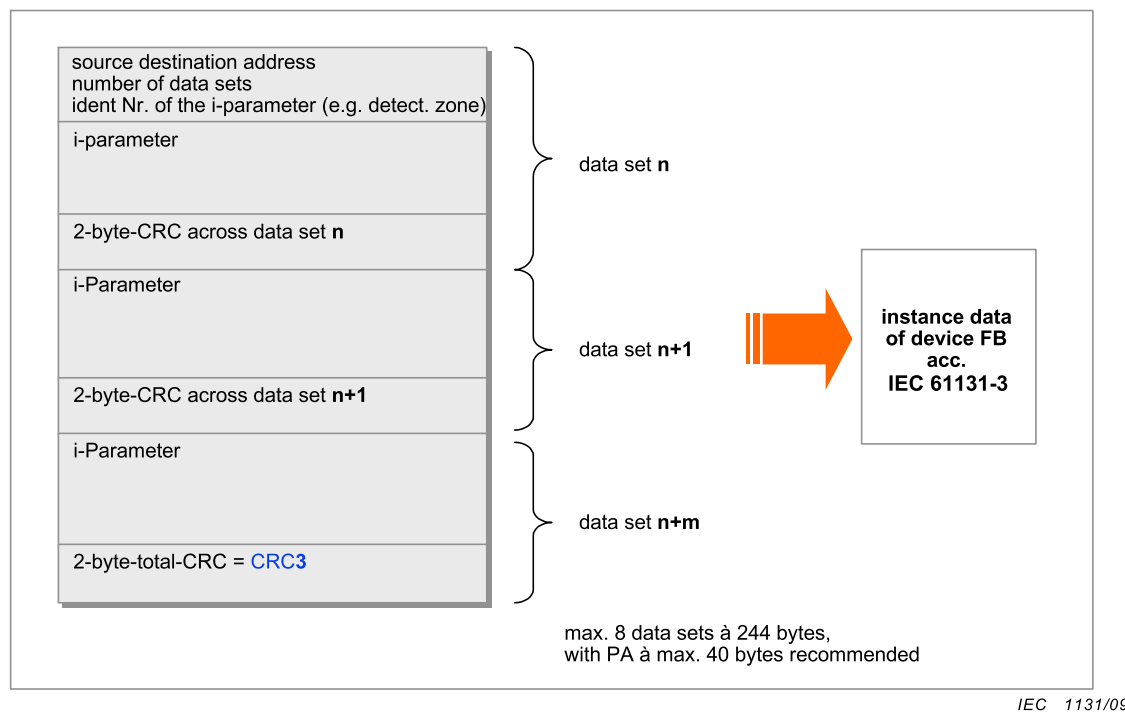
Anglais	Français
F-Host	Hôte F
Manufacturing Process	Processus de fabrication
Process	Processus
i-Control = DTM	Commande I = DTM
i-Parameter	Paramètre I
F-Parameter	Paramètre F
EDD-interpreter	Interpréteur EDD
Engineering Tool	Outil d'étude
Profibus Standard	Norme Profibus
F-Slave with ProfiSafe, e.g. Laserscanner	Esclave F avec ProfiSafe, par exemple: scanner à laser
ProfiSafe-Driver	Pilote ProfiSafe
Technology-Firmware	Micrologiciel («Firmware») Technologie
FDT = Field Device Tool	FDT = Outil de dispositif de terrain
DTM= Device Type Manager	DTM = Gestionnaire de type de dispositifs
DP-ASIC/STACK	DP-ASIC/PILE

Figure 4 – F-Parameter (paramètre F) et paramètre de dispositif individuel

Il est recommandé que les paramètres de dispositifs individuels, qui sont, par nature, spécifiques à un dispositif F, soient traités par les programmes de Gestionnaire de type de dispositif (DTM) provenant des fabricants de dispositifs F. L'outil d'étude fonctionne comme une Frame Application pour un tel DTM d'un esclave F. Il achemine les demandes de communication du DTM vers son dispositif (MS2) et procure une persistance de données. Autrement, il couvre toutes les tâches traditionnelles comme configuration de réseau, paramétrisation, mise en service et diagnostic. Il communique avec l'automate programmable par l'intermédiaire du Master-Master-Protocol (MM, protocole maître-maître). À cet égard, un automate programmable sur le Profibus est un dispositif de sa propre classe («Master Class 1», classe 1 de maître). L'outil d'étude lui-même est défini comme étant un dispositif "Master Class 2" (de classe 2 de maître).

13.3 Paramètre de dispositif individuel ProfiSafe

La structure du paramètre de dispositif individuel ProfiSafe est définie dans les lignes directrices ProfiSafe. Cet ensemble est enveloppé en format XML pour le transport à travers l'interface FDT.



Légende

Anglais	Français
Source destination address	Adresse source destination
Number of data sets	Nombre de jeux de données
Ident. Nr. Of the i-parameter (e.g. detect. Zone)	Numéro d'identification du paramètre I (par exemple: zone de détection)
i-parameter	paramètre i
2 byte-CRC across data set n	CRC de deux octets à travers le jeu de données <i>n</i>
2 byte-total-CRC= CRC3	CRC total deux octets= CRC3
Data set <i>n</i>	Jeu de données <i>n</i>
Max. 8 data sets à 244 bytes, with PA à max. 40 bytes recommended	Max. 8 jeux de données à 244 octets, avec PA à max. 40 octets recommandés
Instance data of FB acc. IEC 61131-3	Données d'instances de FB selon l'IEC 61131-3
Data set <i>n+1</i>	Jeu de données <i>n+1</i>
Data set <i>n+m</i>	Jeu de données <i>n+m</i>

Figure 5 – Structure de données des paramètres de dispositifs individuels ProfiSafe

Les structures de données et les formats de données des paramètres internes de dispositifs individuels sont propriétaires. L'ensemble du jeu est divisé en parties qui s'inscrivent dans un jeu de données Profibus prêt pour le transport. Chaque jeu de données comprend un CRC de deux octets dans le but de vérifier la cohérence des données. Le polynôme est fixe et défini dans les lignes directrices ProfiSafe. Le CRC du dernier jeu vérifie le bloc total.

Bibliographie

- [1] PROFIBUS-PA Profile for Process Control Devices Version 3.0
 - [2] PROFIBUS Profile Guidelines, Part 1, Identification & Maintenance Functions, Version 1.1, May 2003
 - [3] PROFIBUS Guideline (Order No. 2.082): Chapter “Services on communication relationships MSAC_C2”: MSAC2_Initiate
 - [4] PROFIBUS Profile Amendment 3 to the Profibus Profile for Process Control Devices V3.0
 - [5] IEC 61158-6:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 6: Application layer protocol specification*
 - [6] PROFIBUS Specification Specification for PROFIBUS Device Description and Device Integration, Volume 1: GSD Version 5.04, July 2005
 - [7] IEC 61158-5:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 5: Application layer service specification*
 - [8] PROFIBUS Nutzerorganisation e.V., *PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998*
-

FINAL VERSION

VERSION FINALE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope	8
2 Normative references	8
3 Terms, definitions, symbols, abbreviated terms and conventions	9
3.1 Terms and definitions	9
3.2 Symbols and abbreviated terms	9
3.3 Conventions	10
3.3.1 Data type names and references to data types	10
3.3.2 Vocabulary for requirements.....	10
3.3.3 Use of UML.....	10
4 Bus category	10
5 Access to instance and device data	11
5.1 Process Channel objects provided by DTM.....	11
5.2 DTM services to access instance and device data	11
6 Protocol specific behavior	11
6.1 PROFIBUS device model.....	11
6.2 Configuration and parameterization of PROFIBUS devices	12
6.2.1 General	12
6.2.2 Monolithic DTM for a modular PROFIBUS device.....	13
6.2.3 Modular DTM for a modular PROFIBUS device	13
6.3 Support for DPV0 configuration.....	14
6.4 PROFIBUS slaves operating without a cyclic PROFIBUS master	14
6.5 PROFIBUS-related information of a slave DTM	14
6.5.1 General	14
6.5.2 Bus Master Configuration Part (BMCP).....	15
7 Protocol specific usage of general data types.....	25
8 Protocol specific common data types	27
9 Network management data types	27
9.1 General.....	27
9.1.1 Configuration	27
9.1.2 Process Channel.....	28
9.1.3 Parameterization.....	28
9.2 Master-bus parameter set.....	29
9.3 Slave bus parameter set.....	29
9.4 Module and channel data.....	30
9.5 GSD information.....	33
9.5.1 General	33
9.5.2 GSD for gateway devices	33
10 Communication data types.....	34
10.1 General.....	34
10.2 Error information provided by Communication Channel.....	34
10.3 DPV0 communication.....	34
10.4 DPV1 communication.....	41
11 Channel parameter data types	44

12	Device identification	47
12.1	General	47
12.2	Protocol specific handling of the data type STRING	47
12.3	Common device type identification data types	47
12.4	Topology scan data types	52
12.5	Scan identification data types	52
12.6	Device type identification data types – provided by DTM.....	55
12.7	Identification information in GUI	58
13	ProfiSafe	58
13.1	Motivation	58
13.2	General parameter handling.....	58
13.3	ProfiSafe individual device parameter	59
	Bibliography	61
	Figure 1 – Part 303-1 of the IEC 62453 series	7
	Figure 2 – FDT PROFIBUS device model	12
	Figure 3 – Example for IO data within datagrams.....	31
	Figure 4 – F-Parameter and individual device parameter	59
	Figure 5 – Data structure of ProfiSafe individual device parameters	60
	Table 1 – Protocol identifiers.....	10
	Table 2 – Physical layer identifiers	10
	Table 3 – BMCP Part1 – General configuration.....	16
	Table 4 – BMCP Part2 – Parameter data	16
	Table 5 – BMCP Part3 – Configuration data	17
	Table 6 – Part 4: Address table and slave user parameters.....	18
	Table 7 – Part 4: Extended Prm data.....	18
	Table 8 – Complete BMCP	19
	Table 9 – Protocol specific usage of general data types.....	25
	Table 10 – Bus parameter set for master device	29
	Table 11 – Bus parameter set for slave device	30
	Table 12 – Signal channels within the data frame	32
	Table 13 – Simple DPV0 communication data types	35
	Table 14 – Structured DPV0 Communication data types	35
	Table 15 – Availability of services for Master Class1 (C1).....	40
	Table 16 – Availability of services for Master Class2 (C2).....	40
	Table 17 – Simple DPV1 communication data types	41
	Table 18 – Structured DPV1 communication data types	42
	Table 19 – Mapping of DPV1 data types to FDT data types.....	44
	Table 20 – Simple Channel Parameter data types.....	45
	Table 21 – Structured Channel Parameter data types	46
	Table 22 – Identification data types with Profibus DP specific mapping	48
	Table 23 – Identification data types with Profibus I&M specific mapping	49
	Table 24 – Identification data types with Profibus PA specific mapping	51

Table 25 – Simple identification data types with protocol independent semantics	52
Table 26 – Structured identification data types with protocol independent semantics	52
Table 27 – Simple topology scan data types	52
Table 28 – Structured topology scan data types.....	52
Table 29 – Simple scan identification data types.....	53
Table 30 – Structured scan identification data types	53
Table 31 – Structured device identification data types	56
Table 32 – DataLink Layer Identifiers	10

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

DISCLAIMER

This Consolidated version is not an official IEC Standard and has been prepared for user convenience. Only the current versions of the standard and its amendment(s) are to be considered the official documents.

This Consolidated version of IEC 62453-303-1 bears the edition number 1.1. It consists of the first edition (2009-06) [documents 65E/127/FDIS and 65E/140/RVD] and its amendment 1 (2016-06) [documents 65E/336/CDV and 65E/395A/RVC]. The technical content is identical to the base edition and its amendment.

This Final version does not show where the technical content is modified by amendment 1. A separate Redline version with all changes highlighted is available in this publication.

International Standard IEC 62453-303-1 been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

Each part of the IEC 62453-3xy series is intended to be read in conjunction with IEC 62453-2.

The French version of this standard has not been voted upon.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 62453 series, under the general title *Field Device Tool (FDT) interface specification*, can be found on the IEC website.

The committee has decided that the contents of the base publication and its amendment will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The “colour inside” logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.

INTRODUCTION

This part of IEC 62453 is an interface specification for developers of FDT (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

With the integration of fieldbusses into control systems, there are a few other tasks which need to be performed. In addition to fieldbus- and device-specific tools, there is a need to integrate these tools into higher-level system-wide planning- or engineering tools. In particular, for use in extensive and heterogeneous control systems, typically in the area of the process industry, the unambiguous definition of engineering interfaces that are easy to use for all those involved is of great importance.

A device-specific software component, called DTM (Device Type Manager), is supplied by the field device manufacturer with its device. The DTM is integrated into engineering tools via the FDT interfaces defined in this specification. The approach to integration is in general open for all kinds of fieldbusses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

Figure 1 shows how IEC 62453–303-1 is aligned in the structure of the IEC 62453 series.

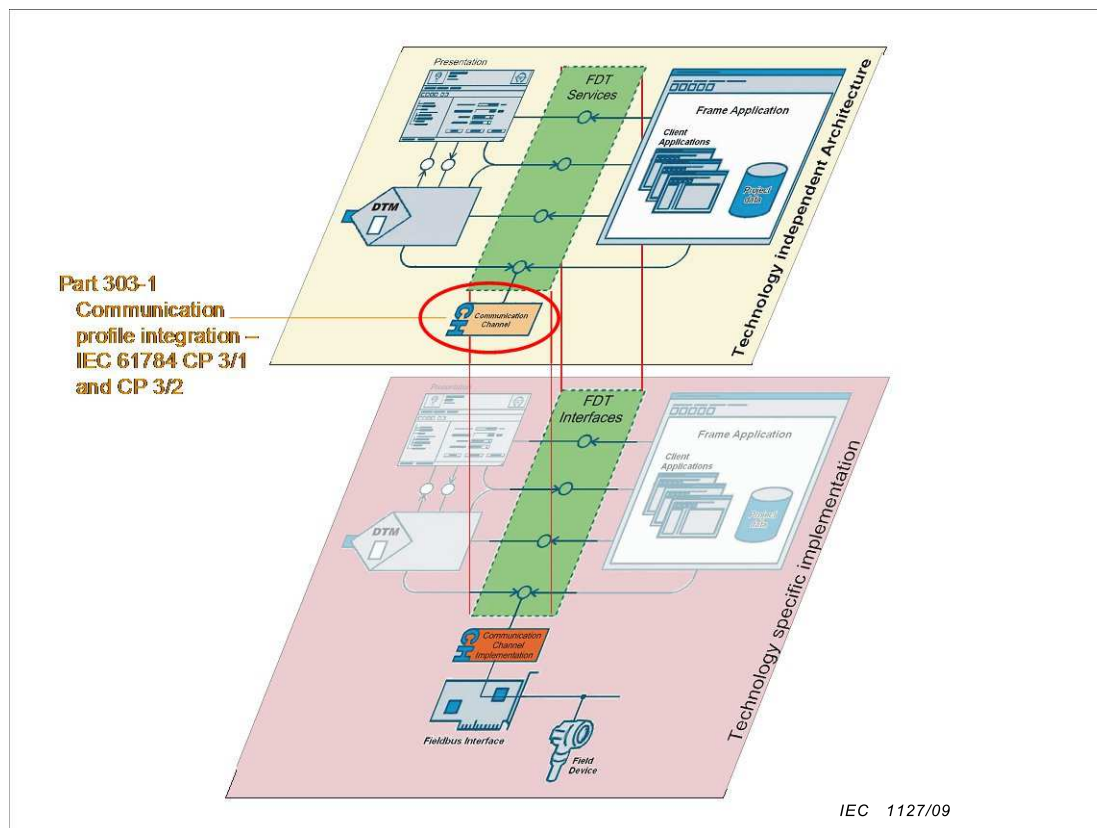


Figure 1 – Part 303-1 of the IEC 62453 series

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2

1 Scope

Communication Profile 3/1 and Communication Profile 3/2 (commonly known as PROFIBUS™¹) defines communication profiles based on IEC 61158-2 Type 3, IEC 61158-3-3, IEC 61158-4-3, IEC 61158-5-3, and IEC 61158-6-3. The basic profiles CP 3/1 (PROFIBUS DP) and CP 3/2 (PROFIBUS PA) are defined in IEC 61784-1.

This part of IEC 62453 provides information for integrating the PROFIBUS protocol into the FDT interface specification (IEC 62453–2).

This part of the IEC 62453 specifies communication and other services.

This specification neither contains the FDT specification nor modifies it.

2 Normative references

The following referenced documents are indispensable for the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*

IEC 61158-2:2014, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-3-3, *Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3 *Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-5-3: *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements*

IEC 61158-6-3, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

¹ PROFIBUS™ is a trade names of the non-profit organization PROFIBUS Nutzerorganisation e.V. (PNO). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for PROFIBUS™. Use of the registered logos for PROFIBUS™ requires permission of PNO.

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62453-1 and IEC 62453-2 apply.

3.1.1

bus interface module

module of a field device that provides the connection to the fieldbus

3.1.2

CP 3/1

Communication profile of CPF3, featuring asynchronous transmission; RS 485 (ANSI TIA/EIA RS-485-A); optional RS 485-IS; plastic fiber; glass multi mode fiber or glass single mode fiber; PCF fiber

3.1.3

CP 3/2

Communication profile of CPF3, featuring synchronous transmission; manchester coded and bus powered (MBP); optional intrinsically safe (MBP-IS) and lower power (MBP-LP)

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviations given in IEC 62453-1, IEC 62453-2 and the following apply.

ANSI	American National Standards Institute (http://www.ansi.org)
BIM	Bus Interface Module
BMCP	Bus Master Configuration Part
CFG	Configuration data used during initialization of PROFIBUS slave device
DCS	Distributed Control System
DP	Decentralized Peripherals
EIA	Electronic Industries Alliance
FDL	Fieldbus Data Link layer
FMA	Fieldbus Management layer
FMS	Fieldbus Message Specification
GSD	General Station Description
MBP	Manchester coded Bus Powered
I&M	Identification and maintenance functions
PA	Process Automation
PCF	Polymer Clad Fibre
PROFIBUS	Process Field Bus
RS	Radio Sector / Recommended Standard
TIA	Telecommunications Industry Association

3.3 Conventions

3.3.1 Data type names and references to data types

The conventions for naming and referencing of data types are explained in IEC 62453-2, Clause A.1

3.3.2 Vocabulary for requirements

The following expressions are used when specifying requirements.

Usage of “shall” or “mandatory”	No exceptions allowed.
Usage of “should” or “recommended”	Strong recommendation. It may make sense in special exceptional cases to differ from the described behaviour.
Usage of “conditional”	Function or behaviour shall be provided, depending on defined conditions.
Usage of “can” or “optional”	Function or behaviour may be provided, depending on defined conditions.

3.3.3 Use of UML

Figures in this document are using the UML notation as defined in Annex A of IEC 62453–1.

4 Bus category

CP 3/1 and CP 3/2 protocols are identified in the protocolId element of the structured data type 'fdt:BusCategory' by the following unique identifiers (Table 1):

Table 1 – Protocol identifiers

Identifier value	ProtocolId name	Description
036D1497-387B-11D4-86E1-00E0987270B9	'Profibus DP/V0'	Support of Profibus DP V0 protocol
036D1499-387B-11D4-86E1-00E0987270B9	'Profibus DP/V1'	Support of Profibus DP V1 protocol

CP 3/1 AND CP 3/2 protocols are using the following unique identifiers in physicalLayer members within PhysicalLayer data type (Table 2):

Table 2 – Physical layer identifiers

Identifier value	Name	Description
036D1590-387B-11D4-86E1-00E0987270B9	MBP	IEC 61158-2 (MBP, Profibus PA)
036D1591-387B-11D4-86E1-00E0987270B9	RS485	IEC 61158-2:2014, Clause 22 (RS485, PROFIBUS DP)
036D1592-387B-11D4-86E1-00E0987270B9	Fiber Optic	IEC 61158-2:2014, Clause 23 (Fiber Optic cable, PROFIBUS DP)
036D1593-387B-11D4-86E1-00E0987270B9		Ethernet (depreciated, do not use)

Table 32 defines which DataLinkLayer shall be used in combination with the BusCategory values defined in Table 32.

Table 32 – DataLink Layer Identifiers

Identifier value	Name	Description
50A21B35-7EE7-4999-8174-70396929C0B4	PROFIBUS FDL	PROFIBUS FDL
CDF338DC-E9A3-4D13-91AC-60A43DCB2904	PROFIBUS FMA1/2	PROFIBUS FMA1/2

5 Access to instance and device data

5.1 Process Channel objects provided by DTM

The minimum set of provided data should be: Process values modeled as channel objects including the ranges and scaling.

5.2 DTM services to access instance and device data

The services InstanceDataInformation and DeviceDataInformation shall provide access to at least all parameters of the Physical Block and the status and Out value of the Function Blocks shall be exposed.

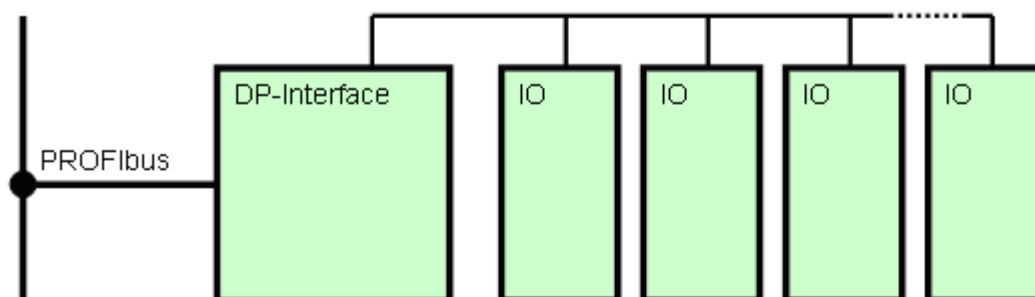
According to IEC 62453-2, at least one set of semantic information (one per supported fieldbus protocol) shall be provided for each accessible data object, using the 'SemanticInformation' general data type. The corresponding data type 'applicationDomain' shall have a value defined for Profibus and the data type 'semanticId' shall have an appropriate value, as specified in Table 9.

6 Protocol specific behavior

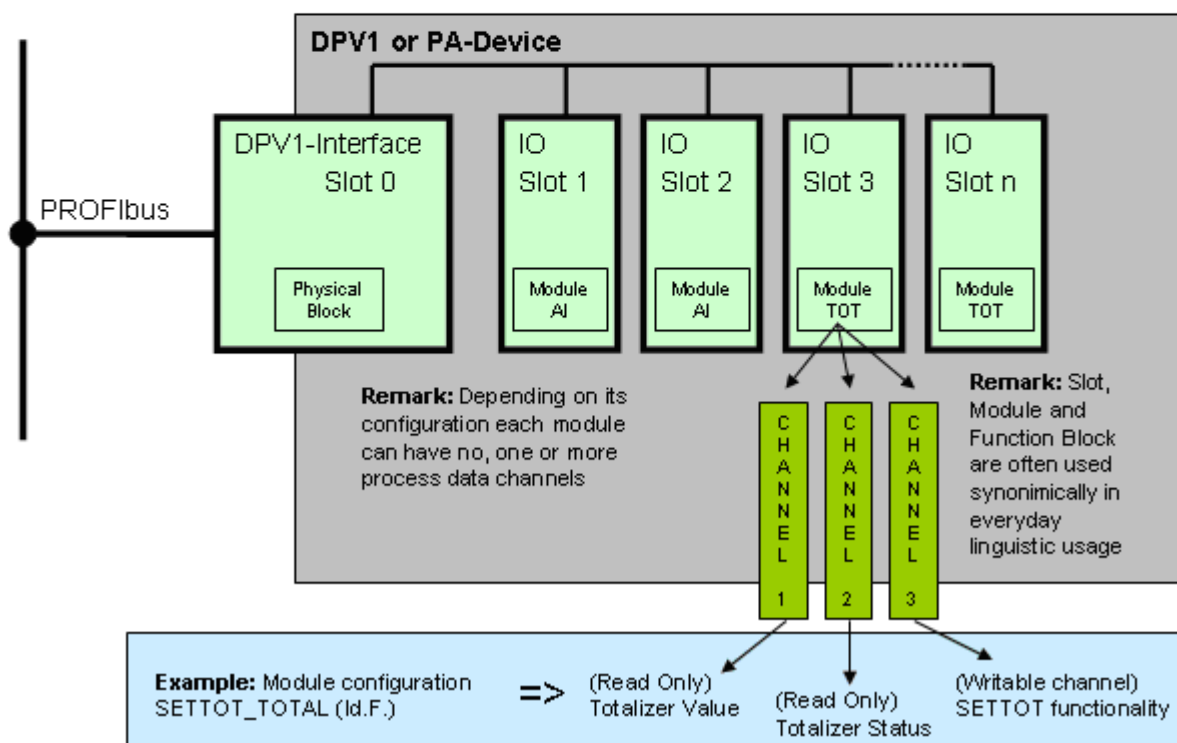
6.1 PROFIBUS device model

FDT extends the PROFIBUS device model by using Process Channels for description of I/O values (see Figure 2).

Classical View of PROFIBUS device



PROFIBUS notations from a monolithic DPV1 or PA device DTMs point of view



IEC 1128/09

Figure 2 – FDT PROFIBUS device model

6.2 Configuration and parameterization of PROFIBUS devices

6.2.1 General

In a GSD-based configuration tool the user defines the configuration and sets the appropriate parameters for the modules. The configuration tool creates the configuration string and the parameter string that are used to set the slave properly.

With FDT the configuration and parameterization of the devices is no longer executed only by a central component; it moved partly into the DTMs. A DTM is responsible for providing configuration and parameterization information for the cyclic master that puts the PROFIBUS slaves in operation.

A DTM is used to adjust a field device to its specific application. Within PROFIBUS, there are three different aspects of adjustment:

- parameterization: usr prm data (used in the PROFIBUS service SET_PRM for setting up the cyclic communication and the specific behavior of the device);

- application parameterization: application specific parameters (transmitted via acyclic read/write PROFIBUS services);
- configuration: configuration data (used in the PROFIBUS service CHK_CFG for definition of the format and length of the input/output data that are transmitted within cyclic communication).

The application parameterization transmitted via acyclic communication is not in the scope of this document. Within this document the term parameterization represents communication parameterization (SET_PRM).

6.2.2 Monolithic DTM for a modular PROFIBUS device

A monolithic DTM is one single DTM that represents the complete device with its Bus Interface Module (BIM) and its I/O modules. In general, such a DTM offers a configuration dialog (presentation object) that allows definition of the used BIM and modules. The configuration dialog must be available via the FDT standard function “Configure” (see [1] 4.3 Operation Configuration).

Not all PROFIBUS devices require a configuration dialog. That is why not all DTMs provide the “Configure” function. This is valid only for non-modular PROFIBUS devices if the Usr_Prm-Data cannot be changed.

The configuration dialog allows changing the data only in offline mode if the data set can be locked.

6.2.3 Modular DTM for a modular PROFIBUS device

Separate DTMs represent the BIM (Device DTM) and the particular I/O modules (Module DTMs). The effort developing such a modular DTM is normally higher than in the case of a monolithic DTM, because:

- a private protocol has to be implemented between BIM and I/O modules to ensure that only a Module DTM can be added to the BIM DTM. This requires an own protocol ID and the adaptation / creation of communication;
- in some cases, additional private interfaces are necessary to exchange information between Device DTM for BIM and Module DTMs.

Implementing a Modular DTM results in the following advantages:

- the project represents the device structure;
- the user is able to access module-related information directly as a function of the Module DTM;
- the FDT specification defines a mechanism to identify DTMs. With these mechanisms it is possible to provide support for scanning the modules below the BIM and generate the topology automatically;
- supporting a new type of BIM or I/O module requires an additional DTM “only” and does not affect existing components. This may result in reduced test effort.

The configuration data to set up the PROFIBUS configuration must be provided by the Device DTM (representing the BIM). This configuration data may be generated from information of the instantiated Child DTMs and by using a configuration dialog.

Modular DTMs should be provided for modular devices (e.g. a plant operator may add/remove modules). Monolithic DTMs are used to represent devices that show no modularity (e.g. PA devices).

6.3 Support for DPV0 configuration

A PROFIBUS slave is configured by a cyclic master and communicates via PROFIBUS DP. In addition to this the slave may support DPV1 communication.

A Gateway DTM for a PROFIBUS slave does not have to provide communication for the DPV0 communication schema. For example, there is a remote I/O system with HART modules. It may have a Gateway DTM that requires the DPV1 protocol and provides the HART protocol (defined in the information document and in the parameter document). This enables HART Device DTMs to communicate with their devices via the Gateway DTM and via Communication DTM for DPV1. Following the specification the Gateway DTM delivers channel parameter documents for both protocols DPV1 and HART. The protocolId is a member of NetworkManagementInfo data type.

The Process Channels shall provide Channel Parameter documents for DPV1 including all information to allow integration into the control system (e.g. DPAddress of the IO value if available).

6.4 PROFIBUS slaves operating without a cyclic PROFIBUS master

In most cases, a PROFIBUS slave is configured and parameterized by a cyclic PROFIBUS master device. So a running master device in the network is required.

Some slaves are able to allow acyclic communication without a running cyclic master. Especially in the case of gateway functionality this is an eminent advantage because they allow the parameterization of field devices connected to them by using an acyclic bus master. So instrument specialists are able to work with field devices also in case the controller is not yet working.

If a master starts communication, these devices start to detect bus speed and settings to react properly. This may take some time.

In the following, two cases are described that a user may keep in mind when working with such devices.

Use case 1:

The user performs a network scan. The Communication DTM tries to read diagnostic data via a GetDiagnose Request but does not receive a response. The device is not detected by the Communication DTM. This occurs mostly when the device has a low PROFIBUS address. The reason is that the device has not completed bus speed / bus setting detection as it was asked for their diagnostic data. The workaround is to give these devices a higher PROFIBUS address.

Use Cases 2:

The user tries to connect a field device linked to the gateway that supports DPV1 without a running cyclic master. This can lead to an error message because the gateway device has not completed bus speed / bus setting detection as it was asked for a connection. So the user has to try to connect again. This happens only in very rare situations.

6.5 PROFIBUS-related information of a slave DTM

6.5.1 General

The information used by a cyclic master device to set up the PROFIBUS network properly and allow cyclic communication between control system and slave devices is provided by a DTM in

- Bus Master Configuration Part (BMCP);
- GSD information;
- internal topology,
- process channels.

A DTM of a PROFIBUS slave must deliver these parts of PROFIBUS-related information to get integrated into a FDT-based Engineering System. In the next subclauses, a more detailed description is given on how to generate and how to provide this information. This depends on the kind of DTM (see 6.2 Configuration and parameterization of PROFIBUS devices).

6.5.2 Bus Master Configuration Part (BMCP)

6.5.2.1 BMCP introduction

The BMCP of one single DTM instance describes the actual parameter and configuration data of the corresponding PROFIBUS slave. Each DTM representing a PROFIBUS slave device must provide a Bus Master Configuration Part. The BMCP is provided in the `busMasterConfigurationPart` member of `NetworkManagementInfo` data type. This information is obtained by calling service `NetworkManagementInfoRead`.

The BMCP includes information about the configuration and the parameters for the slave. The BMCP is provided by the DTM and is required in order to generate the master configuration.

The BMCP contains data which might be changed during master configuration. That means that the BMCP may be changed and transferred back to the slave DTM by calling `NetworkManagementInfoWrite`. A Slave DTM must accept the new information and recompute the configuration / internal parameters to match the new BMCP.

DTM must check whether the new values are according to the capabilities of the device. The `NetworkManagementInfoWrite` call will be refused if the device can not handle the new values.

6.5.2.2 Creating the BMCP

This subclause explains the meaning of the individual elements of the BMCP in details.

The BCMP may be generated from the GSD information of a PROFIBUS device.

The BMCP is divided into four parts that are explained in the following subclauses.

The explanations use the GSD keywords and reference the PROFIBUS specification. See also the table with complete BMCP in 6.5.2.2.1 .

6.5.2.2.1 Part 1: From Slave_Para_Len to Octet 15

The first part consists of a fixed set of bytes described in the table below.

Table 3 – BMCP Part1 – General configuration

Byte	Name	Defined in	Notice
0	Slave_Para_Len	[5] 6.2.12.1	Length of the BMCP including this value
1			
2	SI_Flag	[5] 6.2.12.2	The following GSD values are used [6]: • Bit 0 : reserved, • Bit 1: Extra_Alarm_SAP, • Bit 2: DPV1_Data_Types, • Bit 3: DPV1_Supported, • Bit 4: Publisher_Enable, • Bit 5 : Fail_Safe, . The following bits are not based on GSD values: • Bit 6 : New_Prm, • Bit 7 : Active
3	Slave Type	[5] 6.2.12.3	Value is 0 (= DP-Slave)
4	Max_Diag_Data_Len	[5] 6.2.12.4	The following GSD value is used [6]: Max_Diag_Data_Len
5	Max_Alarm_Len	[5] 6.2.12.5	
6	Max_Channel_Data_Len	[5] 6.2.12.6	This field defines how much data can be transferred between slave and master. In this case, the maximum of these GSD values [6] must be calculated: • Max_Data_Len, • C1_Max_Data_Len plus 4 Bytes (Function Num, Slot_Number, Length)
7	Diag_Update_Delay	[5] 6.2.12.7	The following GSD value is used [6]:Diag_Update_Delay
8	Alarm_Mode	[5] 6.2.12.8	
9	Add_SI_Flag	[5] 6.2.12.9	
10	C1_Timeout	[5] 6.2.12.10	The following GSD value is used [6]:C1_Response_Timeout
11			
12	Reserved		
13			
14			
15			

6.5.2.2.2 Part 2: Parameter data

Part 2 of the BMCP are the slave parameter data.

Table 4 – BMCP Part2 – Parameter data

Byte	Name	Defined in	Notice
16	Prm_Data_Len	[5] 6.2.12.11	Length of the Parameter Data including this value
17			
18	Station Status	[5] 6.2.4.1	The following GSD values are used [6]: • Bit 0-2: reserved, • Bit 3: WD_On, • Bit 4: Freeze_Req, • Bit 5: Sync_Req, • Bit 6,7(Lock/Unlock Request): [5] Table 402
19	WatchDog1	[5] 6.2.4.2	These values (WD_Fact_1 and WD_Fact_2) depend on the baud rate. A master should set these values and slaves should handle new values
20	WatchDog2	[5] 6.2.4.3	
21	Min Tsdr	[5] 6.2.4.4	Default value is 11 bit times [1]
22	Ident_Number	[5] 6.2.3.5	The following GSD value is used [6]: Ident_Number

Byte	Name	Defined in	Notice
23			
24	Group_Ident	[5] 6.2.4.5	Indicates the group assignment of the slave in a bitwise coded form
25	DPV1_Status_1	[5] 6.2.4.7	The following GSD values are used [6]: • Bit 0,1: reserved, • Bit 2: WD_Base_1ms, • Bit 3 – 5: reserved, • Bit 6: Fail_Safe, • Bit 7 : DPV1_Enable, These bits report the slave capabilities to the master and are changed by the MS0 master following its capabilities
26	DPV1_Status_2	[5] 6.2.4.8	The following GSD values are used [6]: • Bit 0: Check_Cfg_Mode, • Bit 1: reserved, • Bit 2: Enable_Update_Alarm, • Bit 3: Enable_Status_Alarm, • Bit 4: Enable_Manufacturer_Specific_Alarm, • Bit 5: Enable_Diagnostic_Alarm, • Bit 6: Enable_Process_Alarm, • Bit 7: Enable_Pull_Plug_Alarm, These bits report the slave capabilities to the master and are changed by the MS0 master following its capabilities
27	DPV1_Status_3	[5] 6.2.4.9	• Bit 0-2: - , • Bit 3: Prm_Structure, • Bit 4: IsoM_Req, • Bit 5-6: reserved, • Bit 7: Prm_Cmd
28	User_Prm_Data		From Byte 28 of the BMCP, a DTM has to insert additional user parameter data. The information is given in the GSD file via value of User_Prm_Data
...			
NOTE 1 It is possible that a module does not have user parameter data. In this case, no parameter string is inserted into the BMCP for this module. NOTE 2 Some slaves have some fixed modules besides the BIM. Even if these modules do not appear in the configuration dialog or are not represented as Module DTMs, they have to be considered in the BCMP if they have parameters. NOTE 3 The three DPV1 status bytes are defined by the MS0 master.			

6.5.2.2.3 Part 3: Configuration data

The configuration data are provided as part 3:

Table 5 – BMCP Part3 – Configuration data

Byte	Name	Source	Notice
16+Prm_Data_Len	Cfg_Data_Len	[5] 6.2.12.13	Length including this value
16+Prm_Data_Len + 1			
16+Prm_Data_Len +2	Cfg_Data		Configuration data (if available)
...			

From byte position (16+Prm_Data_Len +2), the configuration strings for the BIM and the modules are provided in ascending order. The information is given in the GSD file via value module.

NOTE 1 Some slaves have some fixed modules besides the BIM. Even if these modules do not appear in the configuration dialog or are not represented as Module DTMs, they have to be considered in the BCMP if they have parameters.

NOTE 2 Empty slots have to be considered in the configuration data. Refer to the GSD file as to which configuration string has to be used.

6.5.2.2.4 Part 4: Address table and slave user parameters

In Part 4 of the BMCP, the address table and the slave user parameter section is provided.

Table 6 – Part 4: Address table and slave user parameters

Byte	Name	Source	Hints
16 + Prm_Data_Len + Cfg_Data_Len	Add_Tab_Len	[5] 6.2.12.15	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len	Slave_User_Data_Len	[6] 6.2.12.21	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 2	Slave_User_Data		Slave user parameter (if available)
...			

After Add_Tab_Len the address table data is inserted into the BMCP. If the master does not support this table the Add_Tab_Len is set to 2 and no data is inserted.

After Slave_UserDataLen the slave user data is inserted into the BMCP. If the master does not support this data the Slave_User_Data_Len is set to 2 and no data is inserted.

Table 7 – Part 4: Extended Prm data

Byte	Name	Source	Hints
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len	Ext_Prm_Data_Len	[5] 6.2.12.23	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len + 1			

After Ext_Prm_Data_Len, the extended prm data is inserted into the BMCP. If the master does not support this data the Ext_Prm_Data_Len is set to 2 and no data is inserted.

6.5.2.2.5 Complete slave bus parameter set

All values are provided from the Slave DTM. It is responsibility of the Slave DTM to be compatible with the Slave GSD. The Master DTM could change some of these initial values sent by Slave DTM if they depend on the capabilities of the master.

Example

Within the GSD file, it is stated that the device supports the Freeze Mode by the keyword "Freeze_Mode_supp". The master sets the value "Freeze_Mode_Req" within the Slave Bus Parameter Set because only the master knows whether it supports this mode.

The following table explains which component is the source of the parameter values ("Parameter provided by"). Some of the values can be changed by the system or by user interaction. For those values it is indicated which component is allowed to change the values ("Configuration by"). If possible, the default values for the parameters are defined ("Default Value").

Table 8 – Complete BMCP

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
0	Slave_Para_Len	Length of the BMCP including this value	Calculated by Slave DTM	-	-
1					
2	SI_Flag	Bit coded. The meaning of the bits:	-	-	-
		Bit 0: reserved	-	-	0
		Bit 1: Extra_Alarm_SAP	Slave DTM shows with this bit that the feature is supported	Master DTM	Extra_Alarm_SAP_sup p within GSD
		Bit 2: DPV1_Data_Types	Slave DTM shows what data types are used by the slave DTM	Master DTM	DPV1_Data_Types within Slave GSD
		Bit 3: DPV1_Supported	Slave DTM shows with this bit that the feature is supported	Master DTM	DPV1_Slave within GSD
		Bit 4: Publisher_enable	Slave DTM shows with this bit that the feature is supported	Master DTM	Publisher_sup within GSD
		Bit 5: Fail_Safe	Slave DTM can show by this flag that the slave needs fail safe (Fail_Safe or Fail_Safe_Required of the GSD)	Master DTM	Value within Slave GSD
		Bit 6: New_Prm	If the Slave DTM set this value, it requests that the MS0 master should set the changed slave parameter. (see [5] 6.2.12.2)	-	0
		Bit 7: Active	If the Slave DTM has finished the module configuration set it to 1.	-	0
3	Slave Type	0 = DP-Slave	Slave DTM shows that it is a DP-Slave	-	0
4	Max_Diag_Data_Len		Slave DTM provides this information by Max_Diag_Data_Len in the GSD	-	Value within Slave GSD
5	Max_Alarm_Len	Length of the alarm structure see [5] table 396	Slave DTM, Conditions: One of GSD keywords:- Diagnostic_Alarm_sup p - Process_Alarm_sup - Pull_Plug_Alarm_sup - Status_Alarm_sup - Updata_Alarm_sup - Manufacturer_Specific_Alarm_sup OR - Alarm_Type_Mode_sup	-	Is calculated on the base of the different GSD values.
6	Max_Channel_Data_Len		Slave DTM: This field defines how much data can be transferred between slave and master. In this case the maximum of these values must be calculated and set by the slave DTM. Rule for calculation: Max_Data_Len C1_Max_Data_Len plus 4 Bytes (Function Num, Slot_Number, Length)	-	Value within Slave GSD
7	Diag_Update_Delay		Slave DTM	-	Slave GSD [6] Diag_Update_Delay

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
8	Alarm_Mode		Slave DTM	-	Slave GSD [6] Alarm_Sequence_Mode_Count
9	Add_SI_Flag	Bit coded. Following the meaning of the bits			
		Bit 0: NA_To_Abort		Master DTM	0
		Bit 1: Ignore_Aclr		Master DTM	0
		Bit 2 to Bit 7: reserved			0
10	C1_Timeout	The following GSD value is used [6]: C1_Response_Timeout	Slave DTM		Slave GSD [6] C1_Response_Timeout
11					
12	Reserved				0
13	Reserved				0
14	Reserved				0
15	Reserved				0
16	Prm_Data_Len	Length of the Parameter Data including this value	Calculated (or maybe fixed) by Slave DTM		
17					
	PRM_DATA				
18	Prm_Data (Station Status)	Bit 0-2: reserved		0	
		Bit 3: WD_On	MS0 Master defines that watchdog is used or not. If watchdog is enable the master has also to set WD_Fact_1 and WD_Fact_2	Master DTM	0
		Bit 4: Freeze_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	Freeze_Mode_supp within GSD
		Bit 5: Sync_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	(Sync_Mode_supp within GSD)
		Bit 6,7: Lock/Unlock Request [5] Table 402	Master DTM	0	
19	Prm_Data (WatchDog1) WD_Fact_1	These values depend on the baud rate. A master should set these values and slaves should handle new values. Consider within Usr_Prm_Data (DPV1_Status_1, Bit 2 WD_Base_1ms)		Master DTM	1

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
20	Prm_Data (WatchDog2) WD_Fact_2	See Byte 19, Watchdog_Time = 10 ms * WD_Fact_1 * WD_Fact_2 OR Watchdog_Time = 1 ms * WD_Fact_1 * WD_Fact_2		Master DTM	1
21	Prm_Data (Min Tsdr)	Default value is 11 bit times [1].		Master DTM	11
22	Prm_Data (Ident_Number)		SlaveDTM	-	Slave GSD [6] Ident_Number
24	Prm_Data (Group_Ident)	This means the slave is not assigned to a group for global control. It is set by the master		Master DTM	0
25	Prm_Data (DPV1_Status_1)	Bit 0,1: reserved			0
		Bit 2: WD_Base_1ms (See Bytes 19 and 20)	Slave DTM shows with this bit that the feature is supported	Master DTM In some rare case there is a need to set the value to 1	WD_Base_1ms_supp within GSD
		Bit 3 – 5: reserved			0
		Bit 6: Fail_Safe	Slave DTM shows with this bit that the feature is supported	Master DTM	(Fail_safe within GSD)
		Bit 7: DPV1 Enable	Slave DTM	Master DTM	Slave GSD (If the GSD minimum include one of these setting : C1_Read_Write_supp = 1 or Diagnostic_Alarm_sup p = 1 or Process_Alarm_supp = 1 or Pull_Plug_Alarm_supp = 1 or Status_Alarm_supp = 1 or Update_Alarm_supp = 1 or Manufacturer_Specific _Alarm_supp = 1 . The slave supports DPV1 and then the DPV1_Enable should be 1

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
26	Prm_Data (DPV1_Status_2)	Bit 0 : Check_Cfg_Mode			
	Slave DTM shows with this bit that the feature is supported.	Master DTM	Check_Cfg_Mode within GSD		
		Bit 1 : reserved	-	-	0
		Bit 2 : Enable_Update_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Update_Alarm_supp within GSD
		Bit 3 : Enable_Status_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Status_Alarm_supp within GSD
		Bit 4: Enable_Manufacturer_Specific_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Manufacturer_Specific_Alarm_supp within GSD
		Bit 5: Enable_Diagnostic_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Diagnostic_Alarm_supp within GSD
		Bit 6: Enable_Process_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Process_Alarm_supp within GSD
		Bit 7: Enable_Pull_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Pull_Plug_Alarm_supp within Slave GSD
27	Prm_Data (DPV1_Status_3)	Bit 0-2: Alarm_Mode	Slave DTM	Master DTM	Data within Slave GSD
		Bit 3: Prm_Structure.	Slave DTM shows with this bit that the feature is supported		Prm_Block_Structure_supp within GSD
		Bit 4: IsoM_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	Isochron_Mode_supp within GSD
		Bit 5-6: reserved	-	-	0
		Bit 7: Prm_Cmd	Slave DTM shows with this bit that the feature is supported	Master DTM	PrmCmd_supp within GSD
28 to ...	Prm_Data(Usr_Prm_Data)		Calculated by Slave DTM	Slave DTM	Data within GSD
X1	Cfg_Data_Len	Length including this value	Calculated by Slave DTM	-	Data within GSD
X1+1					
X1+2 to ...	Cfg_Data	Cfg_Data	Slave DTM, Depending on Module Configuration	Slave DTM	Data within Slave GSD (Module, EndModule)
X2	Add_Tab_Len	Length including this value	Calculated by the Communication DTM		
X2 + 1					

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
X2 + 2 to	Add_Tab_Data	Address assignment Table (only for DPV0 Masters)	Calculated by the Communication DTM		
X3	Slave_User_Data_Len	Length including this value	Calculated by the Communication DTM		
X3 + 1					
X3 + 2 to ...	Slave_User_Data	Slave_User_Data	Calculated by the Communication DTM		Data within GSD
X4 + 1	Ext_Prm_Data_Len	Length including this value	Calculated by Slave DTM	-	
X4 + 2					
X4	Ext_Prm_Data	Extended Parameter Data	Slave DTM	-	Data within GSD
NOTE X1 = 16+Prm_Data_Len X2 = 16 + Prm_Data_Len + Cfg_Data_Len = 16 + x1 + Cfg_Data_Len X3 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len = X2 + Add_Tab_Len X4 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len = X3 + Slave_User_Data_Len					

6.5.2.3 Modification of the BMCP

6.5.2.3.1 Propagation of changes

The BMCP includes parameter and configuration data. The slave DTM or the FDT Frame Application may modify the BMCP.

The system has to ensure that the Communication DTM representing the PROFIBUS master is aware of these changes. This is achieved by sending the event `InstanceDataChanged` from the Slave DTM to report the change of the BMCP. All changes should be reported as soon as possible, but not before the changes are persistent. The Frame Application informs the Parent DTM via `OnChildInstanceDataChanged`, that parameters of a child have been changed. Then the Communication DTM can get the new BCMP of the Slave DTM.

6.5.2.3.2 When it is allowed to change the BMCP

According to the FDT specification is allowed to change the parameters of a DTM starting from “configured” state (see IEC 62453-2).

The BCMP can be changed multiple times, but only if the DTM is in offline mode and if the data set can be locked.

If a Slave DTM wants to change parameters in online mode it must use DPV0 or DPV1 transaction requests. If there is no way of changing the parameters by transaction requests, the DTM must disable configuration and parameterization in the online state.

6.5.2.3.3 Parameter data

If the user changes parameters of the BIM or one module (for example via user interface) and this affects the BMCP; the DTM has to update the BMCP. In addition to this, it must request a save and inform the Frame Application via `InstanceDataChanged`. The Frame Application must distribute the information to all relevant components.

6.5.2.3.4 Configuration data

Configuration data will change every time if the user adds/removes modules or changes the module type, etc.

In the case of the modular DTM, the BIM will be informed when the user adds or removes modules via the service `ChildAdded` and service `ChildRemoved`. Changes of the parameters in module will be reported by service `OnChildInstanceDataChanged`.

The monolithic DTM or the BIM DTM update their BMCP, request a save and inform the Frame Application via `InstanceDataChanged`.

NOTE 1 Changes that affect the BMCP often take effect on the internal topology and the Process Channels. This information must be updated by the DTMs too.

NOTE 2 The BMCP may be changed by the Slave DTM and by the Communication DTM. The Communication DTM must not change the configuration data and the user parameters.

6.5.2.4 Special cases related to the BMCP

6.5.2.4.1 DPV1 support

In the GSD file there are two flags regarding DPV1. At first, the “DPV1_Slave” value: This means that the slave has the possibility to work as a DPV1 slave. If this value exists and the value is “1” then the respective bit within the `SI_Flag` of Slave Bus Parameter Set must be set. For older systems, there should also be the possibility to work as a DPV0 slave.

Only the Communication DTM knows that its master device is able to provide acyclic services.

After building the Slave Bus Parameter Set, the Communication DTM will receive the slave's initial BMCP. If, within the SI_Flag, the respective bit is set, the Master DTM must set the highest bit in the first byte of Extended DPV1 Status. Now the slave works as a DPV1 slave.

6.5.2.4.2 Extended DPV1 status

All the bits in the 3 bytes Extended DPV1 Status are set by the Communication DTM for the master. A Slave DTM must accept these settings and adapt its functionality if necessary.

7 Protocol specific usage of general data types

The following table (Table 9) explains how data types, defined in IEC 62453–2 within the namespace 'fdt', are used with CP 3/1 AND CP 3/2 protocols.

Table 9 – Protocol specific usage of general data types

Data type	Description for use in IEC 61784 CPF 3
fdt:address	Profibus Parameter Address: The attribute 'address' follows the different device models that are defined for PROFIBUS devices. FDT currently supports the following models: PROFIBUS DP / DPV1, PROFIBUS PA, PROFIdrive (greater or equal profile version 3) PROFIBUS DP / DPV1 The device model is based on devices that are composed of slots, whereas slots do not have to represent physical objects. The data that is contained in the slots, are addressable via Indexes. This data may be variables or composed blocks of data. The address attribute is APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0' PROFIBUS PA The device is represented by a device management structure and a number of blocks that provide different functionality (physical block, function block, transducer block). The blocks are mapped to slot addresses, but this mapping may vary depending on the device type. The address attribute is APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0' PROFIdrive According to the PROFIdrive profile, a device (drive unit) may be composed by a number (1-many) of drive objects (DOs). The DOs may have different type. Each DO is uniquely identifiable and manages its own parameters. Each parameter can be uniquely identified by its number (PNU). Each DO has its own number space. A parameter may contain simple data or composed data (e.g. arrays). The data of the device are accessible via a parameter channel (normally slot 0 index 47). The address attribute is APIxxSLOTyyINDEXzz.DODO-id.pnu xx API yy Slot zz Index

Data type	Description for use in IEC 61784 CPF 3
	do-id Drive Object ID pnu ParameterNumber xx, yy, zz, do-id, pnu are based on decimal format without leading '0'
fdt:protocolId	See Clause 4
fdt:deviceTypeId	The element "fdt:deviceTypeId" shall contain the IDENT_NUMBER of the supported physical device. The IDENT_NUMBER shall be entered in decimal format; however, the value should be displayed as hex to the user
fdt:deviceTypeInfo	A PROFIBUS device has to provide its GSD information as human readable string at this attribute. NOTE The GSD information is accessible via the services NetworkManagementInfoRead GetTypeInfo
fdt:deviceTypeInfoPath	Path to the file containing the information which is provided via the attribute 'deviceTypeInfo'. In case of PROFIBUS the attribute contains the full path to the GSD file including the file name. The file name depends on the current locale according to the usage of service SetLanguage. For PROFIBUS devices it is mandatory to provide this attribute. Examples English: 'C:\MyFolder\ABCD.GSE' German: 'C:\MyFolder\ABCD.GSG'
fdt:manufacturerId	Manufacturer according to Profile specification. For example in Profibus PA : Physical Block Index 10 : DEVICE_MAN_ID
fdt:semanticId fdt:applicationDomain	The SemanticIDs for PROFIBUS follow the different device models that are defined for PROFIBUS devices. FDT currently supports the following models: PROFIBUS DP, PROFIBUS PA, PROFIdrive. PROFIBUS PA The applicationDomain is: FDT_PROFIBUS_PA The device is represented by a device management structure and a number of blocks that provide different functionality (physical block, function block, transducer block). The blocks are mapped to slot addresses, but this mapping may vary depending on the device type. Since the device model is based on blocks, the SemanticIds also are based on the block model. Within each block, the data is identifiable by names of parameters. The semanticId for PROFIBUS profile related parameter follows the following rules: - the semanticId shall be built based on the names defined in the profiles - structured parameters shall be combined with a '.'; - spaces within the profile definition shall be exchanged with an underscore; - blocks shall be counted according to the Object Dictionary; - the block number shall be part of the semanticId. The semanticId is: BlockType.BlockIndex.NameOfParameter.AttributeOfParameter Example AnalogInputFB.3.OUT.Unit PROFIdrive The applicationDomain is: FDT_PROFIBUS_PROFIDRIVE According to the PROFIdrive profile, a device (drive unit) may be composed by a number (1-many) of drive objects (DOs). The DOs may have different types. Each DO is uniquely identifiable and manages its own parameters. Each parameter can be uniquely identified by its number (PNU). Each DO has its own number space. A parameter may contain simple data or composed data (e.g. arrays). The data of the device are accessible via a parameter channel (normally slot

Data type	Description for use in IEC 61784 CPF 3
	0 index 47). The semanticId is: DOdo-id.PNUpnu do-id Drive Object ID pnu ParameterNumber do-id, pnu are based on decimal format without leading '0' Example DO3.PNU64 PROFIBUS DPV1, The applicationDomain is: FDT_PROFIBUS_DPV1 The device model is based on devices that are composed of slots, whereas slots do not have to represent physical objects. The data that is contained in the slots are addressable via Indexes. This data may be variables or composed blocks of data. The semanticID for devices not based on a profile is directly based on the PROFIBUS address information: The semanticId is: APIxx.SLOTyy.INDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0'
fdt:subDeviceType	Manufacturer specific value

8 Protocol specific common data types

Not applicable.

9 Network management data types

9.1 General

The data types specified in this subclause are used at following services:

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

The data type net:DeviceAddress (defined in IEC 62453–2) is used for defining the network address of a device.

The protocol specific data types are based on definitions given in the IEC 61784 and IEC 61158 specifications. Furthermore, they contain additional information about the device that is needed by systems to configure CPF 3 links and to establish communication between the CPF 3 master device and the CPF 3 slave devices.

9.1.1 Configuration

The configuration of the device itself is done with the aid of the DTM's GUI. Downloading the configuration into the slave device is performed via the CPF 3 master device. To do that and in order to set up the bus communication, the master needs information from the DTM as there is:

- GSD file
The GSD information is type-specific information and not instance-specific. It is not stored with single slave instances or in a global accessible file. It is provided by the DTM at the

service `GetTypeInformation`. On this service, a DTM of a PROFIBUS device provides the GSD information within its XML document.

The master device can use the general type-specific information from the slave's GSD information like bus timing parameters, supported baud rates, etc.

- **CFG-String (Cfg_Data)**

The CFG-String provides the instance-specific information about the current configuration of the device. It defines the structure of the data frames that will be transmitted on the PROFIBUS. This structure depends on the modules that are actually configured.

The DTM provides the CFG-String within the attribute `busMasterConfigurationPart` that is part of the XML document available via service `InstanceDataRead`. The structure of the `busMasterConfigurationPart` is defined according to the PROFIBUS-DP-Slave-Bus-Parameter-Set (see IEC 62453-2 and also IEC 61158 series).

The master device uses this information to set up communication with the slave device.

9.1.2 Process Channel

In case of CPF 3 protocols, an FDT-Channel is a representative for a single date or a process value that can be accessed from a Frame Application via the master device. The information available at services for IO related information describes how to access a channel via a PROFIBUS DPV1 command or how to address a channel within a PROFIBUS DP frame for cyclic I/O. Besides all mandatory elements (which include `id` and `dpAddress`) it is highly recommended that the DPV1 address information is provided. This information (DPV1 Slot) is used by some frames to manage the PROFIBUS device module information.

In a DPV0 environment, depending on the situation, the underlying master device may have either Master Class 1 functionality or Master Class 2 functionality. A Class 1 master can write output data to a device and control data exchange, where a Class 2 master can only read the output data. Generally, it is assumed that parameterization as described here is performed as a master Class 2 station.

9.1.3 Parameterization

There are two options to write parameters set from the DTM's GUI to the CPF 3 slave device in the field:

- **User Parameters**

User Parameters are part of the PROFIBUS-DP-Slave-Bus-Parameter-Set. They contain manufacturer-specific data to characterize the DP-Slave. The DTM writes the User Parameters to the `busMasterConfigurationPart`. The User Parameters are stored with the master device during PROFIBUS master configuration and are automatically sent to the slave during set up of bus communication. (This is PROFIBUS-specific; for details, see IEC 61158 series.) When changing User Parameters on runtime, the DTM shall use a DP V0 connection and the appropriate DP V0 commands for parameter exchange as described in the data types.

- **Writing Parameters with DP-V1 services (MSAC2 services)**

The DTM may use DP-V1 transport services to send its parameters to the slave device. For that, it has to use a DP-V1 connection and the corresponding communication commands. During setup of communication, DP-V1 services are not sent automatically. The Frame Application or a DTM shall invoke a download of parameters via DP-V1.

For details on the different behavior of slaves depending on the kind of parameterization, refer to the IEC 61158 series.

DP-V1 connections and communication commands can also be used to execute commands at the slave. For details on the use of DP-V1, see also the IEC 61158 series.

9.2 Master-bus parameter set

The following parameter set represents the contents of the data type busMasterConfigurationPart within the device instance data for Profibus master device (Table 10). This attribute has to be set for each Profibus master device according to the IEC 61158 series. For further details, please refer to the sequence chart “Configuration of a Fieldbus Master” and the IEC 61158 series.

Table 10 – Bus parameter set for master device

Name	Type	Comment
Bus_Para_Len	UINT	Length of the bus parameter set inclusive, the length parameter (range 34 to 216-1)
FDL_Add	USINT	Mandatory part of the Bus Parameter Set according to PROFIBUS-DP Specification; not used in this context as station address is transferred in a separate variable; FDL-Add may differ from real station address used by EE and DTMs
Baud_rate	USINT	Code number for the baud rate
TSL	UINT	Slot time
min TSDR	UINT	Minimum station delay response
max TSDR	UINT	Maximum station delay response
TQUI	USINT	Quit-time
TSET	USINT	Setup-time
TTR	UDINT	Target rotation time
G	USINT	GAP update factor
HSA	USINT	Highest station address
max_retry_limit	USINT	Maximum retry limit
Bp_Flag	USINT	Flags for the user interface, e.g. error action flag
Min_Slave_Interval	UINT	Smallest allowed time period between two slave poll cycles
Poll_Timeout	UINT	Master-master timeout
Data_Control_Time	UINT	Guaranteed time period between two Data_transfer_list updates
Octet 1 (reserved)	ARRAY[6] of USINT	
...		
Octet 6 (reserved)		
Master_User_Data_Len	UINT	Length of Master_User_Data inclusive length parameter
Master_Class2_Name	ARRAY (32) of CHAR	Name of DP class 2 master the parameter set was created with
Master_User_Data	ARRAY of USINT	Manufacturer specific

9.3 Slave bus parameter set

The following parameter set represents the contents of the data type busMasterConfigurationPart within the instance data set for Profibus slave devices (Table 11). This attribute has to be set for each Profibus slave device according to the IEC 61158 series. For further details, please refer to the sequence chart “Configuration of a Fieldbus Master” and the IEC 61158 series.

Table 11 – Bus parameter set for slave device

Name	Type	Comment
Slave_Para_Len	UINT	Length of the slave parameter set inclusive the length parameter
Sl_Flag	USINT	Slave specific flags like New_Prm, Active, Fail_Safe,
Slave_Type	USINT	Manufacturer specific slave type denotation (0 by default for DP-Slaves)
Octet 1 (reserved)	ARRAY[12] of USINT	
...		
Octet 12 (reserved)		
Prm_Data_Len	UINT	Length of Prm_Data inclusive the length parameter (range 9 to 246)
Prm_Data	ARRAY of USINT	
Cfg_Data_Len	UINT	Length of Cfg_Data inclusive the length parameter (range 3 to 246)
Cfg_Data	ARRAY of USINT	
Add_Tab_Len	UINT	Length of Add_Tab inclusive the length parameter (range 2 to 216-31)
Add_Tab	ARRAY of USINT	Address assignment table
Slave_User_Data_Len	UINT	Length of Slave_User_Data inclusive the length parameter (range 2 to 216-31)
Slave_User_Data	ARRAY of USINT	Manufacturer specific data to characterize the DP-Slave for the master

9.4 Module and channel data

A slave's current module configuration including the belonging channels has to be available by FdtChannel objects. That is required as within the environment process variables have to be assigned to single channels and the slave configuration has to be displayed in the environment's system overview without using the DTM's user interface.

The addressing of channels within the PROFIBUS data frame is bit oriented and data type independent. A bit position and a bit length determine each channel. The data type of a parameter determines how to convert the bit field.

- Channels within PROFIBUS data frames (Figure 3):

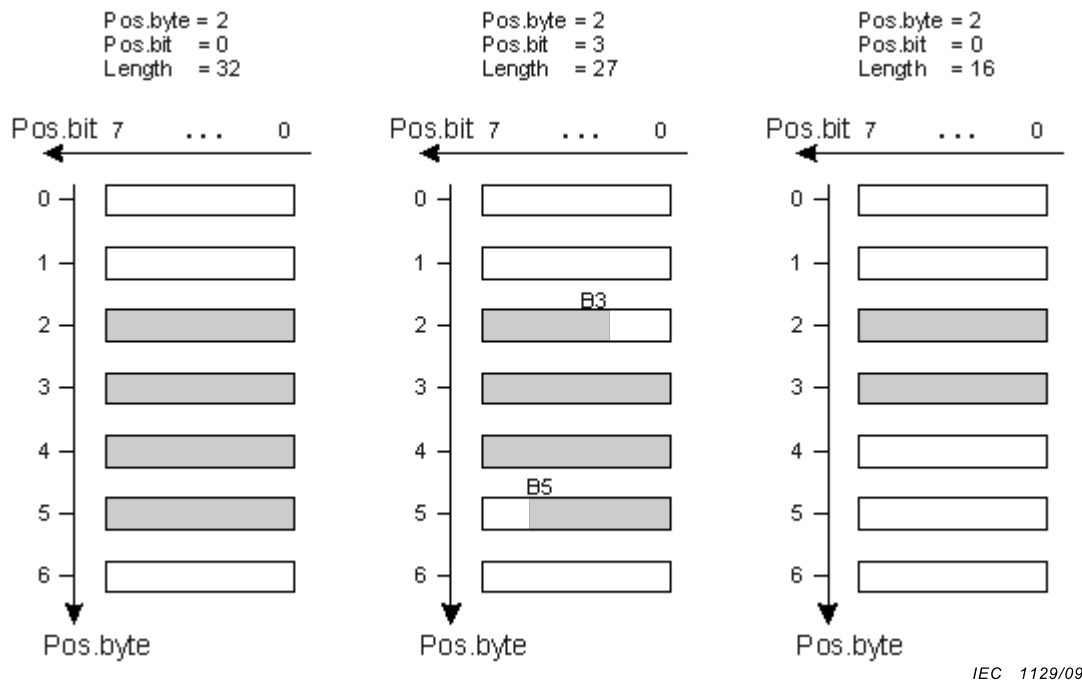


Figure 3 – Example for IO data within datagrams

- Representation in FDT Notation:

bitPosition="16"

bitPosition="19"

bitPosition="16"

bitLength="32"

bitLength="27"

bitLength="16"

The following example shows the expected structure description of a modular slave in a DTM parameter set using FDT-Parameters with the object types defined in the FDT-Specification.

- Representation of the data frame using GSD information:

BeginSlave;;

RIOLB8101.GSD;6;

BeginModules;;

1X03 FrequencyCount;0x51;

Empty ;0x00;

Empty ;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

2XXX ValveBlock.;0x30;

EndModules;;

EndSlave;;

Description of the position of signal channels within the data frame ():

Table 12 – Signal channels within the data frame

# Frame description Remote I/O											
Vendor:	"RIO manufacturer"										
Devicetype :	"LB 8101"		Device ident:			0x8101	# hex				
Frameident:	"B1t1"										
Input data											
Channel	Channel type	Data	Data	Data	Invalid Bit	Invalid Bit	Subst. Value Bit	Subst. Value Bit	Sim. Bit	Sim. Bit	State-Channel
		Pos. byte	Pos. bit	Length	Pos. byte	Pos. bit	Pos. byte	Pos. bit	Pos. byte	Pos. bit	
Module:	"EP01"	"Digital Input (Counter) LB/FB 1X03 (4 Byte)"									
"Count_1_0"	READ INT	0	0	32	-/-	-/-	-/-	-/-	-/-	-/-	0
Module:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input and Output)"									
"DI_8_LFOUT0"	READ BOOL	4	1	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF1"	READ BOOL	4	3	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF2"	READ BOOL	4	5	1	-/-	-/-	-/-	-/-	-/-	-/-	1
Output data											
Channel	Channel type	Data	Data	Data	Invalid Bit	Invalid Bit	Subst. Value Bit	Subst. Value Bit	Sim. Bit	Sim. Bit	State-Channel
		Pos. byte	Pos. bit	Length	Pos. byte	Pos. bit	Pos. byte	Pos. bit			
Modul:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input u. Output)"									
"DO_8_0"	WRITE BOOL	0	0	1	4	1	-/-	-/-	-/-	-/-	0

9.5 GSD information

9.5.1 General

The GSD information is not stored with single slave instances or in a global accessible file. It is provided by the DTM via the service `GetTypeInformation`. A DTM of a PROFIBUS device provides the GSD information within the `fdt:DtmDeviceType` argument.

The existing DCS uses the GSD information to obtain information about the possible configuration and parameters of a DP Slave.

Besides the information about modules and its parameters, a GSD file contains additional information about the slave, such as the supported baud rates.

This information is useful for a DCS system to configure an entire network according to the capabilities of different slaves.

9.5.2 GSD for gateway devices

9.5.2.1 Types of Profibus gateway devices

There are two types of gateway devices.

The visible gateway devices work as a PROFIBUS slave (with a PROFIBUS address) and to the underlying network they act as a master.

The invisible gateway devices just transform the PROFIBUS network to the underlying network.

For both types of devices there is a need for special GSDs.

9.5.2.2 Visible gateway devices

These visible gateway devices are shipped without a GSD file. Instead they have a proprietary software suite that configures and parameterizes it or they are shipped with a tool that creates a GSD for parameterization software. The GSD tool creates a GSD for the slave depending on the underlying network configuration or bus settings (for example baud rates).

In these cases the DTM should provide the functionality of the GSD tool. If the GSD is configuration-dependent, it could call service `NetworkManagementInfoRead` for each of its children, extract the GSD information and create configuration-dependent GSD information in the same way the tool does. After initialization of the DTM, it should deliver a BMCP according to the linking device itself. Every time a child is added or removed, this leads to a change in its own parameter document.

If the GSD depends on bus settings, a DTM's configuration or parameterization dialog could be used to change bus settings. Based on these settings, updated GSD information can be inserted in the information document. Here too the DTM has to request a save and call service `OnInstanceDataChanged`.

Note that the internal structure with its modules and channels has to be updated as well.

9.5.2.3 Transparent gateway devices

There are transparent linking devices on the market (PROFIBUS FMS/DP and PROFIBUS PA) performing a baudrate transformation. This requires a special handling of the slave specific GSD files. There are tools available which are able to adapt existing GSD files according to the higher baudrate (so called 'GSD Converter').

The GSD information must be delivered by the DTM for the device. In order to support this kind of linking devices, a slave DTM must expose the GSD file on hard drive using the attribute 'deviceTypeInformationPath' (refer to DtmDeviceType defined in IEC 62453-2).

A DTM developer should be aware, that the referenced GSD file might be converted in order to support specific system architecture.

It is expected that a Profibus DTM in the attribute 'deviceTypeInformation' is exposing exactly the GSD file which is referenced by the attribute 'deviceTypeInformationPath'.

10 Communication data types

10.1 General

The data types defined here are used at:

- Connect service;
- Disconnect service;
- Transaction service.

10.2 Error information provided by Communication Channel

In every Response data type of FDT Profibus specification an element 'errorCode' is provided. According to PROFIBUS, the error code is standardized to consist of 3 bytes, where each byte carries a meaning.

Since the error code is exchanged between different DTMs (e.g. Communication-DTM and Device-DTM) and since the receiver of the error code will try to understand the error code, the provider shall use the standard format:

- standard length 3 bytes;
- if the device provides error codes, these error codes are provided (and not local error codes from the Master).

If no error occurred, the element 'errorCode' shall be filled with zeros ("000000").

10.3 DPV0 communication

The supported services depend on the type of PROFIBUS master functionality that is provided by the communication infrastructure. Master Class 1 devices typically control the slaves and provide cyclic communication, Master Class 2 devices typically are used to configure the slaves and provide acyclic communication.

Not all defined services are supported if the Master is not in cyclic data exchange with the slaves. In such cases, the following behavior is expected:

If a Communication Channel receives a request that can not be supported, the Transaction service responds with a result="false".

The data types described in this clause are defined for the following namespace:

Namespace: dpv0

Table 13 – Simple DPV0 communication data types

Data type	Definition	Description
busAddress	USINT	Address information according to the IEC 61158 series (see also IEC 62453-2, attribute busAddress)
communicationReference	UUID	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during creation of the connection. The address information has to be used for all following communication calls
connectStatus	enumeration (masterConnectedOnly deviceAtLifeList deviceInDataExchange)	Describes the connection status established by the communication component. The status “masterConnectedOnly” means that the communication component has established a connection to the Profibus master device and will accept an online access to the user parameters, independent of whether the device is available or not. The status “deviceAtLifeList” means that the communication component has established a connection to the Profibus master device and has checked that the device is in the life list of the master stack. In this state, the master will accept an online access to the user parameters and will send the user parameter to the device, independent of whether the device is in data-exchange or not. The status “deviceInDataExchange” means that the communication component has established a connection to the Profibus master device and has checked that the device is in data-exchange. In this state, the master will accept an online access to the user parameters and will send the user parameter to the device, so that the new data will directly influence the process
delayTime	UDINT	Delay time in [ms] between two communication calls
errorCode	ARRAY OF USINT	Status information according to the IEC 61158 series. For description of error code see: DIN 19245 Part 3, PROFIBUS (P. 40ff., 83ff., 39)
schemaVersion	INT	Defines the version of the schema
sequenceTime	UDINT	Period of time in [ms] for the whole sequence

Table 14 – Structured DPV0 Communication data types

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
Abort	STRUCT			Describes the abort	-
	communicationReference	O	[0..1]		
ConnectRequest	STRUCT			Describes the communication request to establish a connection to a Profibus master device. Depending on the used network infrastructure it is possible that this service is not mapped to a field bus request, but is used to manage the software. It is also possible that this service is used as a trigger to set the state of a device	-
	busAddress	M	[1..1]		
	fdt:systemTag	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ConnectResponse	STRUCT			Describes the communication response to the connect request and provides the information how the following WriteUserParameter commands will be sent to the device (see connectStatus). The resulting connection depends on the used communication device and on the configuration of the bus master. The DeviceDTM shall expect that the resulting connection cannot be used to access all services. ConnectResponse may return true even if there is no communication with the device possible. For example if ConnectRequest returns true with connectionStatus = "masterConnectedOnly" this means that there is a connection to the master device, but the device is not at the life list and can therefore not be accessed. Depending on the connectStatus some services may not be available (e.g. in status "masterConnectedOnly" only services provided by the master are accessible (ReadUserParameter, WriteUserParameter, ...) whereas services provided by the device are not accessible. NOTE In this example the DTM is in state "online" but it cannot access the device. The provided services also depend on the type of master device. For an overview of availability of services depending on the type of connection, see the following tables.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
	connectStatus	M	[1..1]		
DisconnectRequest	STRUCT			Describes the communication request to release a connection to a Profibus master device	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
DisconnectResponse	STRUCT			Describes the communication response	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ReadDiagnosisDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification For Master Cl.1 refer to IEC 61158-5, 8.2.3.3.2 "Get slave diag" For Master Cl.2 refer to IEC 61158-5, 8.2.3.3.3 "Read slave diag"	1, 2
	communicationReference	M	[1..1]		
ReadDiagnosisDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadInputDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. Depending on whether the underlying FDTChannel is a Master Cl. 1 or a Master Cl. 2 this service will be local or result in a read access to the slave device. It is possible to specify which data is read by providing an fdt:ChannelReference. For Master Cl.1 refer to IEC 61158-5, 8.2.2.3.3 "Get Input" For Master Cl.2 refer to IEC 61158-5, 8.2.2.3.2 "Read Input"	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadInputDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ReadOutputDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. Depending on whether the underlying FDTChannel is a Master Cl. 1 or a Master Cl. 2 this service will be local or result in a read access to the slave device. It is possible to specify which data will be read by providing an fdt:ChannelReference. For Master Cl. 1 this is a proprietary service (or unavailable). For Master Cl. 2 refer to [1], section 8.2.2.3.6. "Read Output" (an optional service)	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadOutputDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadUserParameterRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The request retrieves only the information that is available from the master device (local service). It may differ from the actual data of the device. The implementation on how to provide the data is proprietary. This is based on a local service or not available	1
	communicationReference	M	[1..1]		
ReadUserParameterResponse	STRUCT			Describes the communication response, read from the Profibus master device, according to the Profibus DPV0 specification. The returned data may reflect only the information that is available from the master device. It may differ from the actual data of the device	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
SequenceBegin	STRUCT			Describes the sequence begin	-
	sequenceTime	O	[0..1]		
	delayTime	O	[0..1]		
	communicationReference	M	[1..1]		
SequenceEnd	STRUCT			Describes the sequence end	-
	communicationReference	M	[1..1]		
SequenceStart	STRUCT			Describes the sequence start	-
	communicationReference	M	[1..1]		
WriteOutputDataRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The output data will be sent according to the established connection. It is necessary to specify which data will be written by providing an fdt:ChannelReference. If the underlying FDTChannel is provided as Master CI. 2, writing Output Data will be not possible. Refer to IEC 61158-5, 8.2.2.3.5 "Set Output". This functionality depends on the used type of master. In terms of PLC this functionality is named "forcing" of values	1
	communicationReference	M	[1..1]		
	fdt:ChannelReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteOutputDataResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
WriteUserParameterRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The user parameter will be sent according to the established connection (see connectStatus). For Master Cl. 1 this is based on the service "DDL_M_Set_Prm". For Master Cl. 2 this is based on the service "DDL_M_Set_Prm" (an optional service)	1, 2
	communicationReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteUserParameterResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Depending on the bus master type and on the returned connectStatus following services are available.

Table 15 – Availability of services for Master Class1 (C1)

Slave DTM Service Request	connectStatus		
	masterConnectedOnly	DeviceAtLifeList	DeviceInDataExchange
Connect	✓	✓	✓
ReadUserParameter	O	O	O
WriteUserParameter	✓	✓	✓
ReadOutputData			✓
WriteOutputData			O
ReadInputData			✓
ReadDiagnosisData		✓	✓
NOTE ✓ : the service is available, O : the service is optional and may be available, depending on the capabilities of the underlying master device.			

For Master Class2 (C2), not all connect states are available:

Table 16 – Availability of services for Master Class2 (C2)

Slave DTM Action	connectStatus	
	DeviceAtLifeList (no DPV1 connection to device)	DeviceInDataExchange (DPV1 connection to device)
Connect	✓	✓
ReadUserParameter		
WriteUserParameter	O	O
ReadOutputData		O

Slave DTM Action	connectStatus	
	DeviceAtLifeList (no DPV1 connection to device)	DeviceInDataExchange (DPV1 connection to device)
WriteOutputData		
ReadInputData		✓
ReadDiagnosisData	✓	✓
NOTE ✓ : the service is available, O : the service is optional and may be available, depending on the capabilities of the underlying master device.		

If the Master Class 2 communication component supports DPV1 and DPV0 and has established a DPV1 connection to the device, a call to service Connect for DPV0 shall return the status “DeviceInDataExchange” even if the device is not in status DataExchange.

If no DPV1 connection is established, the Master Class 2 communication component shall verify the availability of the device (at least by service LifeList).

10.4 DPV1 communication

The data types are defined to provide the address information and the communication data required for DPV1 communication (see Table 17 and Table 18).

The data types described in this subclause are defined for the following namespace:

Namespace: dpv1

Table 17 – Simple DPV1 communication data types

Data type	Definition	Description
api	USINT	Address information according to the IEC 61158 series. If the device needs special values for the DPV1-Initiate, the DeviceDTM is responsible for providing the values in the ConnectRequest. If the values are not provided, the communication component will use default values (0)
busAddress	USINT	Address information according to the IEC 61158 series (see also DTMPParameterSchema, attribute busAddress)
errorCode	UUID	Status information according to the IEC 61158 series. For description of error code see: PROFIBUS Nutzerorganisation e.V., PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998. (P.. 120-121); .For description of abort information see: DIN 19245 Part 3, PROFIBUS, (P..54ff., 11)
index	UDINT	Address information according to the IEC 61158 series
communicationReference	ARRAY OF USINT	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during the connect. The address information has to be used for all following communication calls
delayTime	USINT	Minimum delay time in [ms] between two communication calls

Data type	Definition	Description
maxLenDataUnit	USINT	Optional attribute, to describe the amount of data, which can be transferred via the established connection. If this attribute is not available, no special length restriction is announced. Each communication component within the chain of interfaces concerning nested communication could introduce this attribute. Each communication component should change the contents of the attribute based on the following rule: The new value is the minimum of the current value and the restriction of its own implementation. If a communication component has no restriction, it should hand over the given value. If a communication component is able to reuse an established connection concerning a new connect request, it should take into account the data length determined for the existing connection. If the data length is not applicable, the DTM should send an error message via service ErrorMessage
networkMACAddress	ARRAY OF USINT	Network or MAC address as described in the IEC 61158 series. [5] If there is a networkMACAddress, (i.e. networkMACAddress <> ""), the Profibus-DPV1-Master-DTM shall set the corresponding "address type" in the Initiate request telegram "1", else "0". Example Device with address 20 (14hex) addressed via a linking device: networkMACAddress="00000000000014"
scl	INT	Access level as described in the IEC 61158 series. [5] If the device needs special values for the DPV1-Initiate, the DeviceDTM is responsible for providing the values in the ConnectRequest. If the values are not provided, the communication component will use default values (0)
sequenceTime	USINT	Period of time in [ms] for the whole sequence
slot	UDINT	Address information according to the IEC 61158 series

Table 18 – Structured DPV1 communication data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
Abort	STRUCT			Describes the abort
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Describes the communication request
	api	M	[1..1]	
	busAddress	M	[1..1]	
	fdt:systemTag	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
ConnectResponse	STRUCT			Describes the communication response
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	maxLenDataUnit	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
DestNetworkAddress	STRUCT			Describes the extended address of the destination [5]
	NetworkAddress	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
NetworkAddress	STRUCT			Describes the extended address format
	api	M	[1..1]	
	scl	M	[1..1]	
	networkMACAddress	M	[1..1]	
ReadRequest	STRUCT			Describes the communication request
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
ReadResponse	STRUCT			Describes the communication response
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
RedundantAddresses	STRUCT			Describes the redundant addresses
	fdtparam:SlaveAddress	M	[1..*]	
SequenceBegin	STRUCT			Describes the sequence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
SequenceEnd	STRUCT			Describes the sequence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Describes the sequence start
	communicationReference	M	[1..1]	
SrcNetworkAddress	STRUCT			Describes the extended address of the source
	NetworkAddress	M	[1..1]	
WriteRequest	STRUCT			Describes the communication request
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
WriteResponse	STRUCT			Describes the communication response
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	

Within a connect request, a DTM of a Profibus redundant slave can provide additional redundant slave addresses, set within its parameter document. The busAddress attribute is to be used as the preferred address; the addresses in the redundantAddresses element should be used in this order if an alternative address is used to connect to the redundant slave.

11 Channel parameter data types

The following table (Table 19) describes the mapping of DPV1 Data types to FDT data types.

Table 19 – Mapping of DPV1 data types to FDT data types

PROFIBUS data types	FDT data type
Boolean	BOOL
Integer8	SINT
Integer16	INT
Integer32	DINT
Unsigned8	USINT
Unsigned16	UINT
Unsigned32	UDINT
Floating Point	FLOAT
Visible String	STRING
Octet String	ARRAY of USINT
Date	DATE
Time of Day	TIME
Time Difference	duration
DS-32	structured

PROFIBUS data types	FDT data type
DS-33	structured
DS-34	structured
DS-35	structured
DS-36	structured
...	structured
DS-52	structured
..	structured

The format of all data types is defined in the IEC 61158 series.

If only a part of the retrieved data is process data (e.g. for DS-33 the data consist of the float-value and a byte status), the access information (DPAAddress or DPV1Address) also contains information that is necessary to retrieve the information from the data structure.

Used at ReadChannelInformation service and WriteChannelInformation service.

The Channel parameter data types describe a how to access a channel via a Profibus DPV1 command or how to address a channel within a Profibus DP frame for cyclic I/O. The following tables (Table 20 and Table 21) provide a description of the data types.

The data types described in this clause are defined for the following namespace:

Namespace: profichannel

Table 20 – Simple Channel Parameter data types

Data type	Definition	Description
api	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
bitLength	UDINT	Additional data type information especially for fieldbus specific data types like 12 bit integer
bitPosition	UDINT	Address information according to the IEC 61158 series for channels accessible via Profibus DP
frameApplicationTag	STRING	Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces
gatewayBusCategory	UUID	Unique identifier for a supported bus type like Profibus or HART according to the FDT specific CATID
index	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
invalidBit	UDINT	Bit position of the invalid status channel accessible via Profibus DP
logic	UDINT	Additional data type information: positive 0=FALSE 1=TRUE
number	BOOL	Address information for diagnosis according to the IEC 61158 series for channels accessible via Profibus DP. See [5], 9.3.1 Read DP – Slave Diagnostic Information
protectedByChannelAssignment	INT	TRUE if the channel is set to read only by the Frame Application. Usually set to TRUE if a channel assignment exists
schemaVersion	UDINT	Defines the version of the schema
simulationBit	USINT	Bit position of the simulation status channel accessible via Profibus DP
slotNumber	BOOL	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
statusChannel	UDINT	TRUE if the channel is for status information only
substituteValueBit	STRING	Bit position of the substitute status channel accessible via Profibus DP

Table 21 – Structured Channel Parameter data types

Data type	Definition			Description
	Elementary data type	U s a g e	Multi- plicity	
DpAddress	STRUCT			Address information according to the IEC 61158 series for channels accessible via Profibus DP
	bitPosition	M	[1..1]	
	bitLength	M	[1..1]	
DpV1Address	STRUCT			Address information according to the IEC 61158 series for channels accessible via Profibus DPV1. It is highly recommended to provide this address information to enable assignment of logical addresses to a profibus module
	api	M	[1..1]	
	slotNumber	M	[1..1]	
	index	M	[1..1]	
	bitPosition	O	[0..1]	
	bitLength	O	[0..1]	
FDTChannel	STRUCT			Description of the channel
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannel-Assignment	M	[1..1]	
	number	M	[1..1]	
	fdt:dataType	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	applId:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	DpAddress	O	[0..1]	
	DpV1Address	O	[0..1]	
	StatusInformation	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:StructuredElements	O	[0..1]	

Data type	Definition			Description
	Elementary data type	U s a g e	Multi- plicity	
FDTChannelType	STRUCT			Description of the channel component in case of channels with gateway functionality
	fdt:VersionInformation	M	[1..1]	
	gatewayBusCategory	O	[0..1]	
	statusChannel	O	[0..1]	
StatusInformation	STRUCT			Description of additional status information for channels accessible via Profibus DP
	logic	M	[1..1]	
	invalidBit	O	[0..1]	
	simulationBit	O	[0..1]	

12 Device identification

12.1 General

There are different Profibus specific identification elements.

A Profibus Scan may detect different device types: I&M devices, Profile devices or pure DP devices.

Following rule shall be applied for Profibus Communication Channels:

- if I&M is available, create I&M identification,
- otherwise check Profile PA,
- otherwise create pure DP info = IDENT_NUMBER.

The data types described in this subclause are defined for the following namespace:

Namespace: profiident

12.2 Protocol specific handling of the data type STRING

Profibus char array rules:

- in all strings based on char ranges defined in the field bus protocol specification, the leading spaces are left trimmed. The char array is to be filled with 0x20h (blank);
- in VisibleStrings, invisible characters provided by a device have to be replaced by '?'.

12.3 Common device type identification data types

The data types described in this clause are reused as defined by 12.4 and 12.6 .

Table 22 lists the identification relevant data, source and format for Profibus DP.

Table 22 – Identification data types with Profibus DP specific mapping

Profibus - Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT data type (display format)	Specification reference
busProtocol	IdBusProtocol	For all DP devices : "protocol_DP"	Protocol	-	Enum	-
slaveAddress	IdAddress	Bus address is provided as part of live list by a PROFIBUS master. Service: FMA1/2_LIVE_LIST	Slave Address	Unsigned8	STRING (dec)	[5] Part 2, Section 4.2.3.6
identNumber	IdTypeID	IDENT_NUMBER can be requested by: DP Service DDLM_SLAVE_DIAG Allowed values are: Profile IDENT_NUMBER : 0x9700 (0x9700 to 0x9742) or Manufacturer specific IDENT_NUMBER	IDENT_NUMBER	Unsigned16	ARRAY of USINT (hex)	[5] Part 8, Section 9.3.1
manufacturerSpecificExtension		Can be used by DTM for a vendor specific device identification information, e.g. by combining a number of device parameter values into one string value. This can be used to identify a specific device variant			STRING	

Table 23 shows the mapping for PROFIBUS I&M data.

Table 23 – Identification data types with Profibus I&M specific mapping

Profibus-Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT format (display format)	Specification reference
identNumber, manufacturerSpecificExtension and slaveAddress as defined in DP table (Table 22).						
busProtocol	IdBusProtocol	For all I&M devices : "protocol_IM"	Protocol	-	Enum	-
manufacturerId	IdManufacturer	I&M 0 Element 1 Mapped to PB.DEVICE_MAN_ID	MANUFACTURER_ID	Unsigned16	UINT	[6] Section 3.2.2
orderId	-	I&M 0 Element 2 Mapped to PB.DEVICE_ID	ORDER_ID	20 Octets Visible String	STRING[20]	[6] Section 3.2.3
serialNumber	IdSerialNumber	I&M 0 Element 3 unique serial number Mapped to PB.DEVICE_SER_NUM	SERIAL_NUMBER	16 Octets VisibleString	STRING[16]	[6] Section 3.2.4
hardwareRevision	IdHardwareRevision	I&M 0 Element 4 HARDWARE_REVISION NOTE: If the device is a PA device,(PROFILE_ID is set to 0x9700), HARDWARE_REVISION shall be read from PA_I&M0 [6]	HARDWARE_REVISION	2 Octets - unsigned int 16 Converted to string to cover also: PA_I&M0 : VisibleString 16	STRING	[6] Section 3.2.5, Amendment I&M
softwareRevision	IdSoftwareRevision	I&M 0 Element 5 SOFTWARE_REVISION NOTE: If the device is a PA device (PROFILE_ID is set to 0x9700), SOFTWARE_REVISION shall be read from PA_I&M0 String format is used in FDT to harmonize version handling for all fieldbus protocols.	SOFTWARE_REVISION	4 Octets – 1 char + 3 unsigned 8 , e.g.: V1.2.3 PA_I&M0 : VisibleString 16	STRING	[6] Section 3.2.6, Amendment I&M
profileID	-	I&M 0 Element 7	PROFILE_ID	2 Octets – unsigned 16	UINT	[6] Section 3.2.8

Profibus-Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT format (display format)	Specification reference
profileSpecificType	-	I&M 0 Element 8 MSB = BLOCK_OBJECT, LSB = PARENT_CLASS	PROFILE_SPECIFIC_TYPE	2 Octets – unsigned 16	UINT	[6] Section 3.2.9
imVersion	-	I&M 0 Element 9	IM_VERSION	2 Octets – unsigned 16; MSB major version xxx, LSB minor version yyy	FLOAT xxx.yyy	[6] Section 3.2.10
imSupported	-	I&M 0 Element 10	IM_SUPPORTED	2 Octets -	UINT	[6] Section 3.2.11
tagFunction	IdTag	I&M 1 Element 1 TAG_FUNCTION	TAG_FUNCTION	32 Octets – visible string	STRING[32]	[6] Section 3.2.12
tagLocation	-	I&M 1 Element 2 TAG_LOCATION	TAG_LOCATION	22 Octets – visible string	STRING[22]	[6] Section 3.2.13
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM ProfileDTM, BlockSpecificProfileDTM	DTM Support Level	-	Enum	

Table 24 lists the identification relevant data, source and format for Profibus PA.

Table 24 – Identification data types with Profibus PA specific mapping

Profibus-Attribute name (semantic name)	Semantic element	Data request in physical device	Protocol specific name	Profibus data format	FDT format (display format)	Specification reference
identNumber, manufacturerSpecificExtension and slaveAddress as defined in DP table (Table 22).						
busProtocol	IdBusProtocol	For all PA devices : "protocol_PA"	Protocol	-	enum	-
manufacturerId	IdManufacturer	Physical Block - Index 10 (Unsigned16 – 2 decimal)	DEVICE_MAN_ID	Unsigned16	UINT	[2] Part 1, Section 3.11
device_id	-	Physical Block - Index 11 - VisibleString 16	DEVICE_ID	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
profile	-	Block structure of physical block – element 7 : OctetString - (Index 10 – size 2)	Profile	OctetString2	UINT	[2]0 Part 1, Section 3.7.2
profileRevision	IdBusProtocolVersion	Block structure of physical block – element 8 Unsigned 16 - (Index 6)	Profile Revision	Unsigned16	UINT	[2] Part 1, Section 3.7.2
softwareRevision	IdSoftwareRevision	Physical Block - Index 8 (VisibleString 16)	SOFTWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
hardwareRevision	IdHardwareRevision	Physical Block - Index 9 (VisibleString 16)	HARDWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
serialNumber	IdSerialNumber	Physical Block - Index 12 (VisibleString 16)	DEVICE_SER_NUM	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
deviceTAG	IdTag	BO of PB - OctetString 32	TAG_DESC	OctetString32	STRING[16]	
profileSpecificType	-	FDT attribute mapped to Block Object PARENT_CLASS sort of first TransducerBlock	BO PARENT_CLASS	Unsigned16	UINT	[2] Part 1, Section 3.7.2
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	DTM Support Level		enum	

Table 25 and Table 26 show identification data types with protocol independent semantics.

Table 25 – Simple identification data types with protocol independent semantics

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	enumeration - genericSupport - profileSupport - blockspecificProfileSupport - specificSupport - identSupport
match	STRING	Used by Device DTM to define a regular expression which shall match to scanned physical define identification information
nomatch	STRING	Used by Device DTM to define a regular expression which shall not match to scanned physical define identification information. Used by Device DTM to indicate if identification information may not match

Table 26 – Structured identification data types with protocol independent semantics

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
RegExpr	STRUCT			Includes regular expression string – either for match or for nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.4 Topology scan data types

These data types are used at Scan service response (Table 27 and Table 28).

Table 27 – Simple topology scan data types

Data type	Definition	Description
busAddress	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1

The data type ProfibusDevice describes one entry in the list of scanned PROFIBUS-Devices (see Table 28).

Table 28 – Structured topology scan data types

Data type	Definition			Description
	Elementary data type	Usage	Multiplicity	
ProfibusDevice	STRUCT			Specifies a Profibus device
	schemaVersion	O	[0..1]	
	busAddress	M	[1..1]	
	fdt:deviceTypeId	M	[1..1]	
	fdt:subDeviceType	O	[0..1]	

12.5 Scan identification data types

This subclause defines data types that are used to provide protocol specific scanning (see Table 29 and Table 30).

The data types described in this subclause are used in the following services:

- Scan service

Table 29 – Simple scan identification data types

Data type	Definition	Description
configuredState	enumeration (configuredAndPhysicallyAvailable configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	A communication master shall indicate in this attribute if the scan response is related to a detected physical device which is configured or unconfigured
resultState	enumeration (provisional final error)	Identifies if the result is one of the provisional results or the final result of the split scan results

Table 30 – Structured scan identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdAddress	STRUCT			All data types contain exactly one attribute each including the value of the scanned physical device. All data types with semantic meaning have a prefix “Id” for better identification
	profibusident:slaveAddress	M	[1..1]	
IdBusProtocol	STRUCT			
	profibusident:busProtocol	M	[1..1]	
IdBusProtocolVersion				
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	M	[1..1]	
IdManufacturer				
IdSerialNumber	STRUCT			These data types contain corresponding attributes defined in IEC 62453-2. They are transformed to name value pairs without semantic meaning for Frame Application
	profibusident:serialNumber	M	[1..1]	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	M	[1..1]	
IdTypeID	STRUCT			
	profibusident:identNumber	M	[1..1]	
Manufacturer_id	STRUCT			
	profibusident:manufacturer_id	M	[1..1]	
Order_id	STRUCT			
	profibusident:order_id	M	[1..1]	
ProfileID	STRUCT			
	profibusident:profileID	M	[1..1]	
ProfileSpecificType	STRUCT			
	profibusident:profileSpecificType	M	[1..1]	
ImVersion	STRUCT			
	profibusident:imVersion	M	[1..1]	
TagFunction	STRUCT			

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	profibusident:tagFunction	M	[1..1]	
TagLocation	STRUCT			
	profibusident:tagLocation	M	[1..1]	
ImSupported	STRUCT			
	profibusident:imSupported	M	[1..1]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:software↗RevisionP A	M	[1..1]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardware↗Revision PA	M	[1..1]	
ProfileRevisionPA	STRUCT			
	profibusident:profileRevisionPA	M	[1..1]	
Device_man_id	STRUCT			
	profibusident:device_man_id	M	[1..1]	
Device_id	STRUCT			
	profibusident:device_id	M	[1..1]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profileSpecific↗Type PA	M	[1..1]	
ManufacturerSpecific ↗Extension				These elements contain all elements for the appropriate protocol variant
	STRUCT			
	profibusident:manufacturer↗Speci ficExtension	M	[1..1]	
ScanIdentification_IM	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	ImVersion	M	[1..1]	
	TagFunction	M	[1..1]	
	TagLocation	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	ImSupported	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_PA	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	
	ProfileSpecificTypePA	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	IdTag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_DP	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentifications	STRUCT			Collection of ScanIdentification elements
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	choice of	M	[1..*]	
	ScanIdentification_DP	S	[0..*]	
	ScanIdentification_IM	S	[0..*]	
	ScanIdentification_PA	S	[0..*]	

12.6 Device type identification data types – provided by DTM

This subclause defines data types that are used to provide protocol specific information for device types (see Table 31).

The data types described in this clause are used in the following services:

- GetIdentificationInformation service

Table 31 – Structured device identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdBusProtocol	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix “Id” for better identification
	profibusident:busProtocol	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdBusProtocolVersion				
IdManufacturer				
IdTypeID	STRUCT			
	profibusident:identNumber	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Manufacturer_id	STRUCT			These elements contain corresponding attributes defined in IEC 62453-2. They are transformed to name value pairs without semantic meaning for Frame Application
	profibusident:manufacturer_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Order_id	STRUCT			
	profibusident:order_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileID	STRUCT			
	profibusident:profileID	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificType	STRUCT			
	profibusident:profileSpecificType	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImVersion	STRUCT			
	profibusident:imVersion	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImSupported	STRUCT			
	profibusident:imSupported	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:softwareRevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardwareRevisionPA	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	profibusident:RegExpr	O	[0..*]	
ProfileRevisionPA	STRUCT			
	profibusident:profile↗RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_man_id	STRUCT			
	profibusident:device_man_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_id	STRUCT			
	profibusident:device_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profile↗SpecificTypePA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ManufacturerSpecific↗Extension	STRUCT			
	profibusident:manufacturer↗SpecificExtension	M	[1..1]	
DeviceIdentification_IM	STRUCT			These elements contain all elements for the appropriate protocol variant
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	ImVersion	M	[1..1]	
	ImSupported	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification_PA	STRUCT			
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	ProfileSpecificTypePA	M	[1..1]	
	ManufacturerSpecific¬Extension	O	[0..1]	
DeviceIdentification_DP	STRUCT			
	profibusident:idDTM¬SupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdTypeID	M	[1..1]	
	ManufacturerSpecific¬Extension	O	[0..1]	
DeviceIdentification s	STRUCT			Collection of DeviceIdentification elements
	fdt:protocolId	M	[1..1]	
	choice of	M	[1..*]	
	DeviceIdentification_DP	S	[0..*]	
	DeviceIdentification_IM	S	[0..*]	
	DeviceIdentification_PA	S	[0..*]	

12.7 Identification information in GUI

It is mandatory to display I&M parameter labels in DTMs Presentation Object or Frame Application user interface from IM-Runtime.xml, in order to achieve consistency and acceptance for the user.

13 ProfiSafe

13.1 Motivation

The purpose of the ProfiSafe-UseCase paper is to compile a collection of typical existing failsafe automation examples from the discrete and continuous manufacturing areas and to derive a common systematic approach. This proposed solution ought to be based on

- Profibus DPV1 and ProfiSafe communication principles;
- new possibilities of combined failsafe controllers (standard and failsafe programs within one PLC);
- and current activities of other Profibus working groups like FDT/DTM and Proxy FB.

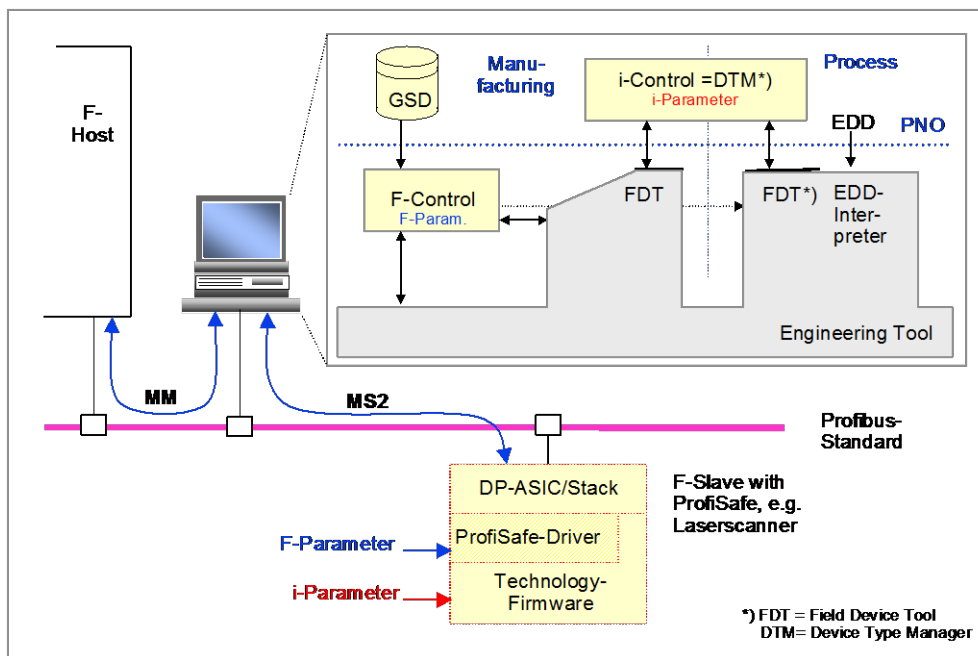
Within the context of the ProfiSafe-UseCase paper this PAS covers the partial aspect "FDT programmers interface for individual device parameter verification".

13.2 General parameter handling

In order to communicate in a safe way, each ProfiSafe device (F-Slave) requires so-called F-Parameters for the adjustment of the operational mode of its ProfiSafe driver. Those parameters comprise watch dog time, safety integrity level, container size, etc. In case of simple F-slaves no further parameters are required. F-Parameters will therefore be defined within GSD Revision 4.0.

In contrast, a complex F-slave requires additional individual and safety relevant parameters (individual device parameter) that, due to its size (>240 Bytes), often cannot be transported by the initial parameterization telegram (Prm-Telegram). As an amendment, ProfiSafe

guidelines are suggesting an additional method via proxy function blocks in a safety PLC, thus providing additional functionality like program-controlled reparameterization, etc.



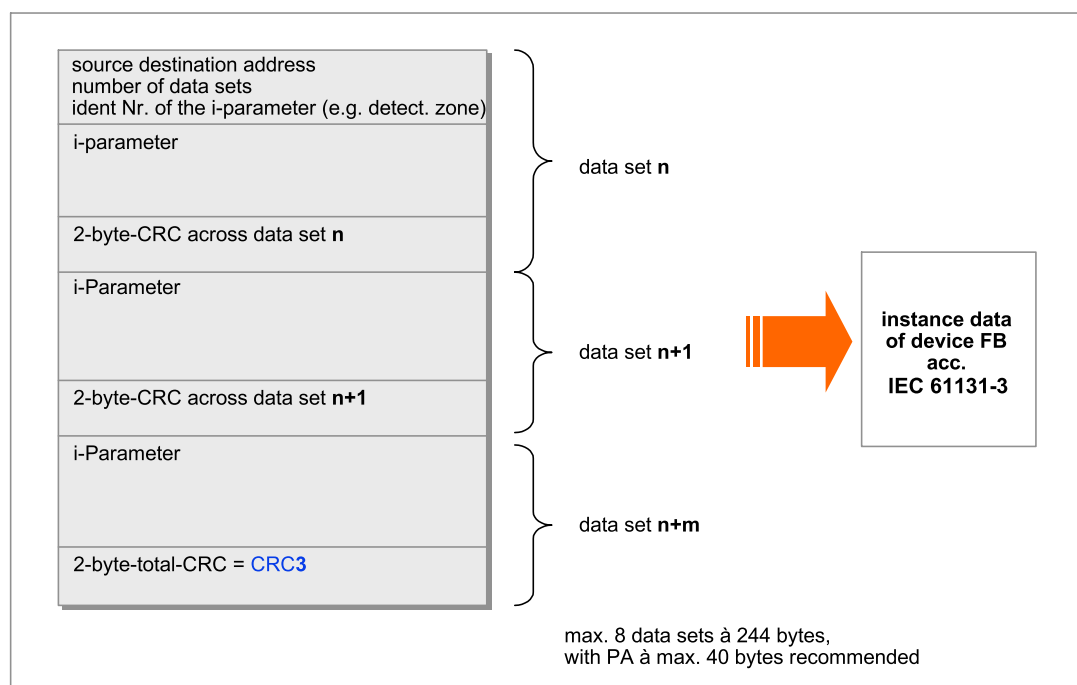
IEC 1130/09

Figure 4 – F-Parameter and individual device parameter

Individual device parameters, being F-device specific by nature, ought to be handled by Device-Type-Manager programs (DTM) coming from the F-device manufacturers. The engineering tool operates as a Frame Application for such a DTM of a F-slave. It is routing the communication requests of a DTM to its device (MS2) and provides data persistence. Otherwise, it is covering all traditional tasks like network configuration, parameterization, commissioning and diagnosis. It is communicating with the PLC via the Master-Master-Protocol (MM). In this respect, a PLC on the Profibus is a device of its own class (Master Class 1). The engineering tool itself is defined to be "a Master Class 2" device.

13.3 ProfiSafe individual device parameter

The structure of the ProfiSafe individual device parameter set is defined within the ProfiSafe guidelines. This set is wrapped up in XML-format for transportation across the FDT interface.



IEC 1131/09

Figure 5 – Data structure of ProfiSafe individual device parameters

The data structures and the data formats of the inner individual device parameters are proprietary. The whole set is divided into parts that fit into a Profibus data set ready for transportation. Each data set comprises a 2-bytes CRC for data consistency check. The polynomial is fixed and defined within the ProfiSafe guidelines. The CRC of the last set checks the total block.

Bibliography

- [1] PROFIBUS-PA Profile for Process Control Devices Version 3.0
 - [2] PROFIBUS Profile Guidelines, Part 1, Identification & Maintenance Functions, Version 1.1, May 2003
 - [3] PROFIBUS Guideline (Order No. 2.082): Chapter “Services on communication relationships MSAC_C2”: MSAC2_Initiate
 - [4] PROFIBUS Profile Amendment 3 to the Profibus Profile for Process Control Devices V3.0
 - [5] IEC 61158-6:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 6: Application layer protocol specification*
 - [6] PROFIBUS Specification Specification for PROFIBUS Device Description and Device Integration, Volume 1: GSD Version 5.04, July 2005
 - [7] IEC 61158-5:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 5: Application layer service specification*
 - [8] PROFIBUS Nutzerorganisation e.V., *PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998*
-

SOMMAIRE

AVANT-PROPOS.....	65
INTRODUCTION.....	67
1 Domaine d'application.....	69
2 Références normatives	69
3 Termes, définitions, symboles, abréviations et conventions	70
3.1 Termes et définitions	70
3.2 Symboles et abréviations	70
3.3 Conventions	71
3.3.1 Noms de type de données et références aux types de données.....	71
3.3.2 Vocabulaire pour les exigences	71
3.3.3 Utilisation de l'UML	71
4 Catégorie de bus	71
5 Accès aux données d'instances et de dispositifs.....	72
5.1 Objets "Process Channel" fournis par le DTM.....	72
5.2 Services DTM pour accéder aux données d'instances et de dispositifs.....	72
6 Comportement spécifique à un protocole.....	72
6.1 Modèle de dispositif PROFIBUS.....	72
6.2 Configuration et paramétrisation des dispositifs PROFIBUS.....	74
6.2.1 Généralités	74
6.2.2 DTM monolithique pour un dispositif PROFIBUS modulaire	74
6.2.3 DTM modulaire pour un dispositif PROFIBUS modulaire	75
6.3 Support pour la configuration DPV0	75
6.4 Esclaves PROFIBUS fonctionnant sans un maître PROFIBUS cyclique.....	75
6.5 Informations relatives à PROFIBUS d'un DTM esclave	76
6.5.1 Généralités	76
6.5.2 Bus Master Configuration Part (partie configuration principale de bus (BMCP))	76
7 Usage spécifique à un protocole des types de données généraux.....	88
8 Types de données communs spécifiques à un protocole	90
9 Types de données de gestion de réseau	90
9.1 Généralités.....	90
9.1.1 Configuration	91
9.1.2 Process Channel (voie de processus).....	91
9.1.3 Paramétrisation.....	92
9.2 Jeu de paramètres du bus maître	92
9.3 Jeu de paramètres du bus esclave	93
9.4 Données de module et de voie	94
9.5 Informations de GSD	97
9.5.1 Généralités	97
9.5.2 GSD pour dispositifs passerelles	97
10 Types de données de communication.....	98
10.1 Généralités.....	98
10.2 Informations d'erreur fournies par la voie de communication Communication Channel	98
10.3 Communication DPV0	98

10.4 Communication DPV1	105
11 Types de données paramètres de voie (Channel)	108
12 Identification de dispositif	112
12.1 Généralités.....	112
12.2 Traitement, spécifique à un protocole, du type de données STRING	112
12.3 Types de données d'identification de types de dispositif communs.....	112
12.4 Types de données de balayage de topologie	117
12.5 Types de données d'identification de balayage (Scan).....	118
12.6 Types de données d'identification de type de dispositif – fournis par DTM.....	121
12.7 Informations d'identification dans l'interface GUI	123
13 ProfiSafe	124
13.1 Motivation	124
13.2 Traitement général des paramètres	124
13.3 Paramètre de dispositif individuel ProfiSafe.....	126
Bibliographie	127
 Figure 1 – Partie 303-1 de la série IEC 62453	68
Figure 2 – Modèle de dispositif PROFIBUS FDT	74
Figure 3 – Exemple pour données E/S dans des datagrammes	94
Figure 4 – F-Parameter (paramètre F) et paramètre de dispositif individuel	125
Figure 5 – Structure de données des paramètres de dispositifs individuels ProfiSafe.....	126
 Tableau 1 – Identificateurs de protocole	71
Tableau 2 – Identificateurs de couche physique.....	71
Tableau 3 – BMCP Partie 1 – Configuration générale	78
Tableau 4 – BMCP Partie 2 – Données de paramètres.....	78
Tableau 5 – BMCP Partie 3 – Données de configuration	79
Tableau 6 – Partie 4: Table d'adresses et paramètres utilisateur d'esclave.....	80
Tableau 7 – Partie 4: Extended Prm data (Données de paramètres étendues).....	80
Tableau 8 – BMCP complète	81
Tableau 9 – Usage spécifique à un protocole des types de données généraux	88
Tableau 10 – Jeu de paramètres de bus pour dispositif maître	92
Tableau 11 – Jeu de paramètres de bus pour dispositif esclave	93
Tableau 12 – Voies de signal au sein de la trame de données	96
Tableau 13 – Types de données simples de communication DPV0.....	99
Tableau 14 – Types structurés de données de communication DPV0	100
Tableau 15 – Disponibilité des services pour la classe 1 de maître (C1).....	105
Tableau 16 – Disponibilité des services pour la classe 2 de maître (C2).....	105
Tableau 17 – Types de données simples de communication DPV1.....	106
Tableau 18 – Types de données structurés de communication DPV1	107
Tableau 19 – Mise en correspondance des types de données DPV1 aux types de données FDT	109
Tableau 20 – Types simples de données pour les paramètres de voies	110
Tableau 21 – Types structurés de données pour les paramètres de voies	111

Tableau 22 – Types de données d'identification avec mise en correspondance spécifique à Profibus DP	113
Tableau 23 – Types de données d'identification avec mise en correspondance spécifique à Profibus I&M	114
Tableau 24 – Types de données d'identification avec mise en correspondance spécifique à Profibus PA	116
Tableau 25 – Types de données d'identification simples avec sémantique indépendante vis-à-vis du protocole	117
Tableau 26 – Types de données d'identification structurés avec sémantique indépendante vis-à-vis du protocole	117
Tableau 27 – Types de données simples de balayage de topologie	117
Tableau 28 – Types de données structurés de balayage de topologie	118
Tableau 29 – Types de données simples d'identification de balayage	118
Tableau 30 – Types de données structurés d'identification de balayage	118
Tableau 31 – Types de données structurés d'identification de dispositif	121
Tableau 32 – Identificateurs de couche DataLink	72

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

SPÉCIFICATION DES INTERFACES DES OUTILS DES DISPOSITIFS DE TERRAIN (FDT) –

Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1 et CP 3/2

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

DÉGAGEMENT DE RESPONSABILITÉ

Cette version consolidée n'est pas une Norme IEC officielle, elle a été préparée par commodité pour l'utilisateur. Seules les versions courantes de cette norme et de son(ses) amendement(s) doivent être considérées comme les documents officiels.

Cette version consolidée de l'IEC 62453-303-1 porte le numéro d'édition 1.1. Elle comprend la première édition (2009-06) [documents 65E/127/FDIS et 65E/140/RVD] et son amendement 1 (2016-06) [documents 65E/336/CDV et 65E/395A/RVC]. Le contenu technique est identique à celui de l'édition de base et à son amendement.

Cette version Finale ne montre pas les modifications apportées au contenu technique par l'amendement 1. Une version Redline montrant toutes les modifications est disponible dans cette publication.

La Norme internationale IEC 62453-303-1 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Chaque partie de la série IEC 62453-3xy est destinée à être lue conjointement à l'IEC 62453-2.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62453, sous le titre général *Spécification des interfaces des outils des dispositifs de terrain (FDT)*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de la publication de base et de son amendement ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "<i>colour inside</i>" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

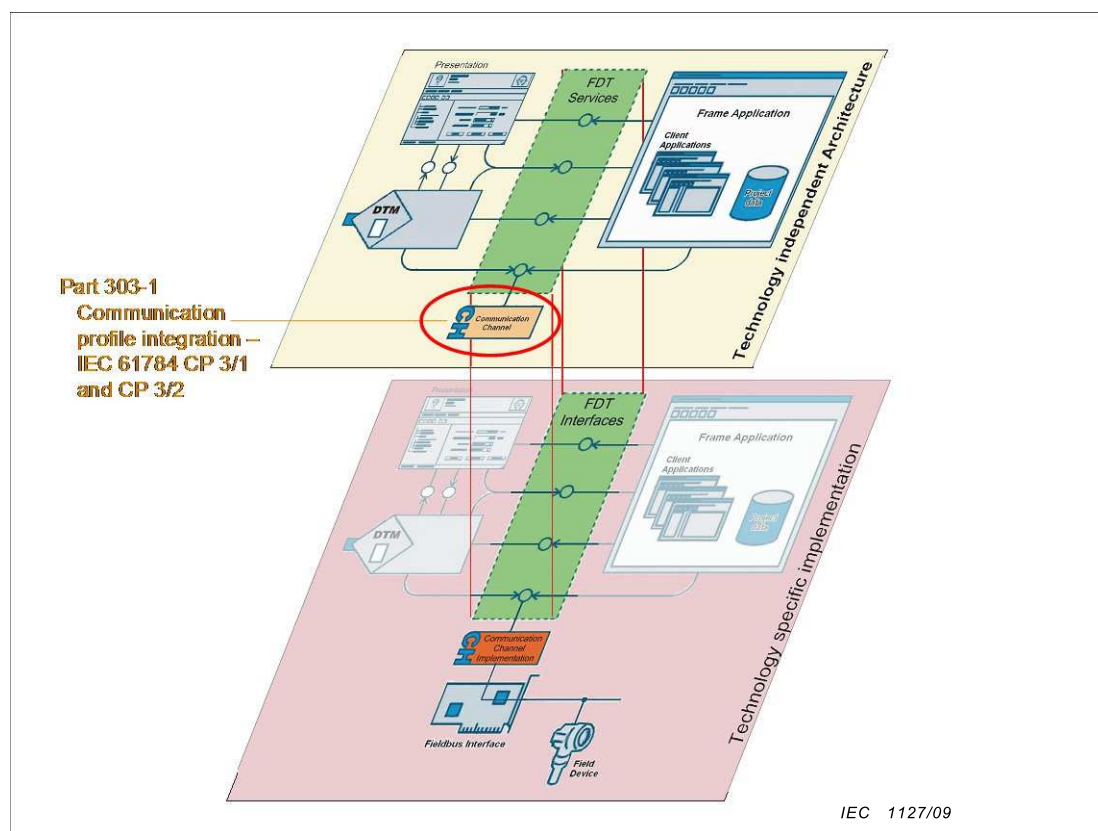
INTRODUCTION

La présente partie de l'IEC 62453 est une spécification d'interface à l'attention des développeurs de composants FDT (Field Device Tool) pour le contrôle de fonctions et l'accès aux données au sein d'une architecture client/serveur. La spécification résulte d'un processus d'analyse et de conception pour développer des interfaces normalisées et faciliter le développement de serveurs et de clients par plusieurs fournisseurs qui ont besoin d'interfonctionner sans couture.

Avec l'intégration des bus de terrain dans des systèmes de commande, il existe quelques autres tâches qu'il est nécessaire d'accomplir. Outre les outils spécifiques à un bus de terrain et à un dispositif, il existe la nécessité d'intégrer ces outils dans des outils d'études ou de planification de plus haut niveau à l'échelle d'un système. En particulier, pour l'utilisation dans des systèmes de commande étendus et hétérogènes, généralement dans le secteur de l'industrie de transformation, la définition sans ambiguïté d'interfaces d'étude qui soient faciles à utiliser pour tous ceux qui sont impliqués a une grande importance.

Un composant logiciel spécifique à un dispositif, appelé DTM (Device Type Manager, «gestionnaire de type de dispositif»), est livré par le fabricant de dispositif de terrain avec son dispositif. Le DTM est intégré dans des outils d'études par l'intermédiaire des interfaces FDT définies dans la présente spécification. L'approche à l'intégration est en général ouverte pour toutes les sortes de bus de terrain et, donc, satisfait aux exigences pour intégrer des types différents de dispositifs dans des systèmes de commande hétérogènes.

La Figure 1 montre comment l'IEC 62453-303-1 est alignée dans la structure de la série IEC 62453.



Légende

Anglais	Français
Part 303-1 Communication profile integration – IEC 61784 CP 3/1 and CP 3/2	Part 303-1 Intégration des profils de communication – IEC 61784 CP 3/1 et CP 3/2
Presentation	Présentation
Technology independent architecture	Architecture indépendante vis-à-vis de toute technologie
Technology specific implementation	Mise en œuvre spécifique à une technologie
Frame application	Application cadre
Client application	Application Client
Communication channel	Voie de communication
FDT Service	Services de FDT
FDT Interfaces	Interfaces des FDT
Fieldbus interface	Interface Fieldbus

Figure 1 – Partie 303-1 de la série IEC 62453

SPÉCIFICATION DES INTERFACES DES OUTILS DES DISPOSITIFS DE TERRAIN (FDT) –

Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1 et CP 3/2

1 Domaine d'application

Le Profil de communication 3/1 et le Profil de communication 3/2 (communément appelés PROFIBUS™¹) définissent des profils de communication basés sur l'IEC 61158-2 Type 3, l'IEC 61158-3-3, l'IEC 61158-4-3, l'IEC 61158-5-3 et l'IEC 61158-6-3. Les profils de base CP 3/1 (PROFIBUS DP) et CP 3/2 (PROFIBUS PA) sont définis dans l'IEC 61784-1.

La présente partie de la IEC 62453 donne des informations pour intégrer le protocole PROFIBUS dans la spécification d'interface FDT (IEC 62453-2).

La présente partie de l'IEC 62453 spécifie les services de communication et autres services.

La présente spécification ne contient la spécification FDT ni ne la modifie.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61131-3:2003, *Automates programmables – Partie 3: Langages de programmation*

IEC 61158 (toutes les parties), *Réseaux de communications industriels – Spécifications des bus de terrain*

IEC 61158-2 :2014, *Réseaux de communications industriels – Spécifications des bus de terrain – Partie 2: Spécification et définition des services de couche physique*

IEC 61158-3-3, *Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3 *Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-5-3: *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements*

IEC 61158-6-3, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

¹ PROFIBUS™ est un nom commercial de l'organisation sans but lucratif PROFIBUS Nutzerorganisation e.V. (PNO). Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. La conformité à la présente norme n'exige pas l'utilisation des logos enregistrés pour PROFIBUS™. L'utilisation des logos enregistrés pour PROFIBUS™ exige la permission de PNO.

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

3 Termes, définitions, symboles, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 62453-1 et l'IEC 62453-2 s'appliquent.

3.1.1

module d'interface de bus

module d'un dispositif de terrain qui fournit la connexion au bus de terrain

3.1.2

CP 3/1

profil de communication de CPF3, caractérisé par une émission asynchrone; RS 485 (ANSI TIA/EIA RS-485-A); RS 485-IS en option; fibre plastique; fibre de verre multimode ou fibre de verre monomode; fibre PCF

3.1.3

CP 3/2

profil de communication de CPF3, caractérisé par une émission synchrone; codage Manchester et alimentation par bus (manchester coded and bus powered (MBP)); sécurité intrinsèque (MBP-IS) et faible puissance (MBP-LP) en option

3.2 Symboles et abréviations

Pour les besoins du présent document, les symboles et abréviations donnés dans l'IEC 62453-1 et l'IEC 62453-2 ainsi que les suivants s'appliquent.

ANSI	American National Standards Institute (Institut national américain des normes (http://www.ansi.org))
BIM	Bus Interface Module (Module d'interface de bus)
BMCP	Bus Master Configuration Part (Partie configuration principale de bus)
CFG	Données de configuration utilisées au cours de l'initialisation de l'équipement esclave PROFIBUS
DCS	Distributed Control System (Système de commande distribué)
DP	Decentralized Peripherals (Périphériques décentralisés)
EIA	Electronic Industries Alliance (Alliance des industries électroniques)
FDL	Fieldbus Data Link layer (Couche liaison de données des bus de terrain)
FMA	Fieldbus Management layer (Couche gestion des bus de terrain)
FMS	Fieldbus Message Specification (Spécification de message des bus de terrain)
GSD	General Station Description (Description de station générale)
MBP	Manchester coded Bus Powered (Alimenté par bus et à codage Manchester)
I&M	Identification and maintenance functions (Fonctions d'identification et maintenance)
PA	Process Automation (Automatisation de processus)
PCF	Polymer Clad Fibre (Fibre à gaine polymère)
PROFIBUS	Process Field Bus (Bus de terrain de processus)
RS	Radio Sector (Secteur radio)/ Recommended Standard (Norme recommandée)

3.3 Conventions

3.3.1 Noms de type de données et références aux types de données

Les conventions de dénomination et de référencement des types de données sont expliquées dans l'IEC 62453-2, Article A.1

3.3.2 Vocabulaire pour les exigences

Les expressions suivantes sont utilisées pour spécifier des exigences.

Utilisation de "doit" ou "obligatoire"	Aucune exception tolérée.
Utilisation de "il convient de" ou "recommandé"	Forte recommandation. Il peut être légitime, dans des cas particuliers exceptionnels, de s'écarter du comportement décrit.
Utilisation de "conditionnel"	La fonction ou le comportement doit être donné(e), selon des conditions définies.
Utilisation de "peut" ou "facultatif"	La fonction ou le comportement peut être donné(e), selon des conditions définies.

3.3.3 Utilisation de l'UML

Les figures dans le présent document utilise la notation UML telle que définie dans l'Annexe A de l'IEC 62453-1.

4 Catégorie de bus

Les protocoles CP 3/1 et CP 3/2 sont identifiés dans l'élément protocolld du type de données structuré 'fdt:BusCategory' par les identificateurs uniques suivants (Tableau 1):

Tableau 1 – Identificateurs de protocole

Valeur d'identificateur	Nom de Protocolld	Description
036D1497-387B-11D4-86E1-00E0987270B9	'Profibus DP/V0'	Prise en charge du protocole Profibus DP V0
036D1499-387B-11D4-86E1-00E0987270B9	'Profibus DP/V1'	Prise en charge du protocole Profibus DP V1

Les protocoles CP 3/1 et CP 3/2 utilisent les identificateurs uniques suivants dans les membres physicalLayer au sein du type de données PhysicalLayer (Tableau 2):

Tableau 2 – Identificateurs de couche physique

Valeur d'identificateur	Nom	Description
036D1590-387B-11D4-86E1-00E0987270B9	MBP	IEC 61158-2 (MBP, Profibus PA)
036D1591-387B-11D4-86E1-00E0987270B9	RS485	IEC 61158-2:2014, Article 22 (RS485, PROFIBUS DP)
036D1592-387B-11D4-86E1-00E0987270B9	Fiber Optic (Fibres optiques)	IEC 61158-2:2014, Article 23 (Câble à fibres optiques, PROFIBUS DP)
036D1593-387B-11D4-86E1-00E0987270B9		Ethernet (déconseillé, ne pas utiliser)

Le Tableau 32 définit quelle DataLinkLayer doit être utilisée en combinaison avec les valeurs de BusCategory définies dans le Tableau 32.

Tableau 32 – Identificateurs de couche DataLink

Valeur d'identificateur	Nom	Description
50A21B35-7EE7-4999-8174-70396929C0B4	PROFIBUS FDL	PROFIBUS FDL
CDF338DC-E9A3-4D13-91AC-60A43DCB2904	PROFIBUS FMA1/2	PROFIBUS FMA1/2

5 Accès aux données d'instances et de dispositifs

5.1 Objets "Process Channel" fournis par le DTM

Il convient que l'ensemble minimal de données fournies soit: les valeurs de processus modélisées comme des objets de voies, y compris les plages et les changements d'échelle.

5.2 Services DTM pour accéder aux données d'instances et de dispositifs

Le services InstanceDataInformation et DeviceDataInformation doivent donner accès à au moins tous les paramètres du Physical Block et le statut et la valeur Out des blocs fonctionnels doivent être exposés.

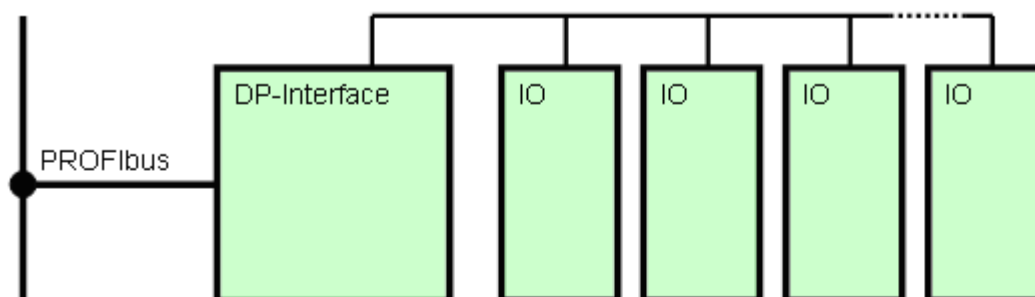
Conformément à l'IEC 62453-2, au moins un jeu d'informations sémantiques (un par protocole de bus de terrain pris en charge) doit être fourni pour chaque objet de données accessibles, en utilisant le type de données général 'SemanticInformation'. Le type de données 'applicationDomain' correspondant doit avoir une valeur définie pour Profibus et le type de données 'semanticId' doit avoir une valeur appropriée, telle que spécifiée dans le Tableau 9.

6 Comportement spécifique à un protocole

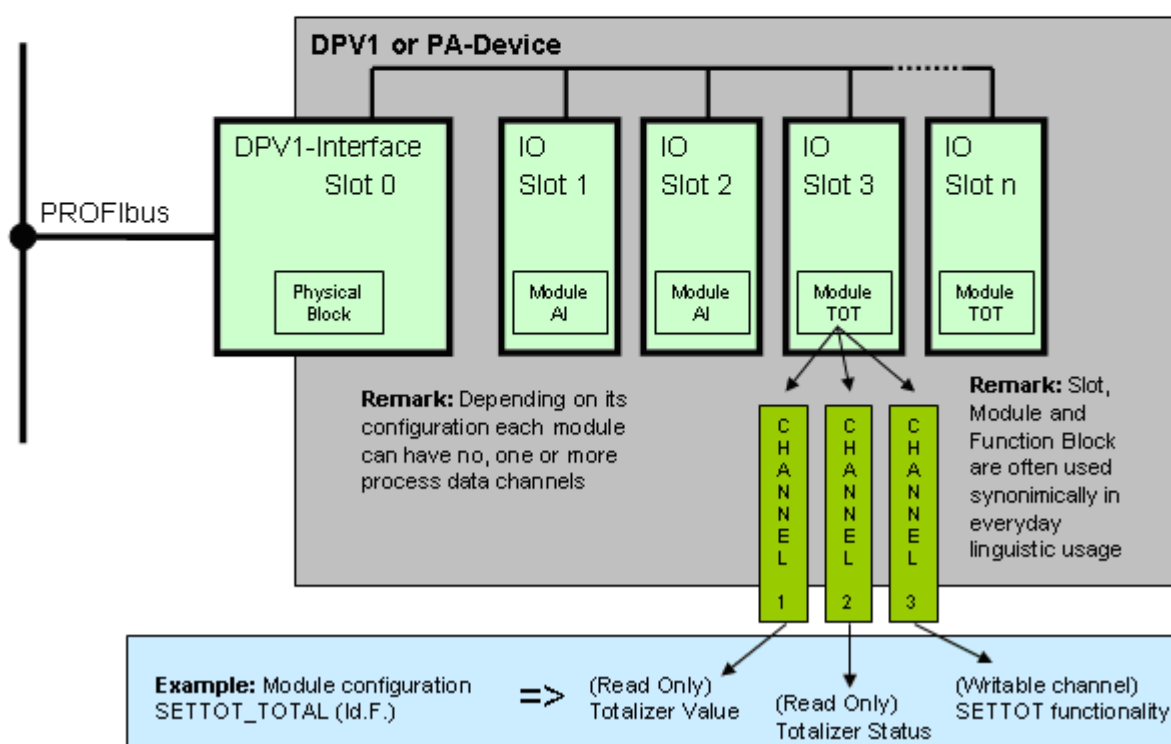
6.1 Modèle de dispositif PROFIBUS

FDT étend le modèle de dispositif PROFIBUS en utilisant des Process Channel (voies de processus) pour la description des valeurs E/S (voir Figure 2).

Classical View of PROFIBUS device



PROFIBUS notations from a monolithic DPV1 or PA device DTMs point of view



IEC 1128/09

Légende

Anglais	Français
Classical View of PROFIBUS device	Vue classique d'un dispositif PROFIBUS
PROFIBus	PROFIBus
DP-Interface	Interface DP
PROFIBUS notation from a monolithic DPV1 or PA device DTMs point of view	Notation PROFIBUS, du point de vue des DTM de dispositif DPV1 ou PA monolithiques
DPV1 or PA device	Dispositif DPV1 ou PA
IO Slot 0	Baie 0 E/S
Physical block	Bloc physique
Module	Module
Remark: Depending on its configuration each module can have no, one or more process data channels.	Remarque: Selon sa configuration, chaque module peut avoir zéro, une ou plusieurs voies de données de processus.
Channel	Voie
Remark: Slot, Module and Function Block are often used synonymically in everyday linguistic usage	Remarque: Slot («baie»), Module et Fonction («Bloc fonctionnel») sont souvent utilisés comme synonymes dans le langage de tous les jours.

Anglais	Français
Example: Module configuration	Exemple: Configuration de module
(Read Only) Totalizer Value	(Lecture seule) Valeur de totaliseur
(Read Only) Totalizer Status	(Lecture seule) Statut de totaliseur
(Writable Channel) SETTOT functionality	(Voie inscriptible) Fonctionnalité SETTOT
IO	E/S (entrée/sortie)

Figure 2 – Modèle de dispositif PROFIBUS FDT

6.2 Configuration et paramétrisation des dispositifs PROFIBUS

6.2.1 Généralités

Dans un outil de configuration GSD, l'utilisateur définit la configuration et établit les paramètres appropriés pour les modules. L'outil de configuration crée la chaîne de configuration et la chaîne de paramètres qui sont utilisées pour établir correctement l'esclave.

Avec FDT, la configuration et la paramétrisation des dispositifs ne sont plus exécutées seulement par un composant central; elles se sont partiellement déplacées dans les DTM. Un DTM est chargé de donner des informations de configuration et de paramétrisation pour le maître cyclique qui met les esclaves PROFIBUS en fonctionnement.

Un DTM est utilisé pour réajuster un dispositif de terrain à son application spécifique. Au sein de PROFIBUS, il y a trois différents aspects du réajustement:

- paramétrisation: usr prm data (utilisée dans le service SET_PRM de PROFIBUS pour établir la communication cyclique et le comportement spécifique du dispositif);
- paramétrisation d'application: paramètres spécifiques à une application (émis au moyen des services read/write (lecture/écriture) acycliques de PROFIBUS);
- configuration: données de configuration (utilisées dans le service CHK_CFG de PROFIBUS pour la définition du format et de la longueur des données d'entrée/sortie qui sont émises dans le cadre de la communication cyclique).

La paramétrisation d'application émise par la communication acyclique ne relève pas du domaine d'application du présent document. Dans le présent document, le terme "paramétrisation" représente la paramétrisation des communications (SET_PRM).

6.2.2 DTM monolithique pour un dispositif PROFIBUS modulaire

Un DTM monolithique est un DTM seul qui représente le dispositif complet avec son module d'interface de bus (BIM pour «Bus Interface Module») et ses modules E/S (d'entrée/sortie). En général, un tel DTM propose un dialogue de configuration (objet "presentation") qui permet la définition des BIM et modules utilisés. Le dialogue de configuration peut être disponible par le biais de la fonction FDT normalisée "Configure" (voir [1] 4.3 Configuration de fonctionnement).

Ce ne sont pas tous les dispositifs PROFIBUS qui exigent un dialogue de configuration. C'est pourquoi tous les DTM ne fournissent pas nécessairement la fonction "Configure". Cela ne vaut que pour les dispositifs PROFIBUS non modulaires si les Usr_Prm-Data ne peuvent pas être modifiées.

Le dialogue de configuration permet de changer les données seulement en mode hors ligne si le jeu de données peut être verrouillé.

6.2.3 DTM modulaire pour un dispositif PROFIBUS modulaire

Des DTM distincts représentent le BIM (DTM de dispositif) et les modules E/S particuliers (DTM de modules). L'effort pour développer un tel DTM modulaire est normalement plus grand que dans le cas d'un DTM monolithique, car:

- un protocole privé doit être mis en œuvre entre le BIM et les modules E/S pour assurer que seul un DTM de module peut être ajouté au DTM de BIM. Cela nécessite un ID de protocole propre et l'adaptation/création d'une communication;
- dans certains cas, des interfaces privées complémentaires sont nécessaires pour échanger des informations entre le DTM de dispositif pour BIM et les DTM de modules.

La mise en œuvre d'un DTM modulaire procure les avantages suivants:

- le projet représente la structure des dispositifs;
- l'utilisateur est à même d'accéder à des informations relatives au module, et ce, directement en fonction du DTM de module ;
- la spécification FDT définit un mécanisme pour identifier les DTM. Avec ces mécanismes, il est possible d'assurer la prise en charge pour le balayage des modules en dessous du BIM et de générer automatiquement la topologie;
- la prise en charge d'un nouveau type de BIM ou de module E/S exige un DTM supplémentaire "seulement" et n'affecte pas les composants existants. Cela peut se traduire par un effort d'essais réduit.

Les données de configuration pour établir la configuration PROFIBUS doivent être fournies par le DTM de dispositif (représentant le BIM). Ces données de configuration peuvent être générées à partir d'informations de DTM enfants instanciés et en utilisant un dialogue de configuration.

Il convient que des DTM modulaires soient fournis pour des dispositifs modulaires (par exemple: un opérateur d'installation peut ajouter/retirer des modules). Les DTM monolithiques sont utilisés pour représenter des dispositifs qui ne présentent aucune modularité (les dispositifs PA, par exemple).

6.3 Support pour la configuration DPV0

Un esclave PROFIBUS est configuré par un maître cyclique et communique par PROFIBUS DP. En plus de cela, l'esclave peut prendre en charge la communication DPV1.

Un DTM de passerelle pour un esclave PROFIBUS n'est pas tenu de fournir la communication pour le schéma de communication DPV0. Par exemple, il existe un système E/S distant avec des modules HART. Il peut avoir un DTM de passerelle qui nécessite le protocole DPV1 et fournit le protocole HART (défini dans le document d'informations et dans le document de paramètres). Cela permet aux DTM de dispositifs HART de communiquer avec leurs dispositifs par le truchement du DTM de passerelle et par le DTM de communication pour DPV1. Respectant la spécification, le DTM de passerelle délivre les documents de paramètres de voies pour les protocoles DPV1 et HART à la fois. Le protocoleId est un membre du type de données NetworkManagementInfo.

Les voies de processus (Process Channels) doivent fournir les documents de Paramètres de Voies (ChannelParameter) pour DPV1, y compris toutes les informations permettant l'intégration dans le système de commande (par exemple: DPAddress de la valeur E/S, si disponible).

6.4 Esclaves PROFIBUS fonctionnant sans un maître PROFIBUS cyclique

Dans la plupart des cas, un esclave PROFIBUS est configuré et paramétré par un dispositif maître PROFIBUS cyclique. Donc, un dispositif maître en fonctionnement dans le réseau est requis.

Certains esclaves sont capables de permettre une communication acyclique sans un maître cyclique en fonctionnement. Notamment dans le cas de la fonctionnalité de passerelle, ceci constitue un insigne avantage car ils permettent la paramétrisation de dispositifs de terrain qui leur sont connectés en utilisant un maître bus acyclique. Ainsi, les spécialistes des instruments peuvent également travailler avec des dispositifs de terrain même si le dispositif de commande ne fonctionne pas encore.

Si un maître lance la communication, ces dispositifs commencent à détecter la vitesse et les réglages de bus pour réagir de façon appropriée. Cela peut prendre un certain temps.

Ci-après, il est décrit deux cas qu'un utilisateur peut garder à l'esprit lorsqu'il travaille avec de tels dispositifs.

Cas d'utilisation 1:

L'utilisateur accomplit un balayage de réseau. Le DTM de communication tente de lire des données de diagnostic au moyen d'une demande GetDiagnose mais ne reçoit pas de réponse. Le dispositif n'est pas détecté par le DTM de communication. Cela se produit surtout lorsque le dispositif a une basse adresse PROFIBUS. La raison en est que le dispositif n'a pas achevé la détection de la vitesse du bus/ des réglages du bus comme cela a été demandé pour les données de diagnostic. La solution est de donner à ces dispositifs une adresse PROFIBUS plus haute.

Cas d'utilisation 2:

L'utilisateur tente de connecter un dispositif de terrain relié à la passerelle qui prend en charge DPV1 sans un maître cyclique en fonctionnement. Cela peut conduire à un message d'erreur car le dispositif passerelle n'a pas achevé la détection de la vitesse du bus/des réglages du bus comme cela a été demandé pour une connexion. L'utilisateur doit donc tenter la connexion de nouveau. Cela se produit seulement dans de très rares cas.

6.5 Informations relatives à PROFIBUS d'un DTM esclave

6.5.1 Généralités

Les informations utilisées par un dispositif maître cyclique pour établir correctement un réseau PROFIBUS et permettre une communication cyclique entre le système de commande et les dispositifs esclaves sont fournies par un DTM dans

- la partie configuration principale de bus (BMCP);
- les informations GSD;
- la topologie interne ;
- les voies de processus.

Un DTM d'un esclave PROFIBUS doit délivrer ces parties d'informations relatives à PROFIBUS pour être intégré dans un système d'ingénierie FDT. Dans les prochains paragraphes, il est donné une description plus détaillée de la manière de générer et de la manière de fournir ces informations. Cela dépend de la sorte de DTM (voir 6.2 Configuration et paramétrisation des dispositifs PROFIBUS).

6.5.2 Bus Master Configuration Part (partie configuration principale de bus (BMCP))

6.5.2.1 Introduction à la BMCP

La BMCP d'une seule instance de DTM prise séparément décrit les données réelles de paramètres et de configuration de l'esclave PROFIBUS correspondant. Chaque DTM représentant un dispositif d'esclave PROFIBUS doit fournir une Bus Master Configuration Part. La BMCP est fournie dans le membre busMasterConfigurationPart du type de données

NetworkManagementInfo. Ces informations sont obtenues en appelant le service NetworkManagementInfoRead.

La BMCP inclut des informations relatives à la configuration et aux paramètres pour l'esclave. La BMCP est fournie par le DTM et est indispensable pour générer la configuration de maître.

La BMCP contient des données susceptibles de changer durant la configuration de maître. Cela signifie que la BMCP peut être modifiée et retournée au DTM d'esclave en appelant NetworkManagementInfoWrite. Un DTM d'esclave doit accepter les nouvelles informations et recalculer les paramètres de configuration/internes pour s'accorder avec la nouvelle BMCP.

DTM doit vérifier si, oui ou non, les nouvelles valeurs sont conformes aux capacités du dispositif. L'appel de NetworkManagementInfoWrite sera refusé si le dispositif ne peut pas traiter les nouvelles valeurs.

6.5.2.2 Création de la BMCP

Le présent paragraphe explique dans le détail la signification des éléments individuels de la BMCP.

La BMCP peut être créée à partir des informations GSD d'un dispositif PROFIBUS.

La BMCP est divisée en quatre parties qui sont expliquées dans les paragraphes suivants.

Les explications utilisent les mots clés de GSD et font référence à la spécification PROFIBUS. Voir aussi le tableau avec la BMCP complète en 6.5.2.2.1.

6.5.2.2.1 Partie 1: De Slave_Para_Len à l'Octet 15

La première partie consiste en un jeu fixe d'octets décrits dans le tableau ci-dessous.

Tableau 3 – BMCP Partie 1 – Configuration générale

Octet	Nom	Défini dans	Remarque
0	Slave_Para_Len	[5] 6.2.12.1	Longueur de la BMCP, y compris cette valeur
1			
2	SI_Flag	[5] 6.2.12.2	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0: réservé, • Bit 1: Extra_Alarm_SAP, • Bit 2: DPV1_Data_Types, • Bit 3: DPV1_Supported, • Bit 4: Publisher_Enable, • Bit 5: Fail_Safe. Les bits suivants ne sont pas basés sur des valeurs GSD: • Bit 6: New_Prm, • Bit 7: Active
3	Slave Type	[5] 6.2.12.3	La valeur est 0 (= DP-Slave, c'est-à-dire esclave DP)
4	Max_Diag_Data_Len	[5] 6.2.12.4	La valeur GSD suivante est utilisée [6]: Max_Diag_Data_Len
5	Max_Alarm_Len	[5] 6.2.12.5	
6	Max_Channel_Data_Len	[5] 6.2.12.6	Ce champ définit la quantité de données pouvant être transférées entre esclave et maître. Dans ce cas, le maximum de ces valeurs GSD [6] doit être calculé: • Max_Data_Len, • C1_Max_Data_Len plus 4 octets (Function Num, Slot_Number, Length)
7	Diag_Update_Delay	[5] 6.2.12.7	La valeur GSD suivante est utilisée [6]:Diag_Update_Delay
8	Alarm_Mode	[5] 6.2.12.8	
9	Add_SI_Flag	[5] 6.2.12.9	
10	C1_Timeout	[5] 6.2.12.10	La valeur GSD suivante est utilisée [6]:C1_Response_Timeout
11			
12	Reserved		
13			
14			
15			

6.5.2.2.2 Partie 2: Données de paramètres

La Partie 2 de la BMCP est constituée des données de paramètres d'esclave.

Tableau 4 – BMCP Partie 2 – Données de paramètres

Octet	Nom	Défini dans	Remarque
16	Prm_Data_Len	[5] 6.2.12.11	Longueur des Données de paramètres, y compris cette valeur
17			
18	Station Status	[5] 6.2.4.1	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0 à 2: réservés, • Bit 3: WD_On, • Bit 4: Freeze_Req, • Bit 5: Freeze_Req, • Bit 6, 7(Lock/Unlock Request c'est-à-dire Demande de verrouillage/déverrouillage): [5] Tableau 402
19	WatchDog1	[5] 6.2.4.2	Ces valeurs (WD_Fact_1 et WD_Fact_2) dépendent du débit en bauds. Il convient qu'un maître établisse ces valeurs et que des esclaves traitent de nouvelles valeurs.
20	WatchDog2	[5] 6.2.4.3	
21	Min Tsdr	[5] 6.2.4.4	La valeur par défaut est un temps de 11 bits [1]

Octet	Nom	Défini dans	Remarque
22	Ident_Number	[5] 6.2.3.5	La valeur GSD suivante est utilisée [6]: Ident_Number
23			
24	Group_Ident	[5] 6.2.4.5	Indique l'affectation de groupe de l'esclave sous une forme codée au niveau du bit
25	DPV1_Status_1	[5] 6.2.4.7	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0, 1: réservés, • Bit 2: WD_Base_1ms, • Bit 3 à 5: réservés, • Bit 6: Fail_Safe, • Bit 7: DPV1_Enable, Ces bits rendent compte des capacités des esclaves au maître et sont modifiés par le maître MS0 suivant ses capacités.
26	DPV1_Status_2	[5] 6.2.4.8	Les valeurs GSD suivantes sont utilisées [6]: • Bit 0: Check_Cfg_Mode, • Bit 1: réservé, • Bit 2: Enable_Update_Alarm, • Bit 3: Enable_Status_Alarm, • Bit 4: Enable_Manufacturer_Specific_Alarm, • Bit 5: Enable_Diagnostic_Alarm, • Bit 6: Enable_Process_Alarm, • Bit 7: Enable_Pull_Plug_Alarm, Ces bits rendent compte des capacités des esclaves au maître et sont modifiés par le maître MS0 suivant ses capacités.
27	DPV1_Status_3	[5] 6.2.4.9	• Bit 0 à 2: - , • Bit 3: Prm Structure, • Bit 4: IsoM_Req, • Bit 5-6: réservés, • Bit 7: Prm_Cmd
28	User_Prm_Data		À partir de l'Octet 28 de la BMCP, un DTM doit insérer des données de paramètres utilisateur supplémentaires. Les informations sont données dans le fichier GSD au moyen de la valeur de User_Prm_Data
...			
NOTE 1 Il est possible qu'un module n'ait pas de données de paramètres utilisateur. Dans ce cas, aucune chaîne de paramètres n'est insérée dans la BMCP pour ce module.			
NOTE 2 Certains esclaves ont un certain nombre de modules fixes en plus de la BIM. Même si ces modules n'apparaissent pas dans le dialogue de configuration ou ne sont pas représentés comme des DTM de modules, ils doivent être pris en considération dans la BMCP s'ils ont des paramètres.			
NOTE 3 Les trois octets de statut DPV1 sont définis par le maître MS0.			

6.5.2.2.3 Partie 3: Données de configuration

Les données de configuration sont fournies comme partie 3:

Tableau 5 – BMCP Partie 3 – Données de configuration

Octet	Nom	Source	Remarque
16+Prm_Data_Len	Cfg_Data_Len	[5] 6.2.12.13	Longueur, y compris cette valeur
16+Prm_Data_Len + 1			
16+Prm_Data_Len +2	Cfg_Data		Données de configuration (si disponibles)
...			

À partir de la position d'octets (16+Prm_Data_Len +2), les chaînes de configuration pour le BIM et les modules sont données dans l'ordre croissant. Les informations sont données dans le fichier GSD par le truchement du module de valeur.

NOTE 1 Certains esclaves ont un certain nombre de modules fixes en plus de la BIM. Même si ces modules n'apparaissent pas dans le dialogue de configuration ou ne sont pas représentés comme des DTM de modules, ils doivent être pris en considération dans la BMCP s'ils ont des paramètres.

NOTE 2 Des baies vides doivent être prises en considération dans les données de configuration. Se référer au fichier GSD pour savoir quelle chaîne de configuration est à utiliser.

6.5.2.2.4 Partie 4: Table d'adresses et paramètres utilisateur d'esclave

Dans la Partie 4 de la BMCP, il est fourni la section table d'adresses et paramètres utilisateur d'esclave.

Tableau 6 – Partie 4: Table d'adresses et paramètres utilisateur d'esclave

Octet	Nom	Source	Conseils
16 + Prm_Data_Len + Cfg_Data_Len	Add_Tab_Len	[5] 6.2.12.15	Longueur, y compris cette valeur
16 + Prm_Data_Len + Cfg_Data_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len	Slave_User_Data_Len	[6] 6.2.12.21	Longueur, y compris cette valeur
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 2	Slave_User_Data		Paramètre utilisateur d'esclave (si disponible)
...			

Après Add_Tab_Len, les données de la table d'adresses sont insérées dans la BMCP. Si le maître ne prend pas en charge cette table, le paramètre Add_Tab_Len est mis à 2 et aucune donnée n'est insérée.

Après Slave_UserDataLen, les données utilisateur d'esclave sont insérées dans la BMCP. Si le maître ne prend pas en charge ces données, le paramètre Slave_User_Data_Len est mis à 2 et aucune donnée n'est insérée.

Tableau 7 – Partie 4: Extended Prm data (Données de paramètres étendues)

Octet	Nom	Source	Conseils
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len	Ext_Prm_Data_Len	[5] 6.2.12.23	Longueur, y compris cette valeur
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len + 1			

Après Ext_Prm_Data_Len, les données de paramètres étendus sont insérées dans la BMCP. Si le maître ne prend pas en charge ces données, le paramètre Ext_Prm_Data_Len est mis à 2 et aucune donnée n'est insérée.

6.5.2.2.5 Jeu complet de paramètres de bus esclave

Toutes les valeurs sont fournies à partir du DTM d'esclave. Il est de la responsabilité du DTM d'esclave d'être compatible avec la GSD d'esclave. Le DTM de maître pourrait changer certaines de ces valeurs initiales envoyées par le DTM d'esclave si elles dépendent des capacités du maître.

Exemple

Au sein du fichier GSD, il est énoncé que le dispositif prend en charge le mode Freeze ("gel") par le mot-clé "Freeze_Mode_supp". Le maître établit la valeur "Freeze_Mode_Req" au sein du jeu de paramètres de bus esclave ("Slave Bus Parameter Set"), car seul le maître sait s'il prend en charge ce mode ou non.

Le tableau suivant explique quel composant est la source des valeurs de paramètres ("Paramètres fournis par"). Certaines de ces valeurs peuvent être modifiées par le système ou par interaction avec l'utilisateur. Pour ces valeurs, il est indiqué quel composant est autorisé à changer les valeurs ("Configuration par"). Si possible, les valeurs par défaut pour les paramètres sont définies ("Valeur par défaut").

Tableau 8 – BMCP complète

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
0	Slave_Para_Len	Longueur de la BMCP, y compris cette valeur	Calculé par le DTM d'esclave	-	-
1					
2	SI_Flag	Bit codé. La signification des bits:	-	-	-
		Bit 0: réservé	-	-	0
		Bit 1: Extra_Alarm_SAP	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Extra_Alarm_SAP_sup p au sein de GSD
		Bit 2: DPV1_Data_Types	Le DTM d'esclave montre quels types de données sont utilisés par le DTM d'esclave.	DTM de maître	DPV1_Data_Types au sien de la GSD d'esclave
		Bit 3: DPV1_Supported	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	DPV1_Slave au sein de la GSD
		Bit 4: Publisher_enable	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Publisher_supp au sein de GSD
		Bit 5: Fail_Safe	Le DTM d'esclave peut montrer par ce fanion que l'esclave a besoin de sécurité intégrée (Fail_Safe ou Fail_Safe_Required de la GSD)	DTM de maître	Valeur au sein de la GSD d'esclave
		Bit 6: New_Prm	Si le DTM d'esclave établit cette valeur, il indique qu'il convient que le maître MS0 établisse le paramètre d'esclave modifié. (voir [5] 6.2.12.2)	-	0
		Bit 7: Active	Si le DTM esclave a fini la configuration de module, le mettre à 1.	-	0
3	Slave Type	0 = Esclave DP	L'esclave de DTM montre qu'il s'agit d'un esclave DP.	-	0
4	Max_Diag_Data_Len		L'esclave DTM fournit ces informations par le paramètre Max_Diag_Data_Len dans la GSD	-	Valeur au sein de la GSD d'esclave
5	Max_Alarm_Len	Longueur de la structure d'alarmes (voir [5] Tableau 396)	DTM d'esclaves, conditions: Un des mots-clés de GSD:- Diagnostic_Alarm_supp - Process_Alarm_supp - Pull_Plug_Alarm_supp - Status_Alarm_supp - Updata_Alarm_supp - Manufacturer_Specific_Alarm_supp OR - Alarm_Type_Mode_supp	-	Est calculée sur la base des différentes valeurs GSD.
6	Max_Channel_Data_Len		DTM d'esclave: Ce champ définit la quantité de données pouvant être transférées entre esclave et maître. Dans ce cas, le maximum de ces valeurs doit être calculé et établi par le DTM d'esclave. Règle pour le calcul: Max_Data_Len C1_Max_Data_Len plus 4 octets (Function Num, Slot_Number, Length)	-	Valeur au sein de la GSD d'esclave
7	Diag_Update_Delay		DTM d'esclave:	-	GSD d'esclave [6] Diag_Update_Delay

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
8	Alarm_Mode		DTM d'esclave:	-	GSD d'esclave [6] Alarm_Sequence_Mode_Count
9	Add_SI_Flag	Bit codé. Suivant la signification des bits:			
		Bit 0: NA_To_Abort		DTM de maître	0
		Bit 1: Ignore_Aclr		DTM de maître	0
		Bit 2 à Bit 7: réservés			0
10	C1_Timeout	La valeur GSD suivante est utilisée [6]: C1_Response_Timeout	DTM d'esclave:		GSD d'esclave [6] C1_Response_Timeout
11					
12	Réservé				0
13	Réservé				0
14	Réservé				0
15	Réservé				0
16	Prm_Data_Len	Longueur des Données de paramètres, y compris cette valeur	Calculé (ou peut-être fixé) par le DTM d'esclave		
17					
	PRM_DATA				
18	Prm_Data (Station Status)	Bits 0 à 2: réservés		0	
		Bit 3: WD_On	Le maître MS0 définit que le chien de garde (watchdog) est utilisé ou non. Si le chien de garde est activé, le maître doit aussi établir WD_Fact_1 et WD_Fact_2	DTM de maître	0
		Bit 4: Freeze_Req	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Freeze_Mode_supp au sein de GSD
		Bit 5: Sync_Req	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	(Sync_Mode_supp au sein de GSD)
		Bits 6, 7: Demande de verrouillage/déverrouillage [5] Tableau 402	DTM de maître	0	

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
19	Prm_Data (WatchDog1) WD_Fact_1	Ces valeurs dépendent du débit en bauds. Il convient qu'un maître établisse ces valeurs et que des esclaves traitent de nouvelles valeurs. Prendre en considération au sein de Usr_Prm_Data (DPV1_Status_1, Bit 2 WD_Base_1ms)		DTM de maître	1
20	Prm_Data (WatchDog2) WD_Fact_2	Voir Octet 19, Watchdog_Time = 10 ms * WD_Fact_1 * WD_Fact_2 OR Watchdog_Time = 1 ms * WD_Fact_1 * WD_Fact_2		DTM de maître	1
21	Prm_Data (Min Tsdr)	La valeur par défaut est un temps de 11 bits [1].		DTM de maître	11
22	Prm_Data (Ident_Number)		SlaveDTM	-	GSD d'esclave [6] Ident_Number
24	Prm_Data (Group_Ident)	Cela signifie que l'esclave n'est pas affecté à un groupe pour commande globale. Il est établi par le maître		DTM de maître	0
25	Prm_Data (DPV1_Status_1)	Bits 0, 1: réservés			0
		Bit 2: WD_Base_1ms (Voir Octets 19 et 20)	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître Dans quelques rares cas, il est nécessaire de mettre la valeur à 1.	WD_Base_1ms_supp au sein de GSD
		Bits 3 à 5: réservés			0
		Bit 6: Fail_Safe	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	(Fail_safe au sein de GSD)

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
		Bit 7: DPV1 Enable	DTM d'esclave:	DTM de maître	GSD d'esclave (Si le minimum de GSD comprend l'une des valeurs établies suivantes: C1_Read_Write_supp = 1 ou Diagnostic_Alarm_supp = 1 ou Process_Alarm_supp = 1 ou Pull_Plug_Alarm_supp = 1 ou Status_Alarm_supp = 1 ou Update_Alarm_supp = 1 ou Manufacturer_Specific_Alarm_supp = 1. L'esclave prend en charge DPV1 et il convient alors que DPV1_Enable soit 1

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
26	Prm_Data (DPV1_Status_2)	Bit 0: Check_Cfg_Mode			
	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Check_Cfg_Mode au sein de GSD		
		Bit 1: réservé	-	-	0
		Bit 2: Enable_Update_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Update_Alarm_supp au sein de GSD
		Bit 3: Enable_Status_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Status_Alarm_supp au sein de GSD
		Bit 4: Enable_Manufacturer_Specific_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Manufacturer_Specific_Alarm_supp au sein de GSD
		Bit 5: Enable_Diagnostic_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Diagnostic_Alarm_supp au sein de GSD
		Bit 6: Enable_Process_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Process_Alarm_supp au sein de GSD
		Bit 7: Enable_Pull_Alarm	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Pull_Plug_Alarm_supp au sein de Slave GSD
27	Prm_Data (DPV1_Status_3)	Bit 0-2: Alarm_Mode	DTM d'esclave:	DTM de maître	Données au sein de la GSD d'esclave
		Bit 3: Prm_Structure.	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.		Prm_Block_Structure_supp au sein de GSD
		Bit 4: IsoM_Req	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	Isochron_Mode_supp au sein de GSD
		Bit 5-6: réservé	-	-	0
		Bit 7: Prm_Cmd	Le DTM d'esclave montre avec ce bit que la caractéristique est prise en charge.	DTM de maître	PrmCmd_supp au sein de GSD
28 à ...	Prm_Data(Usr_Prm_Data)		Calculé par le DTM d'esclave	DTM d'esclave:	Données au sein de GSD
X1	Cfg_Data_Len	Longueur, y compris cette valeur	Calculé par le DTM d'esclave	-	Données au sein de GSD
X1+1					
X1+2 à ...	Cfg_Data	Cfg_Data	DTM d'esclave, Dépend de la configuration des modules	DTM d'esclave:	Données au sein de la GSD d'esclave (Module, EndModule)
X2	Add_Tab_Len	Longueur, y compris cette valeur	Calculée par le DTM de communication		

Octet	Nom	Note	Paramètre fourni par	Configuration par	Valeur par défaut
X2 + 1					
X2 + 2 à ...	Add_Tab_Data	Table d'affectation d'adresses (seulement pour les maîtres DPV0)	Calculée par le DTM de communication		
X3	Slave_User_Data_Len	Longueur, y compris cette valeur	Calculée par le DTM de communication		
X3 + 1					
X3 + 2 à ...	Slave_User_Data	Slave_User_Data	Calculée par le DTM de communication		Données au sein de GSD
X4 + 1	Ext_Prm_Data_Len	Longueur, y compris cette valeur	Calculé par le DTM d'esclave	-	
X4 + 2					
X4	Ext_Prm_Data	Données de paramètres étendues	DTM d'esclave:	-	Données au sein de GSD
NOTE X1 = 16+Prm_Data_Len X2 = 16 + Prm_Data_Len + Cfg_Data_Len = 16 + x1 + Cfg_Data_Len X3 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len = X2 + Add_Tab_Len X4 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len = X3 + Slave_User_Data_Len					

6.5.2.3 Modification de la BMCP

6.5.2.3.1 Propagation des modifications

La BMCP inclut des données de paramètres et de configuration. Le DTM d'esclave ou la FDT Frame Application ("application-cadre FDT") peut modifier la BMCP.

Le système doit assurer que le DTM de communication représentant le maître PROFIBUS est conscient de ces modifications. Cela est accompli en envoyant l'événement InstanceDataChanged depuis le DTM esclave afin de rapporter le changement de la BMCP. Il convient que tous les changements soient rapportés dès que possible, mais pas avant que les changements ne soient persistants. La Frame Application informe le DTM parent, par OnChildInstanceDataChanged, que des paramètres d'un enfant ont été modifiés. Le DTM de communication peut alors récupérer la nouvelle BMCP du DTM d'esclave.

6.5.2.3.2 Quand il est autorisé à modifier la BMCP

Conformément à la spécification, il est autorisé de changer les paramètres d'un DTM en commençant à partir de l'état "configuré" (voir IEC 62453-2).

La BMCP peut être modifiée plusieurs fois, mais seulement si le DTM est en mode hors ligne et si le jeu de données peut être verrouillé.

Si un DTM d'esclave souhaite modifier des paramètres en mode en ligne, il doit utiliser les demandes de transaction DPV0 ou DPV1. S'il n'y a aucun moyen de changer les paramètres par des demandes de transaction, le DTM doit désactiver la configuration et la paramétrisation dans l'état en ligne.

6.5.2.3.3 Données de paramètres

Si l'utilisateur modifie des paramètres du BIM ou d'un module (Par exemple par l'intermédiaire de l'interface utilisateur) et cela affecte la BMCP, le DTM doit mettre à jour la BMCP. En plus de cela, il doit demander une sauvegarde et informer l'application-cadre par le biais de l'InstanceDataChanged. La Frame Application doit distribuer les informations à tous les composants pertinents.

6.5.2.3.4 Données de configuration

Les données de configuration changeront chaque fois que l'utilisateur ajoute/retire des modules ou change le type de module, etc.

Dans le cas d'un DTM modulaire, le BIM sera informé lorsque l'utilisateur ajoute ou retire des modules au moyen du service ChildAdded et du service ChildRemoved. Les changements des paramètres dans un module seront rapportés par le service OnChildInstanceDataChanged.

Le DTM monolithique ou le DTM de BIM met à jour sa BMCP, demande une sauvegarde et informe la Frame Application par le biais de l'InstanceDataChanged.

NOTE 1 Les changements qui affectent la BMCP prennent souvent effet sur la topologie interne et les voies de processus (Process Channel). Il faut que ces informations soient mises à jour par les DTM aussi.

NOTE 2 La BMCP peut être modifiée par le DTM d'esclave et par le DTM de communication. Il ne faut pas que le DTM de communication change les données de configuration et les paramètres utilisateur.

6.5.2.4 Cas particuliers relatifs à la BMCP

6.5.2.4.1 Prise en charge de DPV1

Dans le fichier GSD, il y a deux fanions concernant DPV1. En premier lieu, la valeur "DPV1_Slave": Cela signifie que l'esclave a la possibilité de fonctionner comme esclave

DPV1. Si la valeur existe et la valeur est "1", le bit respectif dans le SI_Flag du Slave Bus Parameter Set doit être mis. Pour des systèmes plus anciens, il convient que la possibilité de fonctionner comme esclave DPV0 existe.

Seul le DTM de communication sait que son dispositif maître est capable de fournir des services acycliques.

Après avoir bâti le Slave Bus Parameter Set, le DTM de communication recevra la BMCP initiale de l'esclave. Si, dans le SI_Flag, le bit respectif est mis, le DTM de maître doit mettre le bit de poids fort dans le premier octet de Extended DPV1 Status. Maintenant, l'esclave fonctionne comme esclave DPV1.

6.5.2.4.2 Extended DPV1 status

Tous les bits dans les trois octets Extended DPV1 Status sont mis par le DTM de communication pour le maître. Un DTM esclave doit accepter ces valeurs de réglage et adapter sa fonctionnalité si besoin est.

7 Usage spécifique à un protocole des types de données généraux

La table ci-après (Tableau 9) explique comment des types de données, définis dans l'IEC 62453–2 au sein de l'espace de noms 'fdt', sont utilisés avec les protocoles CP 3/1 et CP 3/2.

Tableau 9 – Usage spécifique à un protocole des types de données généraux

Type de données	Description à utiliser dans l'IEC 61784 CPF 3
fdt:address	Adresse de paramètre Profibus: L'attribut 'address' suit les différents modèles de dispositifs qui sont définis pour les dispositifs PROFIBUS. FDT prend actuellement en charge les modèles suivants: PROFIBUS DP / DPV1, PROFIBUS PA, PROFIdrive (version supérieure ou égale à la version de profil 3) PROFIBUS DP / DPV1 Le modèle de dispositif est basé sur des dispositifs qui sont constitués de baies, attendu que les baies ne représentent pas des objets physiques. Les données qui sont contenues dans les baies sont adressables par de indices. Ces données peuvent être variables ou des blocs composés de données. L'attribut d'adresses est APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz sont basés sur le format décimal sans zéro ("0") de tête. PROFIBUS PA Le dispositif est représenté par une structure de gestion de dispositif et un nombre de blocs qui fournissent une fonctionnalité différente (bloc physique, bloc fonctionnel, bloc transducteur). Les blocs sont mis en correspondance avec des adresses de baies, mais cette mise en correspondance peut varier en fonction du type de dispositif. L'attribut d'adresses est APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz sont basés sur le format décimal sans zéro ("0") de tête. PROFIdrive Conformément au profil PROFIdrive, un dispositif (unité d'entraînement) peut être composé d'un certain nombre (1 à plusieurs) objets d'entraînement (DO

Type de données	Description à utiliser dans l'IEC 61784 CPF 3
	pour "Drive objects"). Les DO peuvent avoir des types différents. Chaque DO est identifiable de façon univoque et gère ses propres paramètres. Chaque paramètre peut être identifié de façon univoque par son numéro (PNU). Chaque DO a son propre espace de numéros. Un paramètre peut contenir des données simples ou des données composées (des matrices, par exemple). Les données du dispositif sont accessibles par une voie de paramètres (normalement baie 0 indice 47). L'attribut d'adresse est APIxxSLOTyyINDEXzz.Dodo-id.pnu xx API yy Slot zz Index do-id Drive Object ID pnu ParameterNumber xx, yy, zz, do-id, pnu sont basés sur le format décimal sans zéro ("0") de tête
fdt:protocolId	Voir Article 4
fdt:deviceTypeId	L'élément "fdt:deviceTypeId" doit contenir l'IDENT_NUMBER du dispositif physique pris en charge. L'IDENT_NUMBER doit être saisi au format décimal; toutefois, il convient que la valeur soit affichée sous forme hexadécimale pour l'utilisateur.
fdt:deviceTypeInformation	Un dispositif PROFIBUS doit fournir ses informations de GSD sous la forme d'une chaîne lisible par l'homme en cet attribut. NOTE Les informations de GSD sont accessibles au moyen des services NetworkManagementInfoRead GetTypeInformation
fdt:deviceTypeInformationPath	Chemin conduisant au fichier contenant les informations qui sont fournies par l'intermédiaire de l'attribut 'deviceTypeInformation'. Dans le cas de PROFIBUS, l'attribut contient le chemin complet du fichier GSD, y compris le nom du fichier. Le nom de fichier dépend du paramètre de lieu actuel conformément à l'utilisation du service SetLanguage. Pour les dispositifs PROFIBUS, il est obligatoire de fournir cet attribut. Exemples Anglais: 'C:\MyFolder\ABCD.GSE' Allemand: 'C:\MyFolder\ABCD.GSG'
fdt:manufacturerId	Fabricant selon la spécification de profils. Par exemple, dans Profibus PA: Physical Block Index 10: DEVICE_MAN_ID
fdt:semanticId fdt:applicationDomain	Les SemanticID pour PROFIBUS suivent les différents modèles de dispositifs qui sont définis pour les dispositifs PROFIBUS. FDT prend actuellement en charge les modèles suivants: PROFIBUS DP, PROFIBUS PA, PROFIdrive. PROFIBUS PA L' applicationDomain est: FDT_PROFIBUS_PA Le dispositif est représenté par une structure de gestion de dispositif et un nombre de blocs qui fournissent une fonctionnalité différente (bloc physique, bloc fonctionnel, bloc transducteur). Les blocs sont mis en correspondance avec des adresses de baies, mais cette mise en correspondance peut varier en fonction du type de dispositif. Le modèle de dispositifs étant basé sur des blocs, les SemanticIds sont également basés sur le modèle de blocs. Au sein de chaque bloc, les données sont identifiables par des noms de paramètres. Le semanticId pour le paramètre relatif au profil PROFIBUS respecte les règles suivantes: - le semanticId doit être construit sur la base des noms définis dans les profils; - les paramètres structurés doivent être combinés avec un '.'; - les espaces dans la définition du profil doivent être remplacés par un trait de soulignement ("underscore");

Type de données	Description à utiliser dans l'IEC 61784 CPF 3
	- les blocs doivent être comptés conformément au dictionnaire d'objets (Object Dictionary); - le numéro de bloc doit être partie intégrante du semanticID. Le semanticId est: BlockType.BlockIndex.NameOfParameter.AttributeOfParameter Exemple AnalogInputFB.3.OUT.Unit PROFIdrive L' applicationDomain est: FDT_PROFIBUS_PROFIDRIVE Conformément au profil PROFIdrive, un dispositif (unité d'entraînement) peut être composé d'un certain nombre (1 à plusieurs) objets d'entraînement (DO pour "Drive objects"). Les DO peuvent avoir des types différents. Chaque DO est identifiable de façon univoque et gère ses propres paramètres. Chaque paramètre peut être identifié de façon univoque par son numéro (PNU). Chaque DO a son propre espace de numéros. Un paramètre peut contenir des données simples ou des données composées (des matrices, par exemple). Les données du dispositif sont accessibles par une voie de paramètres (normalement baie 0 indice 47). Le semanticId est: DODO-id.PNUpnu do-id Drive Object ID pnu ParameterNumber do-id, pnu sont basés sur le format décimal sans zéro ("0") de tête. Exemple DO3.PNU64 PROFIBUS DPV1, L' applicationDomain est: FDT_PROFIBUS_DPV1 Le modèle de dispositif est basé sur des dispositifs qui sont constitués de baies, attendu que les baies ne représentent pas des objets physiques. Les données qui sont contenues dans les baies sont adressables par des indices. Ces données peuvent être variables ou des blocs composés de données. Le semanticID pour des dispositifs qui n'est pas basé sur un profil est directement basé sur les informations d'adresse PROFIBUS: Le semanticId est: APIxx.SLOTyy.INDEXzz xx API yy Slot zz Index xx, yy, zz sont basés sur le format décimal sans zéro ("0") de tête.
fdt:subDeviceType	Valeur spécifique à un fabricant

8 Types de données communs spécifiques à un protocole

Non applicable.

9 Types de données de gestion de réseau

9.1 Généralités

Les types de données spécifiés dans le présent paragraphe sont utilisés aux services suivants:

- service NetworkManagementInfoRead;
- service NetworkManagementInfoWrite.

Le type de données `net:DeviceAddress` (défini dans l'IEC 62453-2) est utilisé pour définir l'adresse réseau d'un dispositif.

Les types de données spécifiques à un protocole sont basés sur des définitions données dans les spécifications IEC 61784 et IEC 61158. En outre, celles-ci contiennent des informations complémentaires concernant le dispositif dont les systèmes ont besoin pour configurer les liaisons CPF 3 et établir la communication entre le dispositif maître CPF 3 et les dispositifs esclaves CPF 3.

9.1.1 Configuration

La configuration du dispositif en soi est effectuée avec l'aide de l'interface graphique utilisateur (GUI) du DTM. Le téléchargement de la configuration vers le dispositif esclave est accompli par l'intermédiaire du dispositif maître CPF 3. Pour ce faire et afin d'établir la communication de bus, le maître a besoin d'informations de la part du DTM comme elles existent:

- **Fichier GSD**
Les informations de GSD sont des informations spécifiques à un type et pas spécifiques à une instance. Elles ne sont pas stockées avec des instances d'esclaves individuelles ou dans un fichier accessible au niveau global. Elles sont fournies par le DTM au niveau du service `GetTypeInformation`. Sur ce service, un DTM d'un dispositif PROFIBUS fournit les informations de GSD au sein de son document XML.
Le dispositif maître peut utiliser les informations générales spécifiques à un type issues des informations de GSD de l'esclave comme les paramètres de temporisations de bus, les débits en bauds pris en charge, etc.
- **CFG-String (Cfg_Data)**
La CFG-String fournit des informations spécifiques à une instance relatives à la configuration courante du dispositif. Elle définit la structure des trames de données qui seront émises sur le PROFIBUS. Cette structure dépend des modules qui sont réellement configurés.
Le DTM fournit la CFG-String au sein de l'attribut `busMasterConfigurationPart` qui est partie intégrante du document XML disponible par le biais du service `InstanceDataRead`. La structure de l'attribut `busMasterConfigurationPart` est définie conformément au PROFIBUS-DP-Slave-Bus-Parameter-Set (voir l'IEC 62453-2 et aussi la série IEC 61158).

Le dispositif maître utilise ces informations pour établir la communication avec le dispositif esclave.

9.1.2 Process Channel (voie de processus)

Dans le cas des protocoles CPF 3, une FDT-Channel (voie FDT) est une représentation pour une date simple ou une valeur de processus qui peut être atteinte à partir de la Frame Application par le biais du dispositif maître. Les informations disponibles en des services pour des informations relatives aux entrées-sorties (E/S) décrivent comment accéder à une voie par une commande PROFIBUS DPV1 ou comment adresser une voie avec une trame PROFIBUS DP pour E/S cycliques. Outre tous les éléments obligatoires (qui incluent `id` et `dpAddress`), il est fortement recommandé que des informations d'adresse DPV1 soient fournies. Ces informations (DPV1 Slot) sont utilisées par certaines trames pour gérer les informations de modules de dispositifs PROFIBUS.

Dans un environnement DPV0, en fonction de la situation, le dispositif maître sous-jacent peut avoir soit la fonctionnalité de maître de classe 1, soit la fonctionnalité de maître de classe 2. Un maître de classe 1 peut écrire des données de sortie dans un dispositif et commander l'échange de données, là où un maître de classe 2 peut seulement lire les données de sortie. En général, il est présumé que la paramétrisation telle que décrite ici est accomplie en tant que station de maître de classe 2.

9.1.3 Paramétrisation

Il existe deux options pour écrire des paramètres établis à partir de la GUI du DTM dans le dispositif esclave CPF 3 sur le terrain:

- **Paramètres utilisateur**
Les User Parameters (paramètres utilisateur) sont partie intégrante du PROFIBUS-DP-Slave-Bus-Parameter-Set. Ils contiennent certaines données spécifiques à un fabricant pour caractériser l'esclave DP. Le DTM écrit les paramètres utilisateur dans la busMasterConfigurationPart. Les paramètres utilisateur sont stockés avec le dispositif maître pendant la configuration du maître PROFIBUS et sont automatiquement envoyés à l'esclave au cours de l'établissement de la communication de bus. (Ceci est spécifique à PROFIBUS; pour les détails, voir la série IEC 61158.) Lors du changement des paramètres utilisateur pendant le temps d'exécution, le DTM doit utiliser une connexion DP V0 et les commandes DP V0 appropriées pour l'échange de paramètres tel que décrit dans les types de données.
- **Écriture de paramètres avec les services DP-V1 (services MSAC2)**

Le DTM peut utiliser des services de transport DP-V1 pour envoyer ses paramètres vers le dispositif esclave. Pour cela, il doit utiliser une connexion DP-V1 et les commandes de communication correspondantes. Au cours de l'établissement de la communication, les services DP-V1 ne sont pas envoyés automatiquement. La Frame Application ou un DTM doit invoquer le téléchargement de paramètres par l'intermédiaire de DP-V1.

Pour les détails sur le comportement différent des esclaves en fonction du type de paramétrisation, se référer à l'IEC 61158.

Les connexions et les commandes de communication DP-V1 peuvent également être utilisées pour exécuter des commandes au niveau de l'esclave. Pour les détails sur l'utilisation de DP-V1, voir aussi la série IEC 61158.

9.2 Jeu de paramètres du bus maître

Le jeu de paramètres suivants représente le contenu du type de données busMasterConfigurationPart au sein des données d'instances de dispositifs pour le dispositif maître de Profibus (Tableau 10). Cet attribut doit être mis pour chaque dispositif maître de Profibus conforme à la série IEC 61158. Pour plus de détails, se référer au diagramme de séquence "Configuration d'un maître Fieldbus" et à la série IEC 61158.

Tableau 10 – Jeu de paramètres de bus pour dispositif maître

Nom	Type	Commentaire
Bus_Para_Len	UINT	Longueur du jeu de paramètres de bus inclusive, le paramètre de longueur (page 34 à 216-1)
FDL_Add	USINT	Partie obligatoire du jeu de paramètres de bus selon la Spécification DP PROFIBUS; non utilisée dans le présent contexte, car l'adresse est transférée dans une variable distincte; FDL-Add peut différer de l'adresse de station réelle utilisée par EE et les DTM
Baud_rate	USINT	Numéro de code pour le débit en bauds
TSL	UINT	Durée d'intervalle de temps
min TSDR	UINT	Réponse de retard de station minimal
max TSDR	UINT	Réponse de retard de station maximal
TQUI	USINT	Temps de cessation
TSET	USINT	Temps d'établissement
TTR	UDINT	Temps de rotation cible
G	USINT	Facteur mise à jour GAP
HSA	USINT	Plus haute adresse de station

Nom	Type	Commentaire
max_retry_limit	USINT	Limite maximale des répétitions de tentative
Bp_Flag	USINT	Fanions pour l'interface utilisateur, par exemple fanion d'action erronée
Min_Slave_Interval	UINT	Plus petite durée permise entre deux cycles de sondage d'esclaves
Poll_Timeout	UINT	Temporisation maître-maître
Data_Control_Time	UINT	Durée garantie entre deux mises à jour de Data_transfer_list
Octet 1 (réservé)	ARRAY[6] of USINT	
...		
Octet 6 (réservé)		
Master_User_Data_Len	UINT	Longueur de Master_User_Data, y compris le paramètre de longueur
Master_Class2_Name	ARRAY (32) of CHAR	Nom du maître de classe 2 DP avec lequel le jeu de paramètre a été créé
Master_User_Data	ARRAY OF USINT	Spécifique à un fabricant

9.3 Jeu de paramètres du bus esclave

Le jeu de paramètres suivants représente le contenu du type de données busMasterConfigurationPart au sein du jeu de données d'instances pour les dispositifs esclaves de Profibus (Tableau 11). Cet attribut doit être mis pour chaque dispositif esclave de Profibus conforme à la série IEC 61158. Pour plus de détails, se référer au diagramme de séquence "Configuration d'un maître Fieldbus" et à la série IEC 61158.

Tableau 11 – Jeu de paramètres de bus pour dispositif esclave

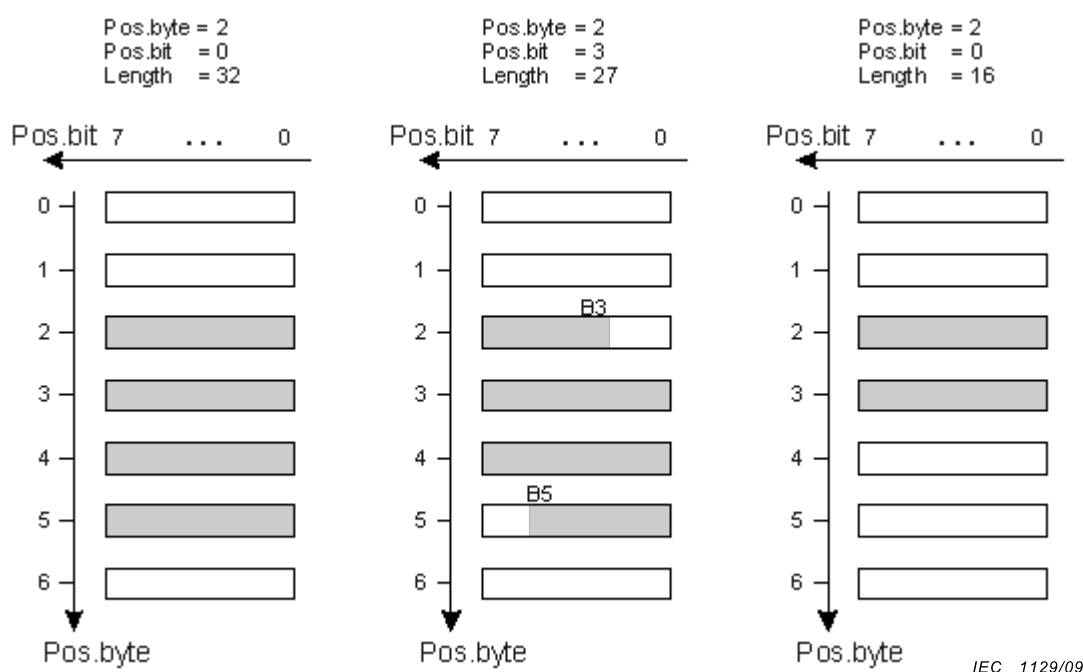
Nom	Type	Commentaire
Slave_Para_Len	UINT	Longueur du jeu de paramètres d'esclave, y compris le paramètre de longueur
SI_Flag	USINT	Fanions spécifiques à un esclave tels que New_Prm, Active, Fail_Safe,
Slave_Type	USINT	Désignation de type d'esclave spécifique à un fabricant (0 par défaut pour les esclaves DP)
Octet 1 (réservé)	ARRAY[12] of USINT	
...		
Octet 12 (réservé)		
Prm_Data_Len	UINT	Longueur de Prm_Data, y compris le paramètre de longueur (page 9 à 246)
Prm_Data	ARRAY OF USINT	
Cfg_Data_Len	UINT	Longueur de Cfg_Data, y compris le paramètre de longueur (page 3 à 246)
Cfg_Data	ARRAY OF USINT	
Add_Tab_Len	UINT	Longueur de Add_Tab, y compris le paramètre de longueur (page 2 à 216-31)
Add_Tab	ARRAY OF USINT	Table d'affectation d'adresses
Slave_User_Data_Len	UINT	Longueur de Slave_User_Data, y compris le paramètre de longueur (page 2 à 216-31)
Slave_User_Data	ARRAY OF USINT	Données spécifiques à un fabricant pour caractériser l'esclave DP pour le maître.

9.4 Données de module et de voie

Une configuration de module courante d'un esclave, y compris les voies connexes, doit être mise à disposition par les objets FdtChannel. Cela est indispensable, car au sein de l'environnement, les variables de processus doivent être affectées à des voies seules et la configuration d'esclave doit être affichée dans la vue d'ensemble du système de l'environnement sans utiliser l'interface utilisateur du DTM.

L'adressage de voies au sein de la trame de données PROFIBUS est orienté bit et indépendant du type de données. Une position de bits et une longueur en bits déterminent chaque voie. Le type de données d'un paramètre détermine comment convertir le champ de bits.

- Voies au sein des trames de données PROFIBUS (Figure 3):



Légende

Anglais	Français
Pos. byte	Octet Pos.
Pos Bit	Bit Pos.
Lenght	Longueur

Figure 3 – Exemple pour données E/S dans des datagrammes

- Représentation dans la notation FDT:

bitPosition="16"

bitPosition="19"

bitPosition="16"

bitLength="32"

bitLength="27"

bitLength="16"

L'exemple suivant montre la description de la structure attendue d'un esclave modulaire dans un jeu de paramètres DTM utilisant des paramètres FDT avec des types d'objets définis dans la Spécification FDT.

- Représentation de la trame de données utilisant des informations de GSD:

```
BeginSlave;;
```

```
RIOLB8101.GSD;6;
```

BeginModules;;

1X03 FrequencyCount;0x51;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

2XXX ValveBlock.;0x30;

EndModules;;

EndSlave;;

Description de la position de voies de signal au sein de la trame de données ():

Tableau 12 – Voies de signal au sein de la trame de données

# Frame description Remote I/O (Description de trames - E/S distantes)											
Vendeur:	"RIO manufacturer"										
Devicetype:	"LB 8101"		Device ident:			0x8101	# hex				
Frameident:	"B1t1"										
Données d'entrée											
Voie	Type de voie	Données	Données	Données	Bit non valide	Bit non valide	Bit de valeur de substitution	Bit de valeur de substitution	Bit de simulation	Bit de simulation	État-voie
		Position Octet.	Position Bit.	Longueur	Position Octet	Position Bit	Position Octet	Position Bit	Position Octet	Position bit	
Module:	"EP01"	"Digital Input (Counter) LB/FB 1X03 (4 Byte)"									
"Count_1_0"	READ INT	0	0	32	-/-	-/-	-/-	-/-	-/-	-/-	0
Module:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input and Output)"									
"DI_8_LFOUT0"	READ BOOL	4	1	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF1"	READ BOOL	4	3	1	-/-	-/-	-/-	-/-	-/-	-/-	1
"DI_8_LF2"	READ BOOL	4	5	1	-/-	-/-	-/-	-/-	-/-	-/-	1
Données de sortie											
Voie	Type de voie	Données	Données	Données	Bit non valide	Bit non valide	Bit de valeur de substitution	Bit de valeur de substitution	Bit de simulation	Bit de simulation.	État-voie
		Position Octet.	Position Bit.	Longueur	Position Octet.	Position Bit.	Position Octet.	Position Bit.			
Modul:	"EP08"	"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input u. Output)"									
"DO_8_0"	WRITE BOOL	0	0	1	4	1	-/-	-/-	-/-	-/-	0

9.5 Informations de GSD

9.5.1 Généralités

Les informations de GSD ne sont pas stockées avec des instances d'esclaves individuelles ou dans un fichier accessible au niveau global. Elles sont fournies par le DTM au niveau du service GetTypeInformation. Un DTM d'un dispositif PROFIBUS fournit les informations de GSD au sein de l'argument fdt:DtmDeviceType.

Le DCS existant utilise les informations de GSD pour obtenir des informations relatives à la configuration et aux paramètres possibles d'un esclave DP.

Outre les informations relatives aux modules et à ses paramètres, un fichier GSD contient des informations complémentaires concernant l'esclave, telles que les débits en bauds pris en charge.

Ces informations sont utiles pour un système DCS pour configurer un réseau tout entier selon les capacités des différents esclaves.

9.5.2 GSD pour dispositifs passerelles

9.5.2.1 Types de dispositifs passerelles Profibus

Il existe deux types de dispositifs passerelles.

Les dispositifs passerelles visibles fonctionnent comme un esclave PROFIBUS (avec une adresse PROFIBUS) et pour le réseau sous-jacent, ils agissent comme un maître.

Les dispositifs passerelles invisibles transforment juste le réseau PROFIBUS en le réseau sous-jacent.

Pour les deux types de dispositifs, il existe un besoin de GSD particulières.

9.5.2.2 Dispositifs passerelles visibles

Ces dispositifs passerelles visibles sont livrés sans un fichier GSD. À la place, ils ont une suite logicielle propriétaire qui le configure et le paramètre ou bien ils sont livrés avec un outil qui crée une GSD pour logiciel de paramétrisation. L'outil de GSD crée une GSD pour esclave en fonction de la configuration du réseau sous-jacent ou en fonction des valeurs de réglage des bus (les débits en bauds, par exemple).

Dans ces cas, il convient que le DTM fournisse la fonctionnalité de l'outil de GSD. Si la GSD est dépendante de la configuration, il pourrait appeler le service NetworkManagementInfoRead pour chacun de ses enfants, extraire les informations de GSD et créer des informations de GSD dépendantes de la configuration de la même manière que le fait l'outil. Après initialisation du DTM, il convient qu'il délivre une BMCP en fonction du dispositif de liaison lui-même. Chaque fois qu'un enfant est ajouté ou retiré, cela conduit à un changement dans son propre document de paramètres.

Si le GSD dépend des valeurs de réglage du bus, un dialogue de configuration ou de paramétrisation de DTM pourrait être utilisé pour modifier les valeurs de réglage du bus. Sur la base de ces valeurs de réglage, des informations de GSD mises à jour peuvent être insérées dans le document d'informations. Ici aussi, le DTM doit demander une sauvegarde et appeler le service OnInstanceDataChanged.

Noter que la structure interne avec ses modules et ses voies doit être mise à jour également.

9.5.2.3 Dispositifs passerelles transparents

Il existe des dispositifs de liaison transparents sur le marché (PROFIBUS FMS/DP et PROFIBUS PA) qui accomplissent une transformation de débits en bauds. Cela exige un traitement particulier des fichiers GSD spécifiques à un esclave. Il existe des outils disponibles qui sont capables d'adapter des fichiers GSD existants selon le débit en bauds plus élevé (lesdits "Convertisseurs de GSD").

Les informations de GSD doivent être délivrées par le DTM pour le dispositif. Afin de prendre en charge cette sorte de dispositifs de liaison, un DTM d'esclave doit exposer le fichier GSD sur le disque dur en utilisant l'attribut 'deviceTypeInformationPath' (se référer à DtmDeviceType défini dans l'IEC 62453-2).

Il convient qu'un développeur de DTM garde à l'esprit que fichier GSD référencé pourrait être converti afin de prendre en charge l'architecture de système spécifique.

Un DTM Profibus dans l'attribut 'deviceTypeInformation' est censé exposer exactement le fichier GSD qui est référencé par l'attribut 'deviceTypeInformationPath'.

10 Types de données de communication

10.1 Généralités

Les types de données définis ici sont utilisés au:

- service Connect (connexion);
- au service Disconnect (déconnexion);
- au service Transaction (transaction).

10.2 Informations d'erreur fournies par la voie de communication Communication Channel

Dans chaque type de données Response de la Spécification FDT Profibus, il est fourni un élément 'errorCode'. Selon PROFIBUS, le code d'erreur est normalisé pour être constitué de trois octets, chaque octet contenant une signification.

Sachant que le code d'erreur est échangé entre des DTM différents (par exemple: DTM de communication et DTM de dispositif) et que le destinataire du code d'erreur tentera de comprendre le code d'erreur, le fournisseur doit utiliser le format normalisé:

- longueur normalisée de trois octets;
- si le dispositif fournit des codes d'erreur, ces codes d'erreur sont fournis (et non des codes d'erreur locaux issus du maître).

Si aucune erreur ne s'est produite, l'élément 'errorCode' doit être rempli de zéros ("000000").

10.3 Communication DPV0

Les services pris en charge dépendent du type de fonctionnalité de maître PROFIBUS qui est fournie par l'infrastructure de communication. Les dispositifs maîtres de classe 1 commandent typiquement les esclaves et fournissent une communication cyclique, les dispositifs maîtres de classe 2 sont typiquement utilisés pour configurer les esclaves et fournir une communication acyclique.

Tous les services définis ne sont pas pris en charge si le maître n'est pas engagé dans un échange cyclique de données avec les esclaves. Dans de tels cas, le comportement suivant est attendu:

Si une voie de communication reçoit une demande qui ne peut pas être prise en charge, le service Transaction répond par result="false".

Les types de données décrits dans le présent article sont définis pour l'espace de noms suivant:

Espace de noms: dpv0

Tableau 13 – Types de données simples de communication DPV0

Type de données	Définition	Description
busAddress	USINT	Informations d'adresse selon la série IEC 61158 (voir aussi l'IEC 62453-2, attribut busAddress)
communicationReference	UUID	Identificateur obligatoire pour une liaison de communication à un dispositif. Cet identificateur est alloué par le composant de communication pendant la création de la connexion. Les informations d'adresse doivent être utilisées pour tous les appels de communication suivants
connectStatus	enumeration (masterConnectedOnly deviceAtLifeList deviceInDataExchange)	Décrit le statut de connexion établi par le composant de communication. Le statut "masterConnectedOnly" signifie que le composant de communication a établi une connexion au dispositif maître Profibus et acceptera un accès en ligne aux paramètres utilisateur, indépendamment du fait que le dispositif soit disponible ou non. Le statut "deviceAtLifeList" signifie que le composant de communication a établi une connexion au dispositif maître Profibus et a vérifié que le dispositif est dans la liste de vie de la pile du maître. Dans cet état, le maître acceptera un accès en ligne aux paramètres utilisateur et enverra le paramètre utilisateur au dispositif, indépendamment du fait que le dispositif soit engagé dans un échange de données ou non. Le statut "deviceInDataExchange" signifie que le composant de communication a établi une connexion au dispositif maître Profibus et a vérifié que le dispositif est engagé dans un échange de données. Dans cet état, le maître acceptera un accès en ligne aux paramètres utilisateur et enverra le paramètre utilisateur au dispositif, afin que les nouvelles données influencent directement le processus.
delayTime	UDINT	Retard en [ms] entre deux appels de communication
errorCode	ARRAY OF USINT	Informations de statut selon la série IEC 61158. Pour la description de code d'erreur, voir: la DIN 19245 Partie 3, PROFIBUS (P. 40ff., 83ff., 39)
schemaVersion	INT	Définit la version du schéma
sequenceTime	UDINT	Durée en [ms] pour toute la séquence

Tableau 14 – Types structurés de données de communication DPV0

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
Abort	STRUCT			Décrit l'abandon	-
	communicationReference	O	[0..1]		
ConnectRequest	STRUCT			Décrit la demande de communication pour établir une connexion à un dispositif maître Profibus. Selon l'infrastructure de réseau utilisée, il est possible que ce service ne soit pas mis en correspondance avec une demande de bus de terrain, mais soit utilisé pour gérer le logiciel. Il est également possible que ce service soit utilisé comme déclencheur pour établir l'état d'un dispositif.	-
	busAddress	M	[1..1]		
	fdt:systemTag	O	[0..1]		
ConnectResponse	STRUCT			Décrit la réponse de communication à la demande de connexion et donne les informations sur la manière dont les commandes WriteUserParameter suivantes seront envoyées vers le dispositif (voir connectStatus). La connexion résultante dépend du dispositif de communication utilisé et de la configuration du maître bus. Le DeviceDTM doit s'attendre à ce que la connexion résultante ne puisse pas être utilisée pour accéder à tous les services. ConnectResponse peut retourner "true" même s'il n'y a aucune communication possible avec le dispositif. Par exemple, si ConnectRequest retourne "true" avec connectionStatus = "masterConnectedOnly", cela signifie qu'il y a une connexion au dispositif maître, mais le dispositif n'est pas sur la liste de vie et ne peut donc pas être accessible. En fonction du connectStatus, certains services peuvent ne pas être disponibles (par exemple: dans le état "masterConnectedOnly", seuls des services fournis par le maître sont accessibles (ReadUserParameter, WriteUserParameter, ...) attendu que les services fournis par le dispositif ne sont pas accessibles. NOTE Dans cet exemple, le DTM est dans un état "online" (en ligne), mais il ne peut pas accéder au dispositif. Les services fournis dépendent également du type du dispositif maître. Pour une vue d'ensemble de la disponibilité des services en fonction du type de connexion, voir les tableaux suivants.	-

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
	connectStatus	M	[1..1]		
Disconnect ¬Request	STRUCT			Décrit la demande de communication pour libérer une connexion à un dispositif maître Profibus.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
Disconnect ¬Response	STRUCT			Décrit la réponse de communication.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
ReadDiagnosis ¬Data ¬Request	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Pour la Cl.1 de maître, se référer à l'IEC 61158-5, 8.2.3.3.2 "Get slave diag". Pour la Cl.2 de maître, se référer à l'IEC 61158-5, 8.2.3.3.3 "Read slave diag"	1, 2
	communicationReference	M	[1..1]		
ReadDiagnosis ¬Data Response	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
ReadInputData→Request	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Selon que la voie FDTChannel sous-jacente est de la Cl. 1 de maître ou de la Cl. 2 de maître, ce service sera local ou conduira à un accès en lecture au dispositif esclave. Il est possible de spécifier quelles données sont lues en fournissant un fdt:ChannelReference. Pour la Cl.1 de maître, se référer à l'IEC 61158-5, 8.2.2.3.3 "Get Input". Pour la Cl.2 de maître, se référer à l'IEC 61158-5, 8.2.2.3.2 "Read Input"	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadInputData→Response	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadOutputData→Request	STRUCT			Décrit la demande de communication selon la Spécification Profibus DPV0. Selon que la FDTChannel sous-jacente est de la Cl. 1 de maître ou de la Cl. 2 de maître, ce service sera local ou conduira à un accès en lecture au dispositif esclave. Il est possible de spécifier quelles données seront lues en fournissant un fdt:ChannelReference. Pour la Cl. 1 de maître, il s'agit d'un service propriétaire (ou non disponible). Pour la Cl. 2 de maître, se référer à [1], section 8.2.2.3.6. "Read Output" (un service facultatif)	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadOutputData→Response	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
ReadUserParameterRequest	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. La demande récupère uniquement les informations qui sont disponibles auprès du dispositif maître (service local). Elles peuvent différer des données réelles du dispositif. La mise en œuvre de la manière de fournir les données est propriétaire. Elle est basée sur un service local ou n'est pas disponible.	1
	communicationReference	M	[1..1]		
ReadUserParameterResponse	STRUCT			Décrit la réponse de communication, lue dans le dispositif maître Profibus, conformément à la spécification Profibus DPV0. Les données retournées peuvent ne refléter que les informations qui sont mises à disposition par le dispositif maître. Elles peuvent différer des données réelles du dispositif	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
SequenceBegin	STRUCT			Décrit la séquence begin	-
	sequenceTime	O	[0..1]		
	delayTime	O	[0..1]		
	communicationReference	M	[1..1]		
SequenceEnd	STRUCT			Décrit la séquence end	-
	communicationReference	M	[1..1]		
SequenceStart	STRUCT			Décrit la séquence start	-
	communicationReference	M	[1..1]		

Type de données	Définition			Description	Classe de maître
	Type de données élémentaire	Usage	Multiplicité		
WriteOutputDataRequest	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Les données de sortie seront envoyées selon la connexion établie. Il est nécessaire de spécifier quelles données seront écrites par fourniture d'un fdt:ChannelReference. Si la FDTChannel sous-jacente est fournie comme Cl. 2 de maître, l'écriture de données de sortie (Output Data) ne sera pas possible. Se référer à l'IEC 61158-5, 8.2.2.3.5 "Set Output". Cette fonctionnalité dépend du type de maître utilisé. En termes de PLC (automate programmable), cette fonctionnalité est appelé "forcing" de valeurs.	1
	communicationReference	M	[1..1]		
	fdt:ChannelReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteOutputDataResponse	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
WriteUserParameterRequest	STRUCT			Décrit la demande de communication conformément à la Spécification Profibus DPV0. Le paramètre utilisateur sera envoyé selon la connexion établie (voir connectStatus). Pour la Cl. 1 de maître, cela est basé sur le service "DDL_M_Set_Prm". Pour la Cl. 2 de maître, cela est basé sur le service "DDL_M_Set_Prm" (un service facultatif).	1, 2
	communicationReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteUserParameterResponse	STRUCT			Décrit la réponse de communication conformément à la Spécification Profibus DPV0.	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Selon le type de maître bus et le connectStatus retourné, les services suivants sont disponibles.

Tableau 15 – Disponibilité des services pour la classe 1 de maître (C1)

Demande de services de DTM d'esclave	connectStatus		
	masterConnectedOnly	DeviceAtLifeList	DeviceInDataExchange
Connect	✓	✓	✓
ReadUserParameter	O	O	O
WriteUserParameter	✓	✓	✓
ReadOutputData			✓
WriteOutputData			O
ReadInputData			✓
ReadDiagnosisData		✓	✓
NOTE ✓: le service est disponible, O: le service est facultatif et peut être disponible, selon les capacités du dispositif maître sous-jacent.			

Pour la Class2 de maître (C2), tous les états "connect" (connexion) ne sont pas nécessairement disponibles:

Tableau 16 – Disponibilité des services pour la classe 2 de maître (C2)

Action de DTM d'esclave	connectStatus	
	DeviceAtLifeList (aucune connexion DPV1 au dispositif)	DeviceInDataExchange (connexion DPV1 au dispositif)
Connect	✓	✓
ReadUserParameter		
WriteUserParameter	O	O
ReadOutputData		O
WriteOutputData		
ReadInputData		✓
ReadDiagnosisData	✓	✓
NOTE ✓: le service est disponible, O: le service est facultatif et peut être disponible, selon les capacités du dispositif maître sous-jacent.		

Si le composant de communication de la classe 2 de maître prend en charge DPV1 et DPV0 et a établi une connexion DPV1 au dispositif, un appel au service Connect pour DPV0 doit retourner le statut "DeviceInDataExchange" même si le dispositif n'est pas dans le statut DataExchange.

Si aucune connexion DPV1 n'est établie, le composant de communication de la classe 2 de maître doit vérifier la disponibilité du dispositif (au moins par le service LifeList).

10.4 Communication DPV1

Les types de données sont définis pour prouver les informations d'adresse et les données de communication requises pour la communication DPV1 (voir Tableau 17 et Tableau 18).

Les types de données décrits dans le présent paragraphe sont définis pour l'espace de noms suivant:

Espace de noms: dpv1

Tableau 17 – Types de données simples de communication DPV1

Type de données	Définition	Description
api	USINT	Informations d'adresse selon la série IEC 61158. Si le dispositif a besoin de valeurs particulières pour DPV1-Initiate, le DeviceDTM a la responsabilité de fournir les valeurs dans la ConnectRequest. Si les valeurs ne sont pas fournies, le composant de communication utiliser des valeurs par défaut (0)
busAddress	USINT	Informations d'adresse selon la série IEC 61158 (voir aussi DTMPParameterSchema, attribut busAddress)
errorCode	UUID	Informations de statut selon la série IEC 61158. Pour la description de code d'erreur, voir: PROFIBUS Nutzerorganisation e.V., PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998. (P.. 120-121); .Pour la description des informations d'abandon, voir: la DIN 19245 Partie 3, PROFIBUS, (P..54ff., 11)
index	UDINT	Informations d'adresse selon la série IEC 61158.
communicationReference	ARRAY OF USINT	Identificateur obligatoire pour une liaison de communication à un dispositif. Cet identificateur est alloué par le composant de communication pendant la connexion. Les informations d'adresse doivent être utilisées pour tous les appels de communication suivants
delayTime	USINT	Retard minimal en [ms] entre deux appels de communication
maxLenDataUnit	USINT	Attribut facultatif, pour décrire la quantité des données qui peuvent être transférées par la connexion établie. Si cet attribut n'est pas valide, aucune restriction de longueur particulière n'est annoncée. Chaque composant de communication au sein de la chaîne d'interfaces concernant la communication imbriquée pourrait introduire cet attribut. Il convient que chaque composant de communication change le contenu de l'attribut en se basant sur la règle suivante: La nouvelle valeur est le minimum de la valeur courante et de la restriction de sa propre mise en œuvre. Si un composant de communication n'a pas de restriction, il convient qu'il transfère la valeur donnée. Si un composant de communication est capable de réutiliser une connexion établie concernant une nouvelle demande de connexion, il convient qu'il prenne en compte la longueur des données déterminée pour la connexion existante. Si la longueur de données n'est pas applicable, il convient que le DTM envoie un message d'erreur au moyen du service ErrorMessage
networkMACAddress	ARRAY OF USINT	Adresse réseau ou adresse MAC telle que décrite dans la série IEC 61158. [5] S'il y a une networkMACAddress, (à savoir networkMACAddress <> ""), le Profibus-DPV1-Master-DTM doit mettre à "1" le "address type" (type d'adresse) correspondant dans le télégramme de demande Initiate, autrement à "0". Exemple Dispositif avec l'adresse 20 (14hex) adressé par l'intermédiaire d'un dispositif de liaison: networkMACAddress="00000000000014"
scl	INT	Le niveau d'accès tel que décrit dans la série IEC 61158. [5] Si le dispositif a besoin de valeurs particulières pour DPV1-Initiate, le DeviceDTM a la responsabilité de fournir les valeurs dans la ConnectRequest. Si les valeurs ne sont pas fournies, le composant de communication utiliser des valeurs par défaut (0)
sequenceTime	USINT	Durée en [ms] pour toute la séquence
slot	UDINT	Informations d'adresse selon la série IEC 61158.

Tableau 18 – Types de données structurés de communication DPV1

Type de données	Définition			Description
	Types de données élémentaires	Usage	Multi-plicité	
Abort	STRUCT			Décrit l'abandon
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Décrit la demande de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	fdt.systemTag	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
ConnectResponse	STRUCT			Décrit la réponse de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	maxLenDataUnit	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
DestNetworkAddress	STRUCT			Décrit l'adresse étendue de la destination [5].
	NetworkAddress	M	[1..1]	
DisconnectRequest	STRUCT			Décrit la demande de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Décrit la réponse de communication.
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
NetworkAddress	STRUCT			Décrit le format d'adresse étendue.
	api	M	[1..1]	
	scl	M	[1..1]	
	networkMACAddress	M	[1..1]	
ReadRequest	STRUCT			Décrit la demande de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multi- plicité	
ReadResponse	STRUCT			Décrit la réponse de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
RedundantAddresses	STRUCT			Décrit les adresses redondantes.
	fdtparam:SlaveAddress	M	[1..*]	
SequenceBegin	STRUCT			Décrit la séquence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	
SequenceEnd	STRUCT			Décrit la séquence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Décrit la séquence start
	communicationReference	M	[1..1]	
SrcNetworkAddress	STRUCT			Décrit l'adresse étendue de la source.
	NetworkAddress	M	[1..1]	
WriteRequest	STRUCT			Décrit la demande de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
WriteResponse	STRUCT			Décrit la réponse de communication.
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	

Au sein d'une demande de connexion, un DTM d'un esclave redondant Profibus peut fournir des adresses complémentaires d'esclaves redondants, établies au sein de son document de paramètres. L'attribut busAddress doit être utilisé comme l'adresse préférentielle; il convient que les adresses dans l'élément redundantAddresses soient utilisées dans cet ordre si une adresse de remplacement est utilisée pour connecter l'esclave redondant.

11 Types de données paramètres de voie (Channel)

Le tableau suivant (Tableau 19) décrit la mise en correspondance des types de données DPV1 avec les types de données FDT.

**Tableau 19 – Mise en correspondance des types de données DPV1
 aux types de données FDT**

Types de données PROFIBUS	Type de données FDT
Boolean	BOOL
Integer8	SINT
Integer16	INT
Integer32	DINT
Unsigned8	USINT
Unsigned16	UINT
Unsigned32	UDINT
Floating Point ("Virgule flottante")	FLOAT
Visible String	STRING
Octet String	ARRAY OF USINT
Date	DATE
Time of Day	TIME
Time Difference	duration
DS-32	structured (structuré)
DS-33	structured (structuré)
DS-34	structured (structuré)
DS-35	structured (structuré)
DS-36	structured (structuré)
...	structured (structuré)
DS-52	structured (structuré)
..	structured (structuré)

Le format de tous les types de données est défini dans la série IEC 61158.

Si seule une partie des données récupérées est constituée de données de processus (par exemple: pour DS-33, les données sont constituées de la valeur float et du statut d'octet), les informations d'accès (DPAddress ou DPV1Address) contiennent également des informations qui sont nécessaires pour récupérer les informations issues de la structure de données.

Utilisés au service ReadChannelInformation et au service WriteChannelInformation.

Les types de données paramètres de voies décrivent le mode d'accès à une voie par une commande Profibus DPV1 ou le mode d'adressage d'une voie au sein d'un cadre Profibus DP pour une E/S cyclique. Les tableaux suivants (Tableau 20 et Tableau 21) fournissent une description des types de données.

Les types de données décrits dans le présent article sont définis pour l'espace de noms suivant:

Espace de noms: profichannel

Tableau 20 – Types simples de données pour les paramètres de voies

Type de données	Définition	Description
api	USINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1
bitLength	UDINT	Informations complémentaires sur les types de données, notamment pour les types de données spécifiques aux bus de terrain comme entier de 12 bits
bitPosition	UDINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DP
frameApplicationTag	STRING	Étiquette spécifique à l'application-cadre utilisée pour l'identification et la navigation. Il convient que le DTM affiche cette étiquette aux interfaces utilisateur spécifiques à une voie.
gatewayBusCategory	UUID	Identificateur unique pour un type de bus pris en charge comme Profibus ou HART conformément au CATID spécifique à FDT
index	USINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1
invalidBit	UDINT	Position de bit de la voie de statut invalide accessible par le biais de Profibus DP
logic	UDINT	Informations de types de données supplémentaires: positive 0=FALSE 1=TRUE
number	BOOL	Informations d'adresse pour diagnostic selon la série IEC 61158 pour les voies accessibles par Profibus DP Voir [5], 9.3.1 Read DP – Slave Diagnostic Information
protectedByChannelAssignment	INT	TRUE si la voie est mise à lecture seule par la Frame Application (application-cadre). Habituellement mis à TRUE s'il existe une attribution de voie.
schemaVersion	UDINT	Définit la version du schéma.
simulationBit	USINT	Position de bit de la voie de statut de simulation accessible par le biais de Profibus DP
slotNumber	BOOL	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1
statusChannel	UDINT	TRUE si la voie est pour informations de statut seulement.
substituteValueBit	STRING	Position de bit de la voie de statut de substitution accessible par le biais de Profibus DP

Tableau 21 – Types structurés de données pour les paramètres de voies

Type de données	Définition			Description
	Type de données élémentaire	U s a g e	Multi- PLICITÉ	
DpAddress	STRUCT			Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DP
	bitPosition	M	[1..1]	
	bitLength	M	[1..1]	
DpV1Address	STRUCT			Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1. Il est fortement recommandé de fournir des informations d'adresse pour permettre l'affectation d'adresses logiques à un module profibus.
	api	M	[1..1]	
	slotNumber	M	[1..1]	
	index	M	[1..1]	
	bitPosition	O	[0..1]	
	bitLength	O	[0..1]	
FDTChannel	STRUCT			Description de la voie
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannel ¹ Assignment	M	[1..1]	
	number	M	[1..1]	
	fdt:dataType	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	applId:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	DpAddress	O	[0..1]	
	DpV1Address	O	[0..1]	
	StatusInformation	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:StructuredElements	O	[0..1]	
FDTChannelType	STRUCT			Description du composant de voie dans le cas des voies avec fonctionnalité de passerelle
	fdt:VersionInformation	M	[1..1]	
	gatewayBusCategory	O	[0..1]	
	statusChannel	O	[0..1]	
StatusInformation	STRUCT			Description d'informations de statut complémentaires pour les voies accessibles par Profibus DP

Type de données	Définition			Description
	Type de données élémentaire	U s a g e	Multi- plicité	
	logic	M	[1..1]	
	invalidBit	O	[0..1]	
	simulationBit	O	[0..1]	

12 Identification de dispositif

12.1 Généralités

Il existe différents éléments d'identification spécifiques à Profibus.

Un Profibus Scan (balayage Profibus) peut détecter différents types de dispositifs: dispositifs I&M, dispositifs Profile ou purs dispositifs DP.

La règle suivante doit être appliquée pour des Profibus Communication Channels (voies de communication Profibus):

- Si I&M est disponible, créer l'identification I&M,
- autrement vérifier Profile PA,
- autrement créer information purement DP = IDENT_NUMBER.

Les types de données décrits dans le présent paragraphe sont définis pour l'espace de noms suivant:

Espace de noms: profiident

12.2 Traitement, spécifique à un protocole, du type de données STRING

Règles pour char array (matrices de caractères) Profibus:

- dans toutes les chaînes basées sur des plages de caractères (char) définies dans la spécification de protocole de bus de terrain, les espaces de têtes sont rognés. La matrice de caractères (char array) doit être remplie de caractères 0x20h (blanc);
- dans les VisibleString, les caractères invisibles fournis par un dispositif doivent être remplacés par '?'

12.3 Types de données d'identification de types de dispositif communs

Les types de données décrits dans cet article sont réutilisés comme défini par 12.4 et 12.6 .

Le Tableau 22 énumère les données, source et format pertinents pour l'identification pour Profibus DP.

Tableau 22 – Types de données d'identification avec mise en correspondance spécifique à Profibus DP

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Type de données FDT (format d'affichage)	Référence de la spécification
busProtocol	IdBusProtocol	Pour tous les dispositifs DP: "protocol_DP"	Protocol	-	Enum	-
slaveAddress	IdAddress	L'adresse de bus est fournie comme partie intégrante de la liste de vie par un maître PROFIBUS. Service: FMA1/2_LIVE_LIST	Slave Address	Unsigned8	STRING (décimal)	[5] Partie 2, Section 4.2.3.6
identNumber	IdTypeID	IDENT_NUMBER peut être demandé par: DP Service DDLM_SLAVE_DIAG Les valeurs permises sont: IDENT_NUMBER de profil: 0x9700 (0x9700 à 0x9742) ou IDENT_NUMBER spécifique à un fabricant	IDENT_NUMBER	Unsigned16	ARRAY of USINT (hexadécimal)	[5] Partie 8, Section 9.3.1
manufacturerSpecificExtension		Peut être utilisé par DTM pour des informations d'identification de dispositif spécifiques à un vendeur, par exemple en combinant un certain nombre de valeurs de paramètres de dispositif en une seule valeur string (chaîne). Cela peut être utilisé pour identifier une variante de dispositif spécifique.			STRING	

Le Tableau 23 montre la mise en correspondance pour les données I&M PROFIBUS.

Tableau 23 – Types de données d'identification avec mise en correspondance spécifique à Profibus I&M

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Format FDT (format d'affichage)	Référence de la spécification
identNumber, manufacturerSpecificExtension et slaveAddress tels que définis dans le tableau DP (Tableau 22).						
busProtocol	IdBusProtocol	Pour tous les dispositifs I&M: "protocol_IM"	Protocol	-	Enum	-
manufacturerId	IdManufacturer	I&M 0 élément 1 mis en correspondance avec PB.DEVICE_MAN_ID	MANUFACTURER_ID	Unsigned16	UINT	[6] Section 3.2.2
orderId	-	I&M 0 élément 2 mis en correspondance avec PB.DEVICE_ID	ORDER_ID	20 Octets Visible String	STRING[20]	[6] Section 3.2.3
serialNumber	IdSerialNumber	I&M 0 élément 3 numéro de série unique mis en correspondance avec PB.DEVICE_SER_NUM	SERIAL_NUMBER	16 Octets VisibleString	STRING[16]	[6] Section 3.2.4
hardwareRevision	IdHardwareRevision	I&M 0 élément 4 HARDWARE_REVISION NOTE: Si le dispositif est un dispositif PA, (PROFILE_ID est mis à 0x9700), HARDWARE_REVISION doit être lu dans PA_I&M0 [6]	HARDWARE_REVISION	2 Octets - entier non signé de 16 bits converti en string (chaîne) pour couvrir également: PA_I&M0: VisibleString 16	STRING	[6] Section 3.2.5, Amendement I&M
softwareRevision	IdSoftwareRevision	I&M 0 élément 5 SOFTWARE_REVISION NOTE: Si le dispositif est un dispositif PA (PROFILE_ID est mis à 0x9700), SOFTWARE_REVISION doit être lu dans PA_I&M0 Le format String est utilisé pour harmoniser la gestion des versions pour tous les protocoles de bus de terrain.	SOFTWARE_REVISION	4 Octets – 1 char + 3 unsigned 8 , par exemple: V1.2.3 PA_I&M0: VisibleString 16	STRING	[6] Section 3.2.6, Amendement I&M

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Format FDT (format d'affichage)	Référence de la spécification
profileID	-	I&M 0 élément 7	PROFILE_ID	2 Octets – unsigned 16	UINT	[6] Section 3.2.8
profileSpecificType	-	I&M 0 élément 8 MSB = BLOCK_OBJECT, LSB = PARENT_CLASS	PROFILE_SPECIFIC_TYPE	2 Octets – unsigned 16	UINT	[6] Section 3.2.9
imVersion	-	I&M 0 élément 9	IM_VERSION	2 Octets – unsigned 16; MSB version majeure xxx, LSB version mineure yyy	FLOAT xxx.yyy	[6] Section 3.2.10
imSupported	-	I&M 0 élément 10	IM_SUPPORTED	2 Octets -	UINT	[6] Section 3.2.11
tagFunction	IdTag	I&M 1 élément 1 TAG_FUNCTION	TAG_FUNCTION	32 Octets – visible string	STRING[32]	[6] Section 3.2.12
tagLocation	-	I&M 1 élément 2 TAG_LOCATION	TAG_LOCATION	22 Octets – visible string	STRING[22]	[6] Section 3.2.13
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM ProfileDTM, BlockSpecificProfileDTM	Niveau de prise en charge de DTM	-	Enum	

Le Tableau 24 énumère les données, source et format pertinents l'identification pour Profibus PA.

Tableau 24 – Types de données d'identification avec mise en correspondance spécifique à Profibus PA

Profibus - Nom d'attribut (nom sémantique)	Élément sémantique	Demande de données dans le dispositif physique	Nom spécifique à un protocole	Format de données Profibus	Format FDT (format d'affichage)	Référence de la spécification
identNumber, manufacturerSpecificExtension et slaveAddress tels que définis dans le tableau DP (Tableau 22).						
busProtocol	IdBusProtocol	Pour tous les dispositifs PA: "protocol_PA"	Protocol	-	enum	-
manufacturerId	IdManufacturer	Physical Block - Index 10 (Unsigned16 – 2 decimal)	DEVICE_MAN_ID	Unsigned16	UINT	[2] Partie 1, Section 3.11
device_id	-	Physical Block - Index 11 - VisibleString 16	DEVICE_ID	VisibleString16	STRING[16]	[2] Partie 1, Section 3.11
profile	-	Structure en blocs du bloc physique – élément 7: OctetString - (Index 10 – size 2)	Profile	OctetString2	UINT	[2]0 Partie 1, Section 3.7.2
profileRevision	IdBusProtocolVersion	Structure en blocs de bloc physique – élément 8 Unsigned 16 - (Index 6)	Profile Revision (révision de profil)	Unsigned16	UINT	[2] Partie 1, Section 3.7.2
softwareRevision	IdSoftwareRevision	Physical Block - Index 8 (VisibleString 16)	SOFTWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
hardwareRevision	IdHardwareRevision	Physical Block - Index 9 (VisibleString 16)	HARDWARE_REVISION	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
serialNumber	IdSerialNumber	Physical Block - Index 12 (VisibleString 16)	DEVICE_SER_NUM	VisibleString16	STRING[16]	[2] Part 1, Section 3.11
deviceTAG	IdTag	BO of PB - OctetString 32	TAG_DESC	OctetString32	STRING[16]	
profileSpecificType	-	Attribut FDT mis en correspondance avec Block Object PARENT_CLASS sorte de premier TransducerBlock	BO PARENT_CLASS	Unsigned16	UINT	[2] Partie 1, Section 3.7.2
GenericSupport	IdDTMSupportLevel	Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	Niveau de prise en charge de DTM		enum	

Le Tableau 25 et le Tableau 26 montrent les types de données d'identification avec une sémantique indépendante vis-à-vis du protocole.

Tableau 25 – Types de données d'identification simples avec sémantique indépendante vis-à-vis du protocole

Type de données	Définition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	enumeration • genericSupport • profileSupport • blockspecificProfileSupport • specificSupport • identSupport
match	STRING	Utilisé par le DTM de dispositif pour définir une expression générique qui doit concorder avec les informations d'identification de définition physique balayées
nomatch	STRING	Utilisé par le DTM de dispositif pour définir une expression générique qui ne doit pas concorder avec les informations d'identification de définition physique balayées utilisé par le DTM de dispositif pour indiquer si les informations d'identification ne peuvent pas concorder.

Tableau 26 – Types de données d'identification structurés avec sémantique indépendante vis-à-vis du protocole

Type de données	Définition			Description
	Types de données élémentaires	Usage	Multiplicité	
RegExpr	STRUCT			Inclut une chaîne d'expression générique – soit pour match (concordance), soit pour nomatch (pas de concordance)
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.4 Types de données de balayage de topologie

Ces types de données sont utilisés à la réponse de service Scan (Tableau 27 et Tableau 28).

Tableau 27 – Types de données simples de balayage de topologie

Type de données	Définition	Description
busAddress	USINT	Informations d'adresse selon la série IEC 61158 pour les voies accessibles par Profibus DPV1

Le type de données ProfibusDevice décrit une entrée dans la liste de dispositifs PROFIBUS balayés (voir Tableau 28).

Tableau 28 – Types de données structurés de balayage de topologie

Type de données	Définition			Description
	Type de données élémentaire	Usage	Multiplicité	
ProfibusDevice	STRUCT			Spécifie un dispositif Profibus.
	schemaVersion	O	[0..1]	
	busAddress	M	[1..1]	
	fdt:deviceTypeId	M	[1..1]	
	fdt:subDeviceType	O	[0..1]	

12.5 Types de données d'identification de balayage (Scan)

Ce paragraphe définit les types de données qui sont utilisés pour fournir un balayage spécifique à un protocole (voir Tableau 29 et Tableau 30).

Les types de données décrits dans le présent paragraphe sont utilisés dans les services suivants:

- Service "Scan"

Tableau 29 – Types de données simples d'identification de balayage

Type de données	Définition	Description
configuredState	enumeration (configuredAndPhysicallyAvailable configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	Un maître de communication doit indiquer dans cet attribut si la réponse de balayage est relative à un dispositif physique détecté qui est configuré ou non configuré.
resultState	enumeration (provisional final error)	Identifie si le résultat est celui des résultats provisoires ou le résultat final des résultats de balayage divisés.

Tableau 30 – Types de données structurés d'identification de balayage

Type de données	Définition			Description
	Types de données élémentaires	Usage	Multiplicité	
IdAddress	STRUCT			Tous les types de données contiennent exactement un attribut chacun incluant la valeur du dispositif physique balayé. Tous les types de données ayant une signification sémantique ont un préfixe "Id" pour une meilleure identification.
	profibusident:slaveAddress	M	[1..1]	
IdBusProtocol	STRUCT			
	profibusident:busProtocol	M	[1..1]	
IdBusProtocolVersion				
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	M	[1..1]	
IdManufacturer				
IdSerialNumber	STRUCT			
	profibusident:serialNumber	M	[1..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	M	[1..1]	
IdTypeID	STRUCT			
	profibusident:identNumber	M	[1..1]	
Manufacturer_id	STRUCT			Ces types de données contiennent des attributs correspondants définis dans l'IEC 62453-2. Ils sont transformés en paires nom-valeur sans signification sémantique pour Frame Application
	profibusident:manufacturer_id	M	[1..1]	
Order_id	STRUCT			
	profibusident:order_id	M	[1..1]	
ProfileID	STRUCT			
	profibusident:profileID	M	[1..1]	
ProfileSpecificType	STRUCT			
	profibusident:profileSpecificType	M	[1..1]	
ImVersion	STRUCT			
	profibusident:imVersion	M	[1..1]	
TagFunction	STRUCT			
	profibusident:tagFunction	M	[1..1]	
TagLocation	STRUCT			
	profibusident:tagLocation	M	[1..1]	
ImSupported	STRUCT			
	profibusident:imSupported	M	[1..1]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:software↗RevisionPA	M	[1..1]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardware↗RevisionPA	M	[1..1]	
ProfileRevisionPA	STRUCT			
	profibusident:profileRevisionPA	M	[1..1]	
Device_man_id	STRUCT			
	profibusident:device_man_id	M	[1..1]	
Device_id	STRUCT			
	profibusident:device_id	M	[1..1]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profileSpecific↗TypePA	M	[1..1]	
ManufacturerSpecific↗Extension				
	STRUCT			
	profibusident:manufacturer↗SpecificExtension	M	[1..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
ScanIdentification_IM	STRUCT			Ces éléments contiennent tous les éléments pour la variante de protocole appropriée
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	ImVersion	M	[1..1]	
	TagFunction	M	[1..1]	
	TagLocation	M	[1..1]	
	ImSupported	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_PA	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	
	ProfileSpecificTypePA	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	IdTag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentification_DP	STRUCT			
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
ScanIdentifications	STRUCT			Ensemble d'éléments ScanIdentification
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	choice of	M	[1..*]	
	ScanIdentification_DP	S	[0..*]	
	ScanIdentification_IM	S	[0..*]	
	ScanIdentification_PA	S	[0..*]	

12.6 Types de données d'identification de type de dispositif – fournis par DTM

Ce paragraphe définit les types de données qui sont utilisés pour fournir des informations spécifiques à un protocole pour des types de dispositifs (voir Tableau 31).

Les types de données décrits dans le présent paragraphe sont utilisés dans les services suivants:

- service GetIdentificationInformation

Tableau 31 – Types de données structurés d'identification de dispositif

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
IdBusProtocol	STRUCT			Tous les éléments contiennent exactement un attribut chacun incluant la valeur du dispositif physique balayé. Tous les éléments ayant une signification sémantique ont un préfixe "Id" pour une meilleure identification
	profibusident:busProtocol	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdBusProtocolVersion				
IdManufacturer				
IdTypeID	STRUCT			
	profibusident:identNumber	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevision	STRUCT			
	profibusident:softwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevision	STRUCT			
	profibusident:hardwareRevision	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Manufacturer_id	STRUCT			
	profibusident:manufacturer_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
Order_id	STRUCT			
	profibusident:order_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileID	STRUCT			
	profibusident:profileID	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificType	STRUCT			
	profibusident:profile↯SpecificType	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImVersion	STRUCT			
	profibusident:imVersion	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ImSupported	STRUCT			
	profibusident:imSupported	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdSoftwareRevisionPA	STRUCT			
	profibusident:software↯RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
IdHardwareRevisionPA	STRUCT			
	profibusident:hardware↯RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileRevisionPA	STRUCT			
	profibusident:profile↯RevisionPA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_man_id	STRUCT			
	profibusident:device_man_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
Device_id	STRUCT			
	profibusident:device_id	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ProfileSpecificTypePA	STRUCT			
	profibusident:profile↯SpecificTypePA	O	[0..1]	
	profibusident:RegExpr	O	[0..*]	
ManufacturerSpecific↯Extension	STRUCT			
	profibusident:manufacturer↯SpecificExtension	M	[1..1]	
DeviceIdentification_IM	STRUCT			Ces éléments contiennent tous les

Type de données	Définition			Description
	Types de données élémentaires	U s a g e	Multiplicité	
	profibusident:idDTM↗SupportLevel	M	[1..1]	éléments pour la variante de protocole appropriée
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	Manufacturer_id	M	[1..1]	
	Order_id	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	ProfileID	M	[1..1]	
	ProfileSpecificType	M	[1..1]	
	ImVersion	M	[1..1]	
	ImSupported	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification_PA	STRUCT			
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdSoftwareRevisionPA	M	[1..1]	
	IdHardwareRevisionPA	M	[1..1]	
	ProfileRevisionPA	M	[1..1]	
	Device_man_id	M	[1..1]	
	Device_id	M	[1..1]	
	ProfileSpecificTypePA	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification_DP	STRUCT			
	profibusident:idDTM↗SupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdTypeID	M	[1..1]	
	ManufacturerSpecific↗Extension	O	[0..1]	
DeviceIdentification s	STRUCT			Ensemble d'éléments DeviceIdentification
	fdt:protocolId	M	[1..1]	
	choice of	M	[1..*]	
	DeviceIdentification_DP	S	[0..*]	
	DeviceIdentification_IM	S	[0..*]	
	DeviceIdentification_PA	S	[0..*]	

12.7 Informations d'identification dans l'interface GUI

Il est obligatoire d'afficher les étiquettes des paramètres I&M dans l'objet Presentation des DTM ou l'interface utilisateur de Frame Application à partir de IM-Runtime.xml, afin d'assurer la cohérence et l'acceptation pour l'utilisateur.

13 ProfiSafe

13.1 Motivation

Le but du papier ProfiSafe-UseCase est de compiler un ensemble d'exemples types existants d'automation de sécurité intégrée à partir d'emplacements de fabrication discrets et continus et de dériver une approche systématique commune. Il est recommandé que cette solution proposée soit fondée sur

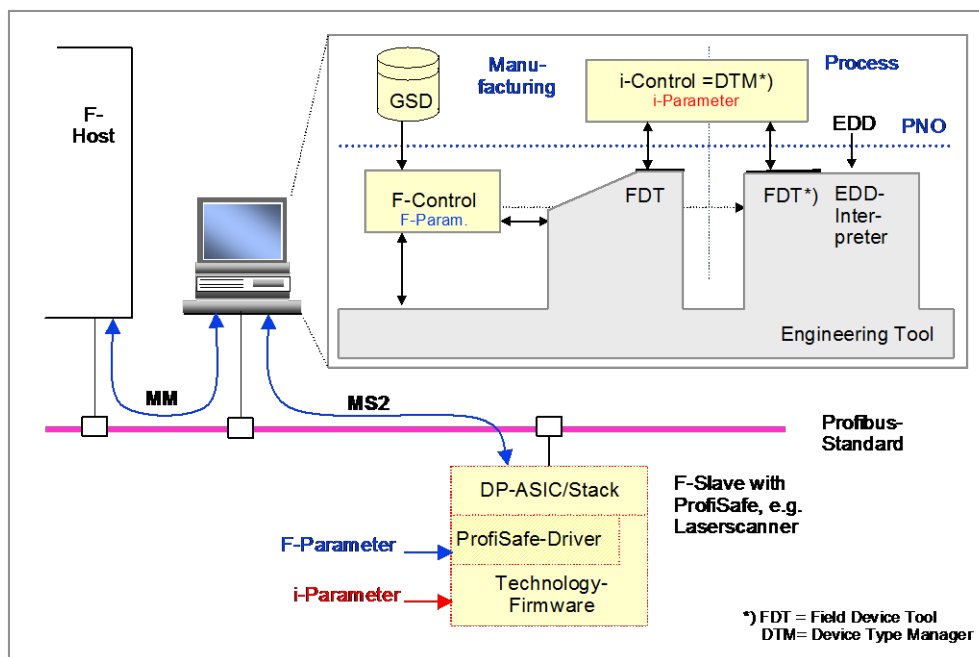
- les principes de communication de Profibus DPV1 et ProfiSafe;
- de nouvelles possibilités d'automates de sécurité intégrée combinés (programmes normalisés et à sécurité intégrée au sein d'un même automate programmable);
- et les activités courantes d'autres groupes de travail Profibus comme FDT/DTM et Proxy FB.

Dans le contexte du papier ProfiSafe-UseCase, la présente spécification accessible au public (PAS) couvre l'aspect partiel "Interface programmeurs FDT pour vérification de paramètres de dispositifs individuels".

13.2 Traitement général des paramètres

Afin de communiquer d'une manière sécurisée, chaque dispositif ProfiSafe (F-Slave) exige les dits F-Parameters (paramètres F) pour le réajustement du mode opérationnel de son pilote ProfiSafe. Ces paramètres comprennent le temps de chien de garde, le niveau d'intégrité de sécurité, la taille de conteneur, etc. Dans le cas des F-slaves (esclaves F) simples, aucun autre paramètre n'est requis. Les F-Parameters seront donc définis dans la GSD Révision 4.0.

En revanche, un F-slave (esclave F) complexe requiert des paramètres pertinents individuels et de sécurité supplémentaires (paramètre de dispositif individuel) qui, en raison de sa taille (>240 octets), ne peuvent pas être transportés par le télégramme de paramétrisation initial (Prm-Telegram). À titre d'amendement, des lignes directrices de ProfiSafe conseillent une méthode supplémentaire par le truchement de blocs fonctionnels proxy dans un automate programmable (PLC) de sécurité, fournissant ainsi une fonctionnalité supplémentaire comme la reparamétrisation commandée par programme, etc.



Légende

IEC 1130/09

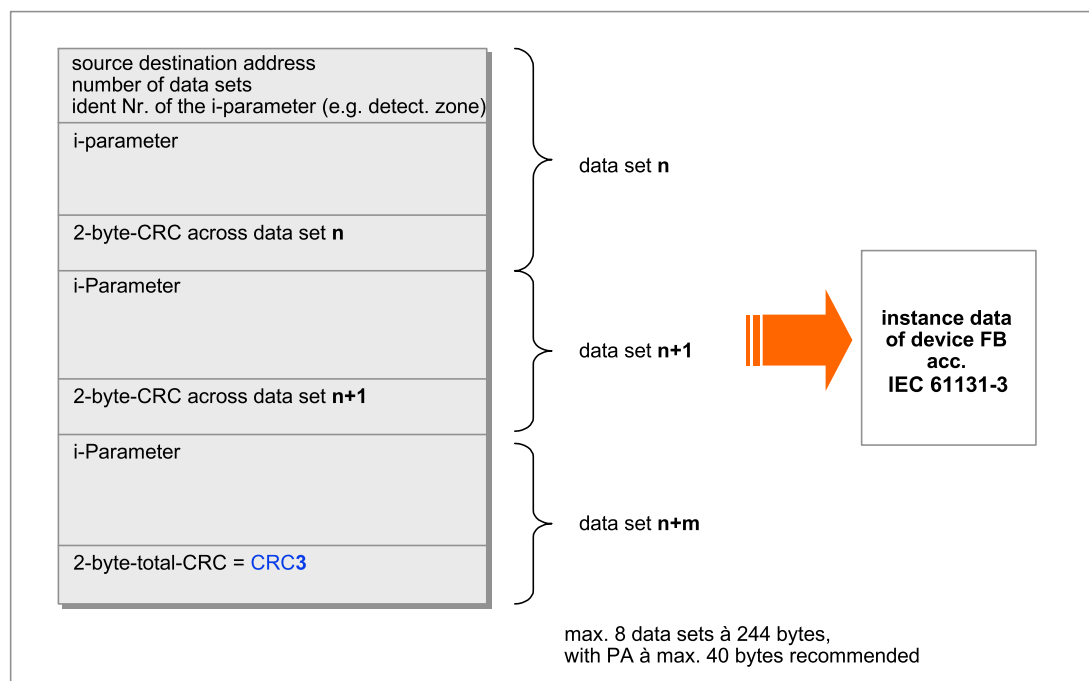
Anglais	Français
F-Host	Hôte F
Manufacturing Process	Processus de fabrication
Process	Processus
i-Control = DTM	Commande I = DTM
i-Parameter	Paramètre I
F-Parameter	Paramètre F
EDD-interpreter	Interpréteur EDD
Engineering Tool	Outil d'étude
Profibus Standard	Norme Profibus
F-Slave with ProfiSafe, e.g. Laserscanner	Esclave F avec ProfiSafe, par exemple: scanner à laser
ProfiSafe-Driver	Pilote ProfiSafe
Technology-Firmware	Micrologiciel («Firmware») Technologie
FDT = Field Device Tool	FDT = Outil de dispositif de terrain
DTM= Device Type Manager	DTM = Gestionnaire de type de dispositifs
DP-ASIC/STACK	DP-ASIC/PILE

Figure 4 – F-Parameter (paramètre F) et paramètre de dispositif individuel

Il est recommandé que les paramètres de dispositifs individuels, qui sont, par nature, spécifiques à un dispositif F, soient traités par les programmes de Gestionnaire de type de dispositif (DTM) provenant des fabricants de dispositifs F. L'outil d'étude fonctionne comme une Frame Application pour un tel DTM d'un esclave F. Il achemine les demandes de communication du DTM vers son dispositif (MS2) et procure une persistance de données. Autrement, il couvre toutes les tâches traditionnelles comme configuration de réseau, paramétrisation, mise en service et diagnostic. Il communique avec l'automate programmable par l'intermédiaire du Master-Master-Protocol (MM, protocole maître-maître). À cet égard, un automate programmable sur le Profibus est un dispositif de sa propre classe («Master Class 1», classe 1 de maître). L'outil d'étude lui-même est défini comme étant un dispositif "Master Class 2" (de classe 2 de maître).

13.3 Paramètre de dispositif individuel ProfiSafe

La structure du paramètre de dispositif individuel ProfiSafe est définie dans les lignes directrices ProfiSafe. Cet ensemble est enveloppé en format XML pour le transport à travers l'interface FDT.



IEC 1131/09

Légende

Anglais	Français
Source destination address	Adresse source destination
Number of data sets	Nombre de jeux de données
Ident. Nr. Of the i-parameter (e.g. detect. Zone)	Numéro d'identification du paramètre I (par exemple: zone de détection)
i-parameter	paramètre i
2 byte-CRC across data set n	CRC de deux octets à travers le jeu de données <i>n</i>
2 byte-total-CRC= CRC3	CRC total deux octets= CRC3
Data set <i>n</i>	Jeu de données <i>n</i>
Max. 8 data sets à 244 bytes, with PA à max. 40 bytes recommended	Max. 8 jeux de données à 244 octets, avec PA à max. 40 octets recommandés
Instance data of FB acc. IEC 61131-3	Données d'instances de FB selon l'IEC 61131-3
Data set <i>n + 1</i>	Jeu de données <i>n + 1</i>
Data set <i>n + m</i>	Jeu de données <i>n + m</i>

Figure 5 – Structure de données des paramètres de dispositifs individuels ProfiSafe

Les structures de données et les formats de données des paramètres internes de dispositifs individuels sont propriétaires. L'ensemble du jeu est divisé en parties qui s'inscrivent dans un jeu de données Profibus prêt pour le transport. Chaque jeu de données comprend un CRC de deux octets dans le but de vérifier la cohérence des données. Le polynôme est fixe et défini dans les lignes directrices ProfiSafe. Le CRC du dernier jeu vérifie le bloc total.

Bibliographie

- [1] PROFIBUS-PA Profile for Process Control Devices Version 3.0
 - [2] PROFIBUS Profile Guidelines, Part 1, Identification & Maintenance Functions, Version 1.1, May 2003
 - [3] PROFIBUS Guideline (Order No. 2.082): Chapter “Services on communication relationships MSAC_C2”: MSAC2_Initiate
 - [4] PROFIBUS Profile Amendment 3 to the Profibus Profile for Process Control Devices V3.0
 - [5] IEC 61158-6:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 6: Application layer protocol specification*
 - [6] PROFIBUS Specification Specification for PROFIBUS Device Description and Device Integration, Volume 1: GSD Version 5.04, July 2005
 - [7] IEC 61158-5:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 5: Application layer service specification*
 - [8] PROFIBUS Nutzerorganisation e.V., *PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998*
-

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch

This page has been left intentionally blank.