

CONSOLIDATED VERSION

VERSION CONSOLIDÉE



**Field device tool (FDT) Interface specification –
Part 315: Communication profile integration – IEC 61784 CPF 15**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 315: Intégration des profils de communication – IEC 61784 CPF 15**



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 315: Communication profile integration – IEC 61784 CPF 15**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 315: Intégration des profils de communication – IEC 61784 CPF 15**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.05; 35.110

ISBN 978-2-8322-3486-0

**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.**

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

REDLINE VERSION

VERSION REDLINE



**Field device tool (FDT) Interface specification –
Part 315: Communication profile integration – IEC 61784 CPF 15**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 315: Intégration des profils de communication – IEC 61784 CPF 15**

CONTENTS

FOREWORD.....	6
INTRODUCTION.....	8
1 Scope	9
2 Normative references	9
3 Terms, definitions, symbols, abbreviated terms and conventions	10
3.1 Terms and definitions	10
3.2 Abbreviated terms	10
3.3 Conventions	10
3.3.1 Data type names and references to data types	10
3.3.2 Vocabulary for requirements.....	10
4 Bus category	10
5 Access to instance and device data	12
5.1 Process Channel objects provided by DTM.....	12
5.2 DTM services to access instance and device data	12
6 Protocol specific behavior	12
6.1 General.....	12
6.2 Broadcasting	13
6.3 Unconfirmed private Modbus request	15
8 Protocol specific common data types	17
8.1 General.....	17
8.2 Address information.....	18
7 Protocol specific usage of general data types.....	17
9 Network management data types	18
10 Communication data types.....	19
10.1 General.....	19
10.2 Connection management data types.....	19
10.3 Transaction service specific data types	20
10.3.1 General	20
10.3.2 Data item addressing	20
10.3.3 Read coils transaction service.....	21
10.3.4 Read discrete inputs transaction service.....	22
10.3.5 Read holding registers transaction service.....	23
10.3.6 Read input registers transaction service	24
10.3.7 Write single coil transaction service.....	25
10.3.8 Write single register transaction service	26
10.3.9 Read exception status transaction service	27
10.3.10Diagnostics transaction service	27
10.3.11Get Comm event counter transaction service	28
10.3.12Get Comm event log transaction service.....	29
10.3.13Write multiple coils transaction service	31
10.3.14Write multiple registers transaction service.....	31
10.3.15Report slave ID transaction service	32
10.3.16Read file record transaction service.....	33
10.3.17Write file record transaction service.....	35
10.3.18Mask write register transaction service	36

10.3.19	Read/write holding registers transaction service	37
10.3.20	Read FIFO queue transaction service	38
10.3.21	Encapsulated interface transport transaction service	39
10.3.22	Read device identification transaction service	40
10.3.23	Private Modbus transaction service	43
10.3.24	Unconfirmed private Modbus transaction service	44
10.3.25	Modbus exception response	45
11	Channel parameter data types	46
12	Device Identification	48
12.1	Common device type identification data types	48
12.2	Topology scan data types	49
12.3	Scan identification data types	50
12.4	Device type identification data types – provided by DTM	52
12.5	Mapping of protocol specific device identification objects to FDT data types	53
	Bibliography	55
	Figure 1 – Part 315 of the IEC 62453 series	8
	Figure 2 – Broadcast sequence with Modbus Serial Line Communication DTM	14
	Figure 3 – Broadcast sequence with Modbus TCP Communication DTM	14
	Figure 4 – Broadcast sequence without Modbus Communication DTM	15
	Figure 5 – Unconfirmed request with Modbus Serial Line Communication DTM	16
	Figure 6 – Unconfirmed request with Modbus TCP Communication DTM	16
	Figure 7 – Unconfirmed request without Modbus Communication DTM	17
	Figure 8 – Data item addressing	21
	Table 1 – Protocol identifiers	10
	Table 2 – Address information for broadcast mode	13
	Table 3 – Broadcast transaction requests	13
	Table 4 – Protocol specific usage of general FDT data types	17
	Table 5 – Simple address information data types	18
	Table 6 – Structured address information data types	18
	Table 7 – Structured network management data types	19
	Table 8 – Simple common communication data types	19
	Table 9 – Structured connection management service data types	20
	Table 10 – Simple ReadCoilsReq data types	21
	Table 11 – Structured ReadCoilsReq data types	21
	Table 12 – Simple ReadCoilsRsp data types	22
	Table 13 – Structured ReadCoilsRsp data types	22
	Table 14 – Simple ReadDiscreteInputsReq data types	22
	Table 15 – Structured ReadDiscreteInputsReq data types	22
	Table 16 – Simple ReadDiscreteInputsRsp data types	23
	Table 17 – Structured ReadDiscreteInputsRsp data types	23
	Table 18 – Simple ReadHoldingRegistersReq data types	23
	Table 19 – Structured ReadHoldingRegistersReq data types	23
	Table 20 – Simple ReadHoldingRegistersRsp data types	24

Table 21 – Structured ReadHoldingRegistersRsp data types.....	24
Table 22 – Simple ReadInputRegistersReq data types.....	24
Table 23 – Structured ReadInputRegistersReq data types	24
Table 24 – Simple ReadInputRegistersRsp data types	25
Table 25 – Structured ReadInputRegistersRsp data types.....	25
Table 26 – Simple WriteSingleCoilReq data types	25
Table 27 – Structured WriteSingleCoilReq data types	25
Table 28 – Structured WriteSingleCoilRsp data types	26
Table 29 – Simple WriteSingleRegisterReq data types.....	26
Table 30 – Structured WriteSingleRegisterReq data types	26
Table 31 – Structured WriteSingleRegisterRsp data types	26
Table 32 – Structured ReadExceptionStatusReq data types.....	27
Table 33 – Simple ReadExceptionStatusRsp data types	27
Table 34 – Structured ReadExceptionStatusRsp data types	27
Table 35 – Simple diagnostics data types.....	28
Table 36 – Structured DiagnosticsReq data types.....	28
Table 37 – Structured DiagnosticsRsp data types	28
Table 38 – Structured GetCommEventCounterReq data types	29
Table 39 – Simple GetCommEventCounterRsp data types	29
Table 40 – Structured GetCommEventCounterRsp data types.....	29
Table 41 – Structured GetCommEventLogReq data types	30
Table 42 – Simple GetCommEventLogRsp data types	30
Table 43 – Structured GetCommEventLogRsp data types	30
Table 44 – Simple WriteMultipleCoilsReq data types	31
Table 45 – Structured WriteMultipleCoilsReq data types	31
Table 46 – Structured WriteMultipleCoilsRsp data types	31
Table 47 – Simple WriteMultipleRegistersReq data types.....	32
Table 48 – Structured WriteMultipleRegistersReq data types	32
Table 49 – Structured WriteMultipleRegistersRsp data types	32
Table 50 – Structured ReportSlaveIDReq data types	32
Table 51 – Simple ReportSlaveIDRsp data types.....	33
Table 52 – Structured ReportSlaveIDRsp data types.....	33
Table 53 – Simple ReadFileSubRequest data types.....	33
Table 54 – Structured ReadFileSubRequest data types	34
Table 55 – Structured ReadFileRecordReq data types	34
Table 56 – Simple ReadFileSubResponse data types	34
Table 57 – Structured ReadFileSubResponse data types.....	34
Table 58 – Structured ReadFileRecordRsp data types	35
Table 59 – Simple WriteFileSubRequest data types.....	35
Table 60 – Structured WriteFileSubRequest data types	35
Table 61 – Structured WriteFileRecordReq data types	36
Table 62 – Structured WriteFileRecordRsp data types	36
Table 63 – Simple MaskWriteRegisterReq data types	36

Table 64 – Structured MaskWriteRegisterReq data types.....	37
Table 65 – Structured MaskWriteRegisterRsp data types.....	37
Table 66 – Simple ReadWriteRegistersReq data types	37
Table 67 – Structured ReadWriteRegistersReq data types	38
Table 68 – Simple ReadWriteRegistersRsp data types.....	38
Table 69 – Structured ReadWriteRegistersRsp data types	38
Table 70 – Simple ReadFifoQueueReq data types	38
Table 71 – Structured ReadFifoQueueReq data types.....	38
Table 72 – Simple ReadFifoQueueRsp data types	39
Table 73 – Structured ReadFifoQueueRsp data types.....	39
Table 74 – Simple EncapsulatedInterfaceTransportReq data types	39
Table 75 – Structured EncapsulatedInterfaceTransportReq data types	40
Table 76 – Simple EncapsulatedInterfaceTransportRsp data types	40
Table 77 – Structured EncapsulatedInterfaceTransportRsp data types.....	40
Table 78 – Simple ReadDeviceIdentificationReq data types	41
Table 79 – Structured ReadDeviceIdentificationReq data types.....	41
Table 80 – Simple IdentificationObject data types.....	41
Table 81 – Structured IdentificationObject data types	41
Table 82 – Simple ReadDeviceIdentificationRsp data types	42
Table 83 – Structured ReadDeviceIdentificationRsp data types.....	43
Table 84 – Simple PrivateModbusReq data types	43
Table 85 – Structured PrivateModbusReq data types.....	43
Table 86 – Simple PrivateModbusRsp data types	43
Table 87 – Structured PrivateModbusRsp data types	44
Table 88 – Simple UnconfirmedPrivateModbusReq data types	44
Table 89 – Structured UnconfirmedPrivateModbusReq data types.....	44
Table 90 – Structured UnconfirmedPrivateModbusRsp data types.....	44
Table 91 – Simple ModbusExceptionRsp data types	45
Table 92 – Structured ModbusExceptionRsp data types.....	45
Table 93 – Simple channel parameter data types.....	46
Table 94 – Structured channel parameter data types	47
Table 95 – Simple common identification data types	48
Table 96 – Simple device information data types	49
Table 97 – Structured device information data types.....	50
Table 98 – Simple scan identification data types.....	51
Table 99 – Structured scan identification data types	51
Table 100 – Structured device type identification data types	52
Table 101 – Mapping of protocol specific identification objects in FDT	54
Table 102 – Additional information for optional identification objects	54
Table 103 – Physical layer identifiers for Modbus TCP.....	11
Table 104 – Physical layer identifiers for Modbus Serial	12

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

**Part 315: Communication profile integration –
IEC 61784 CPF 15**

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 62453-315 bears the edition number 1.1. It consists of the first edition (2009-07) [documents 65E/131/FDIS and 65E/144/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 62453-315 has 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 document using a colour printer.

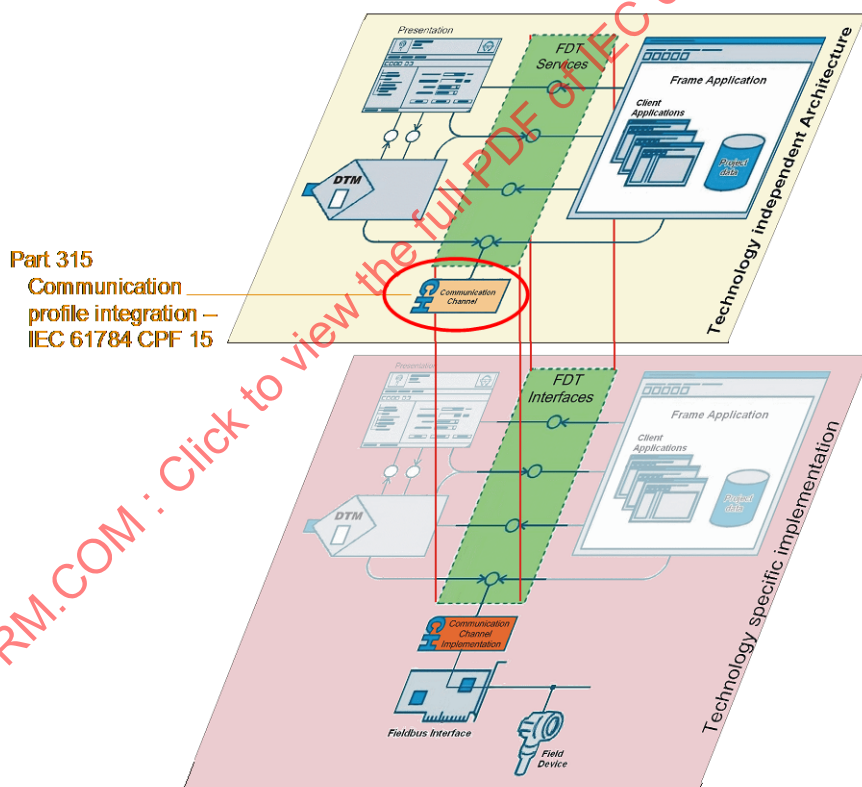
INTRODUCTION

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 fieldbuses 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 fieldbuses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

Figure 1 shows how IEC 62453-315 is aligned in the structure of the IEC 62453 series.



IEC 1136/09

Figure 1 – Part 315 of the IEC 62453 series

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 315: Communication profile integration – IEC 61784 CPF 15

1 Scope

Communication Profile Family 15 (commonly known as Modbus¹) defines communication profiles based on IEC 61158-5-15 and IEC 61158-6-15. The basic profile CP 15/1 (Modbus TCP) is defined in IEC 61784-1. An additional communication profile (Modbus Serial Line) is defined in [2].

This part of the IEC 62453 provides information for integrating Modbus TCP® and Modbus Serial Line® protocol support into FDT based systems.

NOTE This part of IEC 62453 only specifies the mapping of Modbus parameters to FDT data types. For restrictions of protocol specific parameters concerning allowed values and concerning limitations of arrays used in the definition of FDT data types, refer to IEC 61158-5-15 and the MODBUS Application Protocol Specification.

2 Normative references

The following referenced documents are indispensable for the application of this document. 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, *Programmable controllers – Part 3: Programming languages*

IEC 61158-5-15, *Industrial communication networks – Fieldbus specifications – Part 5-15: Application layer service definition – Type 15 elements*

IEC 61158-6-15, *Industrial communication networks – Fieldbus specifications – Part 6-20: Application layer protocol specification – Type 15 elements*

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

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

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*

RFC 791, *Internet Protocol* (available at <<http://www.ietf.org/rfc/rfc0791.txt>>)

1) Modbus is the trademark of Schneider Automation Inc. It is registered in the United States of America. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trademark Modbus. Use of the trademark Modbus requires permission from Schneider Automation Inc.

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, IEC 62453-2 and the following apply.

3.2 Abbreviated terms

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

ASCII	American Standard Code for Information Interchange
DTM	Device Type Manager
FA	Frame Application
IP	Internet Protocol
RFC	Request For comment
TCP	Transmission Control Protocol

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

4 Bus category

The Modbus protocol is identified in the protocolId element of the structured data type 'fdt:BusCategory' by the following unique identifiers (see Table 1).

Table 1 – Protocol identifiers

Identifier value	ProtocolId name	Description
59629a40-285f-11db-a98b-0800200c9a66	'Modbus over Serial Line'	Modbus over Serial Line
59629a41-285f-11db-a98b-0800200c9a66	'Modbus over TCP'	Modbus over TCP

Modbus TCP is using the following unique identifiers in physicalLayer members within PhysicalLayer data type (see Table 103):

Table 103 – Physical layer identifiers for Modbus TCP

Identifier value	Description
307DD810-C010-11DB-90E7-0002B3ECDCBE	100BaseTXFD (default for Media Type Copper)
307DD812-C010-11DB-90E7-0002B3ECDCBE	100BaseFXFD (default for Media Type Fiber Optic)
307DD813-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD816-C010-11DB-90E7-0002B3ECDCBE	1000BaseXFD
307DD818-C010-11DB-90E7-0002B3ECDCBE	1000BaseLXFD
307DD81A-C010-11DB-90E7-0002B3ECDCBE	1000BaseSXF
307DD81C-C010-11DB-90E7-0002B3ECDCBE	1000BaseTFD
307DD81D-C010-11DB-90E7-0002B3ECDCBE	10GbaseFX
307DD81E-C010-11DB-90E7-0002B3ECDCBE	10GbaseLX4
307DD81F-C010-11DB-90E7-0002B3ECDCBE	10GBaseR
307DD820-C010-11DB-90E7-0002B3ECDCBE	10GbaseER
307DD821-C010-11DB-90E7-0002B3ECDCBE	10GbaseLR
307DD822-C010-11DB-90E7-0002B3ECDCBE	10GbaseSR
307DD823-C010-11DB-90E7-0002B3ECDCBE	10GbaseW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseEW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseLW
307DD825-C010-11DB-90E7-0002B3ECDCBE	10GbaseSW
307DD826-C010-11DB-90E7-0002B3ECDCBE	10GbaseCX4
307DD827-C010-11DB-90E7-0002B3ECDCBE	2BaseTL
307DD828-C010-11DB-90E7-0002B3ECDCBE	10PassTS
307DD829-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10D
307DD82A-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10U
307DD82B-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD82C-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10D
307DD82D-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10U
307DD82F-C010-11DB-90E7-0002B3ECDCBE	1000BaseLX10
307DD830-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10D
307DD831-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10U
307DD832-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20D
307DD833-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20U
307DD834-C010-11DB-90E7-0002B3ECDCBE	10GBaseT or 100BasePXFD
307DD835-C010-11DB-90E7-0002B3ECDCBE	10GBaseLRM
307DD836-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX
307DD837-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX4
307DD838-C010-11DB-90E7-0002B3ECDCBE	1000BaseKR
307DD839-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXD1
307DD83A-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR2
307DD83B-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR3
307DD83C-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXU1
307DD83D-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR2U
307DD83E-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR3U
307DD83F-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD1
307DD840-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD2

Identifier value	Description
307DD841-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD3
307DD842-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU1
307DD843-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU3
307DD844-C010-11DB-90E7-0002B3ECDCBE	40GbaseKR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseCR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseSR4
307DD846-C010-11DB-90E7-0002B3ECDCBE	40GbaseFR
307DD847-C010-11DB-90E7-0002B3ECDCBE	40GbaseLR4
307DD848-C010-11DB-90E7-0002B3ECDCBE	100GbaseCR10
307DD849-C010-11DB-90E7-0002B3ECDCBE	100GbaseSR10
307DD84A-C010-11DB-90E7-0002B3ECDCBE	100GbaseLR4
307DD84B-C010-11DB-90E7-0002B3ECDCBE	100GbaseER4
307DD84C-C010-11DB-90E7-0002B3ECDCBE	100BasePXFD
307DD84D-C010-11DB-90E7-0002B3ECDCBE	Radio Communication
307DD84E-C010-11DB-90E7-0002B3ECDCBE	Speed of 100 Mbit/s (and more) and full duplexity

Modbus Serial is using the following unique identifiers in physicalLayer members within PhysicalLayer data type (see Table 104):

Table 104 – Physical layer identifiers for Modbus Serial

Identifier value	Description
C0458028-F240-45A5-8664-70DC84FDC6FE	RS-232
3BF008DC-5A44-4220-8C3E-3C46A589A0B4	RS-422
036D1591-387B-11D4-86E1-00E0987270B9	RS-485

The DataLinkLayer property is not applicable for Modbus and has to be set to null.

5 Access to instance and device data

5.1 Process Channel objects provided by DTM

For Modbus no minimum set of parameters is defined which must be provided by a DTM. If a DTM provides process values, the process variables should be modeled as Process Channels.

5.2 DTM services to access instance and device data

Not applicable.

6 Protocol specific behavior

6.1 General

This clause specifies how unconfirmed Modbus requests shall be handled in FDT. Two types of unconfirmed Modbus requests are supported in FDT:

- broadcasts;
- unconfirmed private Modbus requests.

6.2 Broadcasting

In broadcast mode, a DTM can send a Modbus request to all devices connected to the bus. This mode is only supported for devices, which are connected via Modbus Serial Line. The connection can either be a direct connection or a connection via a gateway. The broadcast mode must be initiated by a fdtmodbus:ConnectRequest with the slave address of the target device set to 0 (slaveAddress=0). For more detailed information about the address information which has to be provided for broadcast mode, see Table 2.

Table 2 – Address information for broadcast mode

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ModbusSerial	STRUCT			Address information for Modbus Serial Line
	slaveAddress	M	[1..1]	Shall be 0 for broadcast mode
ModbusTCP	STRUCT			Address information for Modbus TCP:
	slaveAddress	M	[1..1]	Shall be 0 for broadcast mode
	tcpAddress	M	[1..1]	IP address of the Modbus TCP/ Modbus Serial Line gateway
	tcpPort	O	[0..1]	Port for Modbus TCP connection (if no port information is provided, the default port 502 shall be used)

Because in broadcast mode no response will be returned by the device, the broadcast mode shall be used only with the transaction requests shown in Table 3:

Table 3 – Broadcast transaction requests

Broadcast transaction requests	Restrictions
WriteSingleCoilReq	None
WriteSingleRegisterReq	None
DiagnosticsReq	This transaction request shall be used in broadcast mode only with the following sub-functions: - 0x01: Restart Communication Option - 0x03: Change ASCII Input Delimiter - 0x04: Force Listen Only Mode - 0x0A: Clear Counters and Diagnostic Register - 0x14: Clear Overrun Counter and Flag
WriteMultipleCoilsReq	None
WriteMultipleRegistersReq	None
WriteFileRecordReq	None
MaskWriteRegisterReq	None
PrivateModbusReq	Shall be used only with private services where no response is required from the device

Although in broadcast mode no response will be returned by the target devices a transaction response, which corresponds to the transaction request shall be generated. This transaction response shall be generated in order to inform the DTM that the broadcast request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it has to provide this generated transaction response (see Figure 2 and Figure 3).

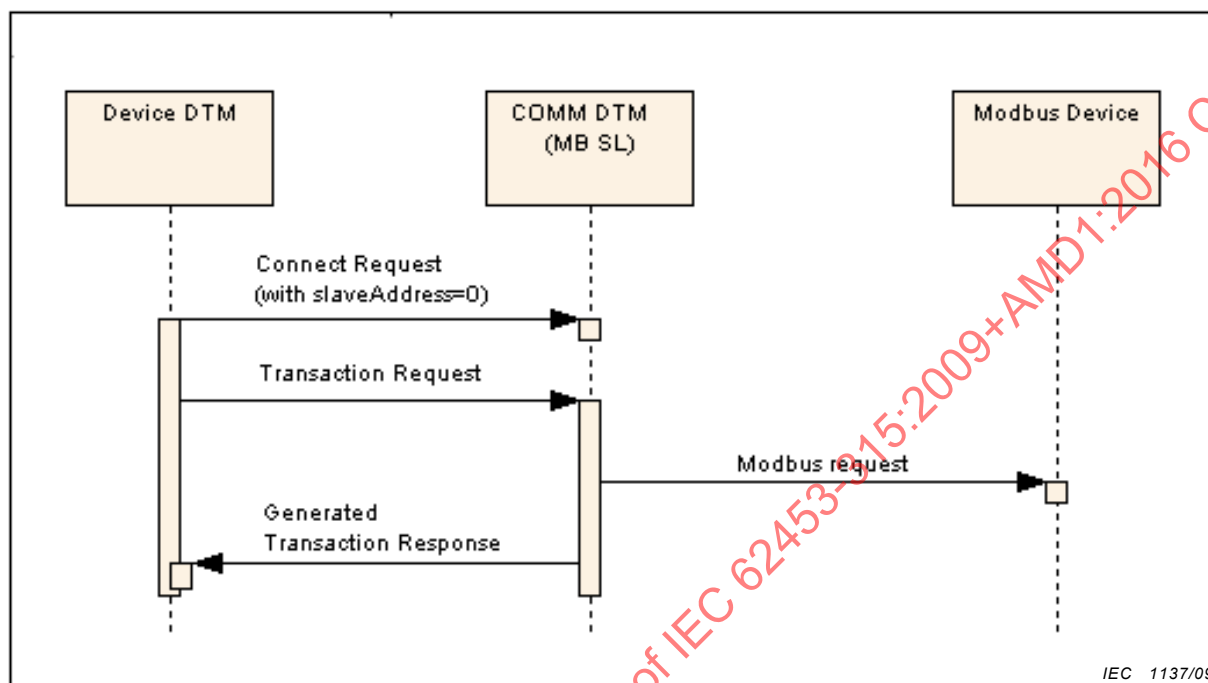


Figure 2 – Broadcast sequence with Modbus Serial Line Communication DTM

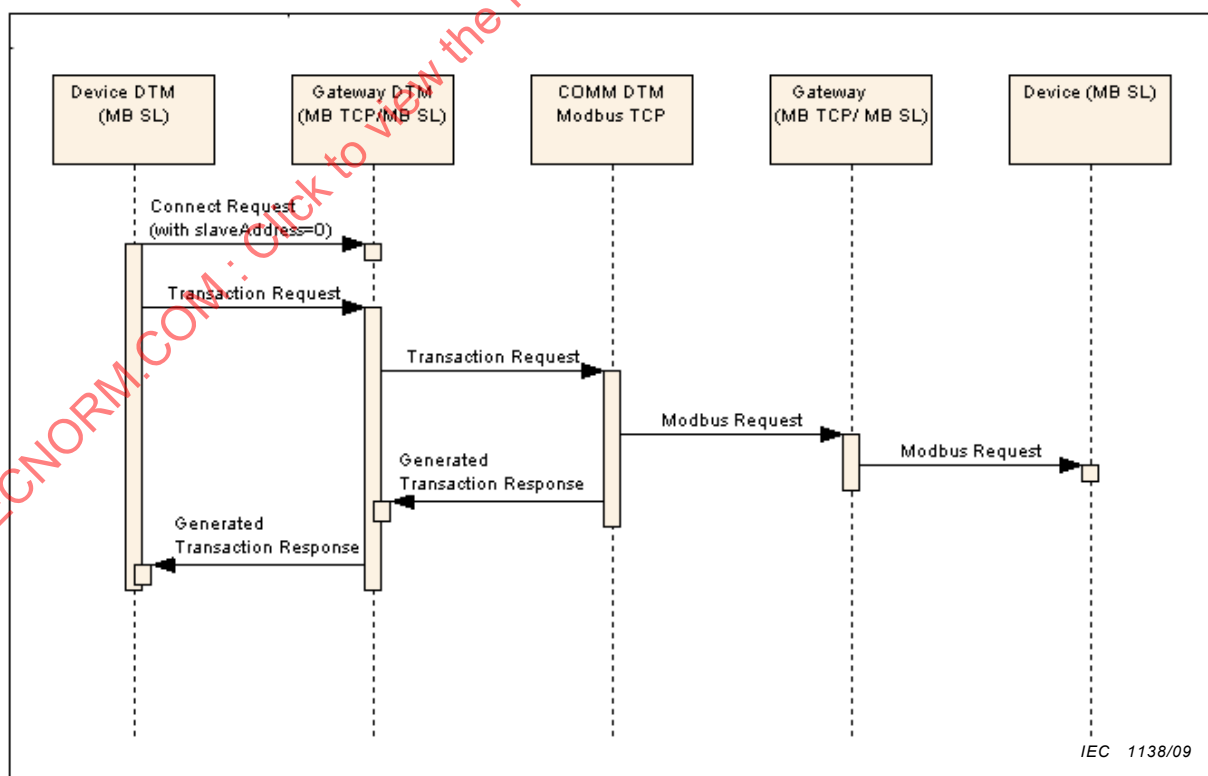


Figure 3 – Broadcast sequence with Modbus TCP Communication DTM

If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, has to provide this transaction response (see Figure 4).

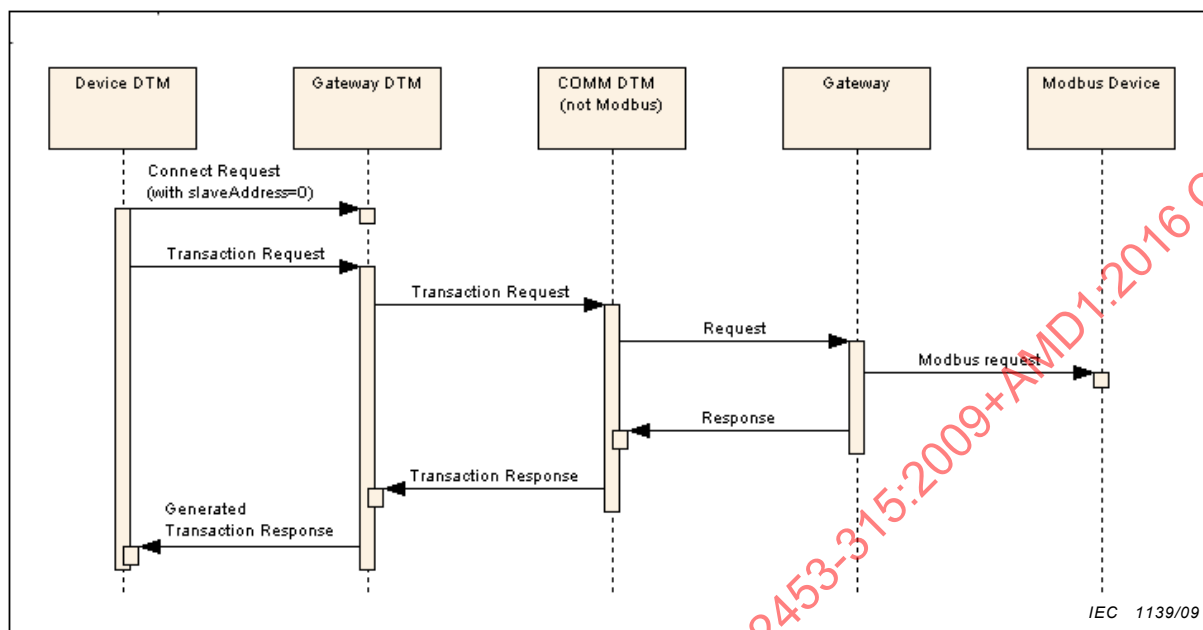


Figure 4 – Broadcast sequence without Modbus Communication DTM

6.3 Unconfirmed private Modbus request

This transaction request can be used to send any unconfirmed private Modbus request. Although no response will be returned by the target device, a transaction response, which corresponds to the transaction request, shall be generated. This transaction response shall be generated in order to inform the DTM that the unconfirmed request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it has to provide this generated transaction response (see Figure 5 and Figure 6).

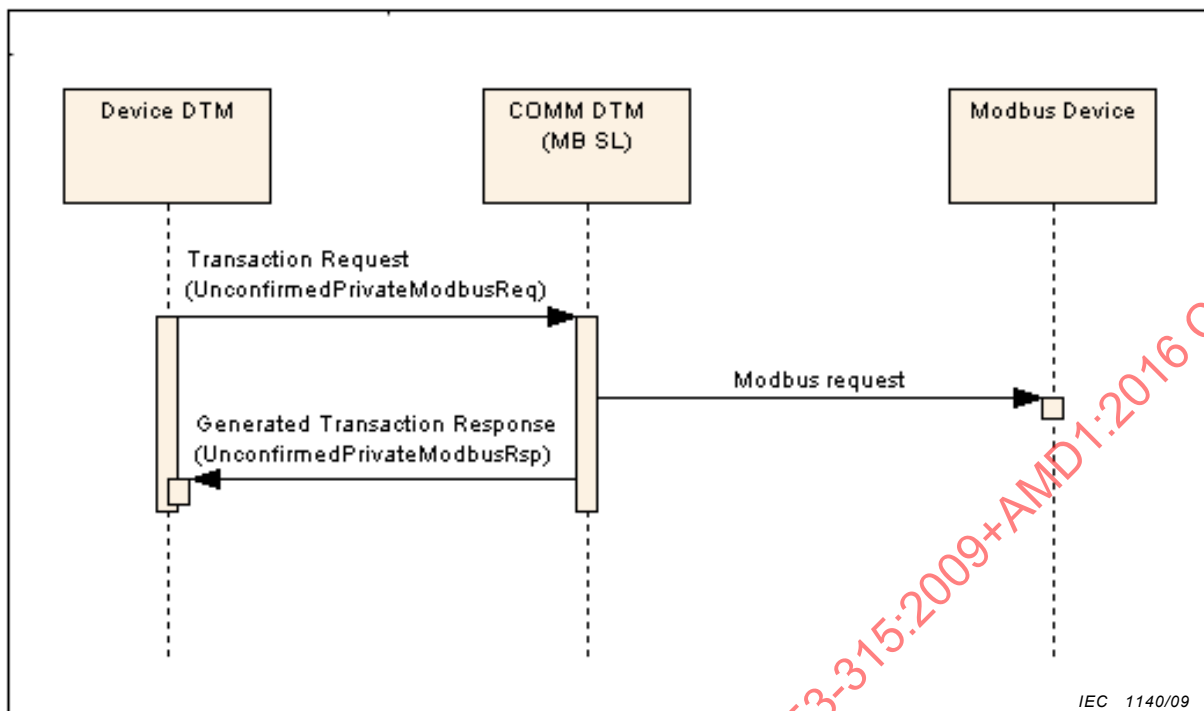


Figure 5 – Unconfirmed request with Modbus Serial Line Communication DTM

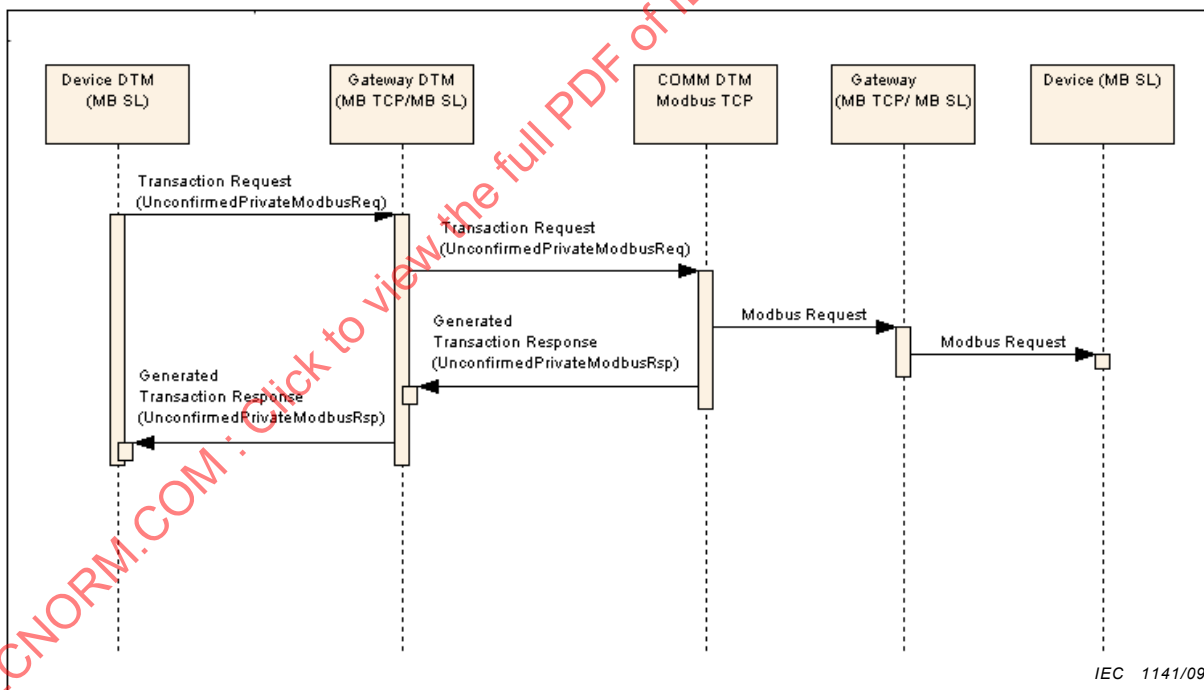


Figure 6 – Unconfirmed request with Modbus TCP Communication DTM

If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, has to provide this transaction response (see Figure 7).

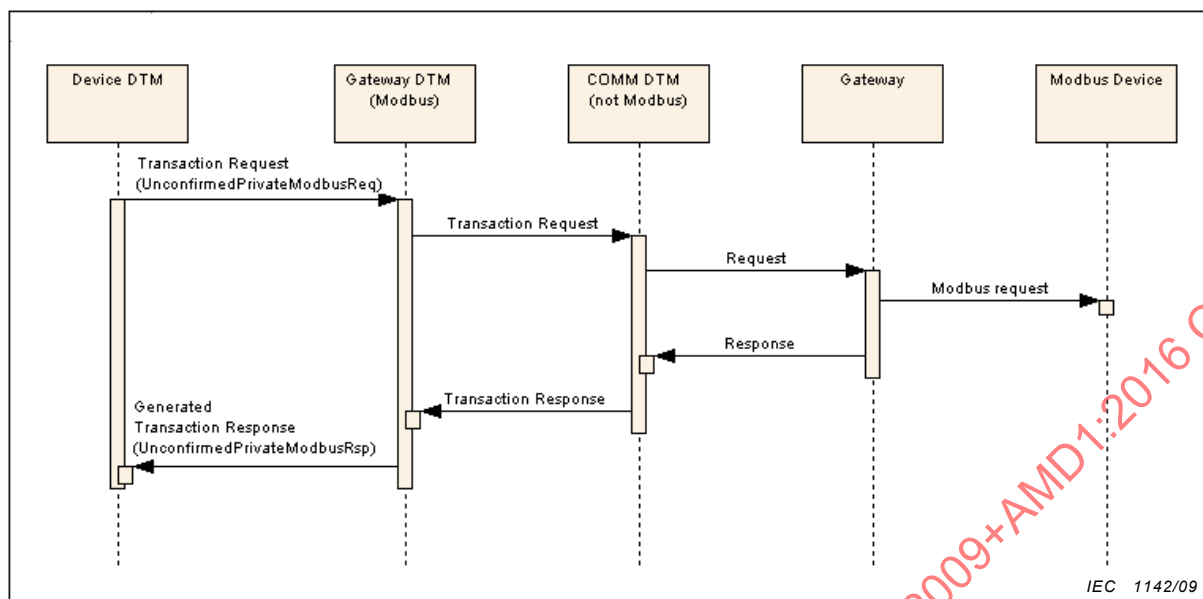


Figure 7 – Unconfirmed request without Modbus Communication DTM

7 Protocol specific usage of general data types

The following table (Table 4) shows how general data types, defined in IEC 62453-2 within the namespace 'fdt', are used with IEC 61784 CPF 15 devices.

Table 4 – Protocol specific usage of general FDT data types

Data type	Description for use
fdt:dataType	Identifier for the FDT data type of the channel object
fdt:busCategory	See Clause 4
deviceTypeInfoInformation	Modbus device description in form of human readable string, which can be provided with a DTM
deviceTypeInfoInformationPath	Path to file which contains the Modbus device description if available
fdt:readAccess	Specifies whether the value can be read from the device: “0”, if reading access is not allowed “1”, if reading access is allowed
fdt:systemTag	Unique identifier of a device instance within a project of a Frame Application
fdt:writeAccess	Specifies whether the value can be written to the device: “0”, if writing access is not allowed “1”, if writing access is allowed

8 Protocol specific common data types

8.1 General

This clause specifies protocol specific common data types which are used in the definition of other data types.

8.2 Address information

This subclause specifies the structure and the semantics of the protocol specific address information in FDT, which is used on the different underlying communication layers for Modbus TCP and Modbus Serial Line.

For a detailed description of how the address information is mapped to data types, see Table 5 and Table 6.

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

Table 5 – Simple address information data types

Data type	Definition	Description
slaveAddress	USINT	Slave address of a Modbus device
tcpAddress	STRING	String representation of the IP address as described in RFC 791 Internet Protocol (IP)
tcpPort	UINT	Port for Modbus TCP connection (if no port information is provided the default port 502 shall be used)

Table 6 – Structured address information data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ModbusSerial	STRUCT			Address information for Modbus Serial Line
	slaveAddress	M	[1..1]	
ModbusTCP	STRUCT			Address information for Modbus TCP: IP address of the target Modbus TCP server Slave address of the target device if the Modbus TCP server provides routing services over Modbus Serial Line
	slaveAddress	O	[0..1]	
	tcpAddress	M	[1..1]	
	tcpPort	O	[0..1]	

9 Network management data types

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

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

This clause describes the information, which has to be provided for Modbus devices, in the UserDefinedBus data type (see Table 7).

Table 7 – Structured network management data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UserDefinedBus	STRUCT			The UserDefinedBus data type contains the address information of the device. It is required that the UserDefinedBus data type contains at least one address information data type of the target device
	mb:ModbusSerial	O	[0..*]	
	mb:ModbusTCP	O	[0..*]	

Some devices and their modules require more than one address on the bus. It is possible that the UserDefinedBus data type may contain several address-elements. In this case, the first address element shall contain the information about the principal address of the device.

10 Communication data types

10.1 General

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

- connect service;
- disconnect service;
- transaction service.

This clause describes the structure and the semantics of the protocol specific communication related data, which is exchanged between a DTM and its parent component.

For a detailed description of the common communication data types used in these services, see Table 8.

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

Table 8 – Simple common communication data types

Data type	Definition	Description
communicationReference	UUID	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

10.2 Connection management data types

For a detailed description of the information used in the connect service and in the disconnect service see Table 9.

Table 9 – Structured connection management service data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
Abort	STRUCT			If the Abort command is sent to the next communication component or to the connected device, it terminates all pending requests and returns without waiting for a result. The termination of the connection will not be confirmed
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Describes the communication request: – to establish a communication link to a Modbus TCP server device; – to establish a communication link to a Modbus Serial Line slave device. The request contains the address information of the target device
	fdt:systemTag	M	[1..1]	
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
ConnectResponse	STRUCT			Describes the communication response to the connect request
	communicationReference	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request to release a communication link
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response to a disconnect request
	communicationReference	M	[1..1]	

10.3 Transaction service specific data types

10.3.1 General

This subclause describes the information which is used in the transaction services.

10.3.2 Data item addressing

This document specifies the mapping of Modbus parameters to FDT data types. For this reason, all address information provided as parameters and concerning the Modbus data items

- coils,
- discrete inputs,
- holding registers and
- input registers,

shall be the same information as will be later sent on the bus (see Figure 8).

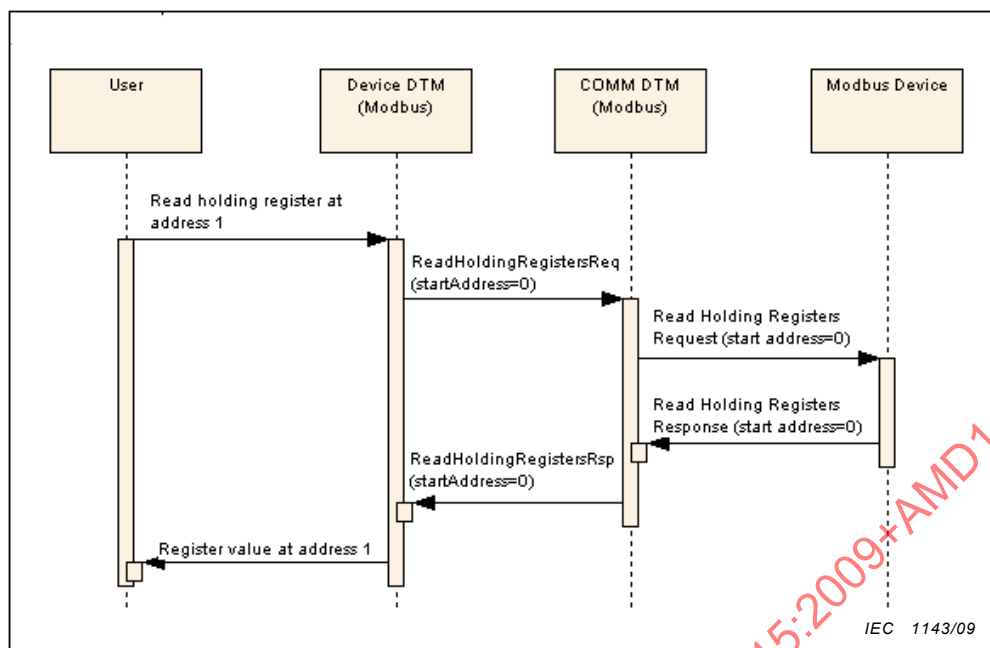


Figure 8 – Data item addressing

10.3.3 Read coils transaction service

The parameters of the transaction service, which is equivalent to the Read Coils service defined in IEC 61158-5-15, are described in Table 10, Table 11, Table 12 and Table 13.

Table 10 – Simple ReadCoilsReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first coil to be read	Address of first coil
quantity	UINT	Number of coils to be read	Quantity of coils

Table 11 – Structured ReadCoilsReq data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ReadCoilsReq	STRUCT			Communication request equivalent to the Read Coils service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Table 12 – Simple ReadCoilsRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
multipleCoilValues	STRING	ASCII string with each coil state coded in one character, where the first character in the string represents the first coil to be read: – “0” = FALSE or “OFF” – “1” = TRUE or “ON”	Data

Table 13 – Structured ReadCoilsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadCoilsRsp	STRUCT			Communication response equivalent to the Read Coils service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	multipleCoilValues	M	[1..1]	

10.3.4 Read discrete inputs transaction service

The parameters of the transaction service, which is equivalent to the Read Discretes service defined in IEC 61158-5-15, are described in Table 14, Table 15, Table 16 and Table 17.

Table 14 – Simple ReadDiscreteInputsReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first discrete input to be read	Address of first discrete
quantity	UINT	Number of discrete inputs to be read	Quantity of discretes

Table 15 – Structured ReadDiscreteInputsReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDiscreteInputsReq	STRUCT			Communication request equivalent to the Read Discretes service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Table 16 – Simple ReadDiscreteInputsRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
discreteInputsStatus	STRING	ASCII string with each discrete input state coded in one character, where the first character in the string represents the first discrete input to be read:: – “0”= FALSE or “OFF” – “1”= TRUE or “ON”	Data

Table 17 – Structured ReadDiscreteInputsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDiscreteInputsRsp	STRUCT			Communication response equivalent to the Read Discretes service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	discreteInputsStatus	M	[1..1]	

10.3.5 Read holding registers transaction service

The parameters of the transaction service, which is equivalent to the Read Holding Registers service defined in IEC 61158-5-15, are described in Table 18, Table 19, Table 20 and Table 21.

Table 18 – Simple ReadHoldingRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first holding register to be read	Address of first holding register to read
quantity	UINT	Number of holding registers to be read	Quantity of holding registers to read

Table 19 – Structured ReadHoldingRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadHoldingRegistersReq	STRUCT			Communication request equivalent to the Read Holding Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	startAddress	M	[1..1]	
	quantity	M	[1..1]	

Table 20 – Simple ReadHoldingRegistersRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
registerValues	ARRAY OF WORD	Read holding register values	Data

Table 21 – Structured ReadHoldingRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadHoldingRegistersRsp	STRUCT			Communication response equivalent to the Read Holding Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	registerValues	M	[1..1]	

10.3.6 Read input registers transaction service

The parameters of the transaction service, which is equivalent to the Read Input Registers service defined in IEC 61158-5-15, are described in Table 22, Table 23, Table 24 and Table 25.

Table 22 – Simple ReadInputRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first input register to be read	Address of first input register
quantity	UINT	Number of input registers to be read	Quantity of input registers

Table 23 – Structured ReadInputRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadInputRegistersReq	STRUCT			Communication request equivalent to the Read Input Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Table 24 – Simple ReadInputRegistersRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
registerValues	ARRAY OF WORD	Read input register values	Data

Table 25 – Structured ReadInputRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadInputRegistersRsp	STRUCT			Communication response equivalent to the Read Input Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	registerValues	M	[1..1]	

10.3.7 Write single coil transaction service

The parameters of the transaction service, which is equivalent to the Write Single Coil service defined in IEC 61158-5-15, are described in Table 26, Table 27 and Table 28.

Table 26 – Simple WriteSingleCoilReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
outputAddress	UINT	Address of the coil to be forced	Address of first coil
singleCoilValue	BOOL	Coil state to be forced, with: – “0”=FALSE or “OFF” – “1”=TRUE or “ON”	Data single coil

Table 27 – Structured WriteSingleCoilReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleCoilReq	STRUCT			Communication request equivalent to the Write Single Coil service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	singleCoilValue	M	[1..1]	

Table 28 – Structured WriteSingleCoilRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleCoilRsp	STRUCT			Communication response equivalent to the Write Single Coil service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.8 Write single register transaction service

The parameters of the transaction service, which is equivalent to the Write Single Holding Register service defined in IEC 61158-5-15, are described in Table 29, Table 30 and Table 31.

Table 29 – Simple WriteSingleRegisterReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
outputAddress	UINT	Address of the holding register to be written	Address of first holding register to write
singleRegister	WORD	Value to be written to the register	Data

Table 30 – Structured WriteSingleRegisterReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleRegisterReq	STRUCT			Communication request equivalent to the Write Single Holding Register service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	singleRegister	M	[1..1]	

Table 31 – Structured WriteSingleRegisterRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleRegisterRsp	STRUCT			Communication response equivalent to the Write Single Holding Register service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.9 Read exception status transaction service

The parameters of the transaction service, which is equivalent to the Read Exception Status service defined in the MODBUS Application Protocol specification, are described in Table 32, Table 33 and Table 34.

Table 32 – Structured ReadExceptionStatusReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadExceptionStatusReq	STRUCT			Communication request equivalent to the Read Exception Status service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line				

Table 33 – Simple ReadExceptionStatusRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
exceptionStatus	BYTE	Exception status of Modbus Serial Line device	Output Data

Table 34 – Structured ReadExceptionStatusRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadExceptionStatusRsp	STRUCT			Communication response equivalent to the Read Exception Status service response defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	exceptionStatus	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.10 Diagnostics transaction service

The parameters of the transaction service, which is equivalent to the Diagnostics service defined in the MODBUS Application Protocol specification, are described in Table 35, Table 36 and Table 37.

Table 35 – Simple diagnostics data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
diagnosticsData	ARRAY OF WORD	Required data, depending on the sub-function type	Data
diagnosticsSubFct	UINT	Sub-function code	Sub-function

Table 36 – Structured DiagnosticsReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
DiagnosticsReq	STRUCT			Communication request equivalent to the Diagnostics service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	diagnosticsData	M	[1..1]	
	diagnosticsSubFct	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 37 – Structured DiagnosticsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
DiagnosticsRsp	STRUCT			Communication response equivalent to the Diagnostics service response defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	diagnosticsData	M	[1..1]	
	diagnosticsSubFct	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.11 Get Comm event counter transaction service

The parameters of the transaction service, which is equivalent to the Get Comm Event Counter service defined in the MODBUS Application Protocol specification, are described in Table 38, Table 39 and Table 40.

Table 38 – Structured GetCommEventCounterReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventCounterReq	STRUCT			Communication request equivalent to the Get Comm Event Counter service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 39 – Simple GetCommEventCounterRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
commStatus	WORD	Two-byte status information. The status information will be 0xFFFF if a previously-issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000.	Status
eventCount	UINT	Number of events counted by the device	Event Count

Table 40 – Structured GetCommEventCounterRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventCounterRsp	STRUCT			Communication response equivalent to the Get Comm Event Counter service response defined in the MODBUS Application Protocol specification
	commStatus	M	[1..1]	
	communicationReference	M	[1..1]	
	eventCount	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.12 Get Comm event log transaction service

The parameters of the transaction service, which is equivalent to the Get Comm Event Log service defined in the MODBUS Application Protocol specification, are described in Table 41, Table 42 and Table 43.

Table 41 – Structured GetCommEventLogReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventLogReq	STRUCT			Communication request equivalent to the Get Comm Event Log service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 42 – Simple GetCommEventLogRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
commStatus	WORD	Two-byte status information. The status information will be 0xFFFF if a previously-issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000	Status
eventCount	UINT	Number of events counted by the device	Event Count
events	ARRAY OF BYTE	Each byte corresponds to the status of one MODBUS send or receive operation for the remote device	Events
messageCount	UINT	Quantity of messages processed by the remote device since its last restart, clear counters operation, or power-up	Message Count

Table 43 – Structured GetCommEventLogRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventCounter Rsp	STRUCT			Communication response equivalent to the Get Comm Event Log service response defined in the MODBUS Application Protocol specification
	commStatus	M	[1..1]	
	communicationReference	M	[1..1]	
	eventCount	M	[1..1]	
	events	O	[0..1]	
	messageCount	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.13 Write multiple coils transaction service

The parameters of the transaction service, which is equivalent to the Write Multiple Coils service defined in IEC 61158-5-15, are described in Table 44, Table 45 and Table 46.

Table 44 – Simple WriteMultipleCoilsReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
multipleCoilValues	STRING	ASCII string with each coil state coded in one character, where the first character in the string represents the first coil to be written: – “0”=FALSE or “OFF” – “1”=TRUE or “ON”	Data
outputAddress	UINT	Address of the first coil to be forced	Address of first coil

Table 45 – Structured WriteMultipleCoilsReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleCoilsReq	STRUCT			Communication request equivalent to the Write Multiple Coils service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	multipleCoilValues	M	[1..1]	
	outputAddress	M	[1..1]	

Table 46 – Structured WriteMultipleCoilsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleCoilsRsp	STRUCT			Communication response equivalent to the Write Multiple Coils service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.14 Write multiple registers transaction service

The parameters of the transaction service, which is equivalent to the Write Multiple Holding Registers service defined in IEC 61158-5-15, are described in Table 47, Table 48 and Table 49.

Table 47 – Simple WriteMultipleRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
outputAddress	UINT	Address of the first holding register to be written	Address of first holding register to write
registerValues	ARRAY OF WORD	Register values to be written	Data

Table 48 – Structured WriteMultipleRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleRegisters Req	STRUCT			Communication request equivalent to the Write Multiple Holding Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	registerValues	M	[1..1]	

Table 49 – Structured WriteMultipleRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleRegisters Rsp	STRUCT			Communication response equivalent to the Write Multiple Holding Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.15 Report slave ID transaction service

The parameters of the transaction service, which is equivalent to the Report Slave ID service defined in the MODBUS Application Protocol specification, are described in Table 50, Table 51 and Table 52.

Table 50 – Structured ReportSlaveIDReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReportSlaveIDReq	STRUCT			Communication request equivalent to the Report Slave ID service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 51 – Simple ReportSlaveIDRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
data	ARRAY OF BYTE	This attribute contains the: - Slave ID, - the Run Indicator Status (0x00 or 0xFF) - and the additional device specific data in the same format and order as defined in the MODBUS Application Protocol Specification	Slave ID, Run Indicator Status Additional data

Table 52 – Structured ReportSlaveIDRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReportSlaveIDRsp	STRUCT			Communication response equivalent to the Report Slave ID service response defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	data	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.16 Read file record transaction service

The parameters of the transaction service, which is equivalent to the Read File Record service defined in IEC 61158-5-15, are described in Table 53 to Table 58.

Table 53 – Simple ReadFileSubRequest data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fileNumber	UINT	File number	File number
quantity	UINT	Length of the record to be read as quantity of registers	Record length
recordNumber	UINT	Starting record number within the file	Record number
referenceType	USINT	Reference type	Reference type
NOTE According to IEC 61158-5-15 the reference type shall be specified as 0x06.			

Table 54 – Structured ReadFileSubRequest data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileSubRequest	STRUCT			Sub-request equivalent to the Read File Record sub-request element defined in IEC 61158-5-15
	fileNumber	M	[1..1]	
	quantity	M	[1..1]	
	recordNumber	M	[1..1]	
	referenceType	M	[1..1]	

Table 55 – Structured ReadFileRecordReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileRecordReq	STRUCT			Communication request equivalent to the Read File Record service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	ReadFileSubRequest	M	[1..*]	

Table 56 – Simple ReadFileSubResponse data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
recordData	ARRAY OF WORD	Record data	Record data array

Table 57 – Structured ReadFileSubResponse data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileSubResponse	STRUCT			Sub-response equivalent to the Read File Record sub-response element defined in IEC 61158-5-15
	recordData	M	[1..1]	

Table 58 – Structured ReadFileRecordRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileRecordRsp	STRUCT			Communication response equivalent to the Read File Record service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	ReadFileSubResponse	M	[1..*]	

10.3.17 Write file record transaction service

The parameters of the transaction service, which is equivalent to the Write File Record service defined in IEC 61158-5-15, are described in Table 59 to Table 62.

Table 59 – Simple WriteFileSubRequest data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fileNumber	UINT	File number	File number
recordData	ARRAY OF WORD	Length of the record to be read as quantity of registers	Record data array
recordNumber	UINT	Starting record number within the file	Record number
referenceType	USINT	Reference type	Reference type
NOTE According to IEC 61158-5-15 the reference type shall be specified as 0x06.			

Table 60 – Structured WriteFileSubRequest data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteFileSubRequest	STRUCT			Sub-request equivalent to the Write File Record sub-request element defined in IEC 61158-5-15
	fileNumber	M	[1..1]	
	recordData	M	[1..1]	
	RecordNumber	M	[1..1]	
	ReferenceType	M	[1..1]	

Table 61 – Structured WriteFileRecordReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteFileRecordReq	STRUCT			Communication request equivalent to the Write File Record service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	WriteFileSubRequest	M	[1..*]	

Table 62 – Structured WriteFileRecordRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteFileRecordRsp	STRUCT			Communication response equivalent to the Write File Record service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.18 Mask write register transaction service

The parameters of the transaction service, which is equivalent to the Mask Write Holding Register service defined in IEC 61158-5-15, are described in Table 63, Table 64 and Table 65.

Table 63 – Simple MaskWriteRegisterReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
andMask	WORD	AND mask	AND Mask
orMask	WORD	OR mask	OR Mask
referenceAddress	UINT	Address of the holding register to which the mask shall be applied	Address of first holding register to write

Table 64 – Structured MaskWriteRegisterReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
MaskWriteRegisterReq	STRUCT			Communication request equivalent to the Mask Write Holding Register service request defined in IEC 61158-5-15
	andMask	M	[1..1]	
	communicationReference	M	[1..1]	
	orMask	M	[1..1]	
	referenceAddress	M	[1..1]	

Table 65 – Structured MaskWriteRegisterRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
MaskWriteRegisterRsp	STRUCT			Communication response equivalent to the Mask Write Holding Register service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.19 Read/write holding registers transaction service

The parameters of the transaction service, which is equivalent to the Read/Write Holding Registers service defined in IEC 61158-5-15, are described in Table 66, Table 67, Table 68 and Table 69.

Table 66 – Simple ReadWriteRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
readQuantity	UINT	Number of holding registers to be read	Quantity of holding registers to read
readStartAddress	UINT	Address of the first holding register to be read	Address of first holding register to read
writeRegisterValues	ARRAY OF WORD	Register values to be written	Data
writeStartAddress	UINT	Address of the first holding register to be written	Address of first holding register to write

Table 67 – Structured ReadWriteRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadWriteRegistersReq	STRUCT			Communication request equivalent to the Read/Write Holding Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	readQuantity	M	[1..1]	
	readStartAddress	M	[1..1]	
	writeRegisterValues	M	[1..1]	
	writeStartAddress	M	[1..1]	

Table 68 – Simple ReadWriteRegistersRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
readRegisterValues	ARRAY OF WORD	Read holding register values	Data

Table 69 – Structured ReadWriteRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadWriteRegistersRsp	STRUCT			Communication response equivalent to the Read/Write Holding Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	readRegisterValues	M	[1..1]	

10.3.20 Read FIFO queue transaction service

The parameters of the transaction service, which is equivalent to the Read FIFO service defined in IEC 61158-5-15, are described in Table 70, Table 71, Table 72 and Table 73.

Table 70 – Simple ReadFifoQueueReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fifoPointerAddress	UINT	Address of the first register to be read in a First-In-First-Out (FIFO) queue of registers	Address of FIFO queue

Table 71 – Structured ReadFifoQueueReq data types

Data type	Definition	Description
-----------	------------	-------------

	Elementary data types	U s a g e	Multiplicity	
ReadFifoQueueReq	STRUCT			Communication request equivalent to the Read FIFO service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	fifoPointerAddress	M	[1..1]	

Table 72 – Simple ReadFifoQueueRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fifoRegisterValues	ARRAY OF WORD	Register values read from the FIFO queue	Data

Table 73 – Structured ReadFifoQueueRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFifoQueueRsp	STRUCT			Communication response equivalent to the Read FIFO service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	fifoRegisterValues	M	[1..1]	

10.3.21 Encapsulated interface transport transaction service

The parameters of the transaction service, which is equivalent to the service provided by the Encapsulated Interface defined in IEC 61158-5-15, are described in Table 74, Table 75, Table 76 and Table 77.

Table 74 – Simple EncapsulatedInterfaceTransportReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
meiData	ARRAY OF BYTE	Modbus encapsulated interface type specific data	MEI type specific data
meiType	USINT	Modbus encapsulated interface type (MEI type)	MEI type

Table 75 – Structured EncapsulatedInterfaceTransportReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
EncapsulatedInterfaceTransportReq	STRUCT			Communication request equivalent to the service request provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	meiData	M	[1..1]	
	meiType	M	[1..1]	

Table 76 – Simple EncapsulatedInterfaceTransportRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
meiData	ARRAY OF BYTE	Modbus encapsulated interface type specific data	MEI type specific data
meiType	USINT	Modbus encapsulated interface type (MEI type)	MEI type

Table 77 – Structured EncapsulatedInterfaceTransportRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
EncapsulatedInterfaceTransportRsp	STRUCT			Communication response equivalent to the service response provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	meiData	M	[1..1]	
	meiType	M	[1..1]	

10.3.22 Read device identification transaction service

The parameters of the transaction service, which is equivalent to the Read Device Identification service provided by the Encapsulated Interface defined in IEC 61158-5-15, are described in Table 78 to Table 83.

Table 78 – Simple ReadDeviceIdentificationReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
readDeviceIdCode	USINT	The parameter " Read Device ID code " allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code
objectId	USINT	Identification of the first object to obtain	Requested object ID

Table 79 – Structured ReadDeviceIdentificationReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDeviceIdentificationReq	STRUCT			Communication request equivalent to the Read Device Identification service request provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	readDeviceIdCode	M	[1..1]	
	objectId	M	[1..1]	

Table 80 – Simple IdentificationObject data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
objectId	USINT	Identification of the returned object	Returned object ID
objectValue	ARRAY OF BYTES	ASCII string representing the object value	Object value

Table 81 – Structured IdentificationObject data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdentificationObject	STRUCT			Identification objects of the Read Device Identification service response provided by the Encapsulated Interface defined in IEC 61158-5-15
	objectId	M	[1..1]	
	objectValue	M	[1..1]	

Table 82 – Simple ReadDeviceIdentificationRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
conformityLevel	BYTE	Identification conformity level of the device and type of supported access 01 : basic identification (stream access only) 02 : regular identification (stream access only) 03 : extended identification (stream access only) 81 : basic identification (stream access and individual access) 82 : regular identification (stream access and individual access) 83 : extended identification (stream access and individual access)	Conformity level
moreFollows	BOOL	In case of readDeviceIdCode 01, 02 or 03 (stream access), if the identification data does not fit into a single response and several request/response transactions may be required: – “0” : no more objectst are available – “1” : other identification objects are available and further Modbus transactions are required In case of ReadDevId code 04 (individual access), this field shall be set to “0”	More-available flag
nextObjectId	USINT	– “0”, if no more identification objects are available (moreFollows=“0”) – Identification of the next object to be obtained, if more identification objects are available (moreFollows=“1”)	Next object ID
numberOfObjects	USINT	Number of identification objects returned in this response (for an individual access, numberOfObjects=“1”)	Number of objects
readDeviceIdCode	USINT	The parameter " Read Device ID code " allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code

Table 83 – Structured ReadDeviceIdentificationRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDeviceIdentificationRsp	STRUCT			Communication response equivalent to the Read Device Identification service response provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	conformityLevel	M	[1..1]	
	IdentificationObject	M	[1..*]	
	moreFollows	M	[1..1]	
	nextObjectId	M	[1..1]	
	numberOfObjects	M	[1..1]	
	readDeviceIdCode	M	[1..1]	

10.3.23 Private Modbus transaction service

This transaction service can be used in order to send any private Modbus request. The parameters of this transaction service are described in Table 84, Table 85, Table 86 and Table 87.

Table 84 – Simple PrivateModbusReq data types

Data type	Definition	Description
privateRequest	ARRAY OF BYTE	Sequence of hexadecimal digits representing the private Modbus request

Table 85 – Structured PrivateModbusReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
PrivateModbusReq	STRUCT			Communication request of the private Modbus transaction service
	communicationReference	M	[1..1]	
	privateRequest	M	[1..1]	

Table 86 – Simple PrivateModbusRsp data types

Data type	Definition	Description
privateResponse	ARRAY OF BYTE	Sequence of hexadecimal digits representing the private Modbus response

Table 87 – Structured PrivateModbusRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
PrivateModbusRsp	STRUCT			Communication response of the private Modbus transaction service
	communicationReference	M	[1..1]	
	privateResponse	M	[1..1]	

10.3.24 Unconfirmed private Modbus transaction service

This transaction service can be used in order to send any private unconfirmed Modbus request. The parameters of this transaction service are described in Table 88, Table 89 and Table 90. Although this service specifies an unconfirmed Modbus service, at the FDT level a transaction response shall be returned in order to inform the DTM that the request was sent. For more information about this behaviour see 6.3.

Table 88 – Simple UnconfirmedPrivateModbusReq data types

Data type	Definition	Description
privateRequest	ARRAY OF BYTE	Sequence of hexadecimal digits representing the private unconfirmed Modbus request

Table 89 – Structured UnconfirmedPrivateModbusReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UnconfirmedPrivateModbusReq	STRUCT			Communication request of the private unconfirmed Modbus transaction service
	communicationReference	M	[1..1]	
	privateRequest	M	[1..1]	

Table 90 – Structured UnconfirmedPrivateModbusRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UnconfirmedPrivateModbusRsp	STRUCT			Communication response of the private unconfirmed Modbus transaction service
	communicationReference	M	[1..1]	

10.3.25 Modbus exception response

The parameters of the transaction service response, which is equivalent to the exception response defined in IEC 61158-5-15, are described in Table 91 and Table 92.

Table 91 – Simple ModbusExceptionRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
modbusExceptionCode	BYTE	Modbus Exception Code	Exception code
modbusService	enumeration (ReadCoils ReadDiscreteInputs ReadHoldingRegisters ReadInputRegisters WriteSingleCoil WriteSingleRegister ReadExceptionStatus Diagnostics GetCommEventCounter GetCommEventLog WriteMultipleCoils WriteMultipleRegisters ReportSlaveID ReadFileRecord WriteFileRecord MaskWriteRegister ReadWriteRegisters ReadFifoQueue EncapsulatedInterface Transport ReadDeviceIdentification PrivateModbus)	Name of the Modbus service which failed: – ReadCoils – ReadDiscreteInputs – ReadHoldingRegisters – ReadInputRegisters – WriteSingleCoil – WriteSingleRegister – ReadExceptionStatus – Diagnostics: Diagnostics – GetCommEventCounter – GetCommEventLog – WriteMultipleCoils – WriteMultipleRegisters – ReportSlaveID – ReadFileRecord – WriteFileRecord – MaskWriteRegister – ReadWriteRegisters – ReadFifoQueue – EncapsulatedInterfaceTransport – ReadDeviceIdentification – PrivateModbus	Not defined

Table 92 – Structured ModbusExceptionRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ModbusExceptionRsp	STRUCT			Communication response of the private Modbus transaction service
	communicationReference	M	[1..1]	
	modbusExceptionCode	M	[1..1]	
	modbusService	M	[1..1]	

11 Channel parameter data types

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

- ReadChannelInformation;
- WriteChannelInformation.

A DTM can provide Process Channels, in order to expose detailed information to external systems such as controllers, HMI applications or to the Frame Application itself, about the process values available on the device. External systems can use this information to interpret and to directly access the runtime values in the device with standard protocol services.

This clause describes the structure and the semantic of the information which can be exposed by means of Process Channels provided by DTMs. It is up to a DTM whether it provides any channels. For a detailed description of the Process Channel parameters see Table 93 and Table 94.

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

Table 93 – Simple channel parameter data types

Data type	Definition	Description
address	UINT	Address of the discrete input, coil or register which shall be accessed. In case of a range of these Modbus data items shall be accessed, the address data type contains the address of the first Modbus data item within this range
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
iecDataType	enumeration (BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TimeOfDay DateAndTime STRING BYTE WORD DWORD LWORD WSTRING)	Identifier for IEC 61131-3 data type of the channel object
modbusDataTypes	enumeration (coil discreteInput holdingRegister inputRegister)	Identifier for the Modbus data item of the channel object
protectedByChannelAssignment	BOOL	– “0”, no restriction regarding the access on the parameter values – “1”, if the channel is set to read only by the Frame Application and neither the Frame Application nor the DTM itself are allowed to modify the parameter values. Usually set to “1” if a channel assignment exists
quantity	UINT	Number of discrete inputs, coils or registers which represent the channel object in the device

Data type	Definition	Description
scaleValue	STRING	ASCII coded REAL value with a dot as decimal separator. This value represents the scaling factor by which an actual value has to be multiplied to form a scaled value
statusChannel	BOOL	TRUE if the channel is for status information only

Table 94 – Structured channel parameter data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
FDTChannel	STRUCT			Description of the channel
	applId:applicationId	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:dataType	M	[1..1]	
	fdt:Deadband	O	[0..1]	
	fdt:descriptor	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:id	M	[1..1]	
	fdt:Ranges	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:signalType	M	[1..1]	
	fdt:StructuredElements	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:tag	M	[1..1]	
	fdt:Unit	O	[0..1]	
	frameApplicationTag	O	[0..1]	
	iecDataType	O	[0..1]	
	ModbusAccessData	O	[0..1]	
	protectedByChannel Assignment	M	[1..1]	
	UnitScaling	O	[0..1]	
FDTChannelType	STRUCT			Description of the channel component in case of channels with gateway functionality
	fdt:VersionInformation	M	[1..1]	
ModbusAccessData	STRUCT			Collection of required information to access the process value with standard Modbus services for reading or writing access
	address	M	[1..1]	
	modbusDataTypes	M	[1..1]	
	quantity	M	[1..1]	
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UnitScaling	STRUCT			Scaling information for the actual value of the parameter provided by the channel object
	scaleValue	M	[1..1]	

12 Device Identification

The following clauses describe what kind of device identification information has to be provided by a device and how this information is interpreted in FDT in order to explicitly identify Modbus devices in FDT.

12.1 Common device type identification data types

This subclause describes the information, which is used for the protocol specific device identification. For a detailed description of the common device identification parameters see Table 95.

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

Table 95 – Simple common identification data types

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	Defines the support level of a DTMDerivativeType for a physical device. This data type can be used by a Frame Application to display the support level of a DTM to the user. Users may use this information for an assignment decision. - genericSupport: DTMDerivativeType applies to all kind of physical devices of the corresponding fieldbus protocol; - profileSupport: DTMDerivativeType applies to physical devices of a certain profile of the fieldbus protocol; - blockspecificProfileSupport DTMDerivativeType applies to blocks included in physical device types (e.g.: Pressure TransducerBlock in Profibus PA); - specificSupport (DTMDerivativeType is developed for this physical device type; - identSupport: The DTMDerivativeType is capable of identifying the physical device in a vendor specific manner and to propose a better DTMDerivativeType.
majorMinorRevision	STRING	Information of the protocol specific identification object MajorMinorRevision
match	STRING	Regular expression string, which shall match with a data type provided by scan result
name	enumeration (ProductName ModelName VendorURL UserApplicationName)	Name of the protocol specific identification object (Regular category)

Data type	Definition	Description
nomatch	STRING	Regular expression string, which shall not match with a data type provided by scan result
productCode	STRING	Information of the protocol specific identification object ProductCode
protocolName	STRING	Name of the supported communication protocol. This data type shall contain the information "protocol_Modbus"
protocolSpecificName	enumeration (ProductName ModelName VendorURL UserApplicationName)	Fieldbus protocol specific name of one single identification information
value	STRING	Single identification information
vendorName	STRING	Information of the protocol specific identification object VendorName

12.2 Topology scan data types

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

- Scan.

This subclause describes the device information, for one device in the list of scanned devices. For a detailed description of the device information parameters see Table 96 and Table 97.

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

Table 96 – Simple device information data types

Data type	Definition	Description
majorMinorRevision	STRING	Information of the protocol specific identification object MajorMinorRevision
name	enumeration (ProductName ModelName VendorURL UserApplicationName)	Name of the protocol specific identification object (Regular category)
productCode	STRING	Information of the protocol specific identification object ProductCode
protocolName	STRING	Name of the supported communication protocol. This data type shall contain the information "protocol_Modbus"
protocolSpecificName	enumeration (ProductName ModelName VendorURL UserApplicationName)	Fieldbus protocol specific name of one single identification information
value	STRING	Single identification information
vendorName	STRING	Information of the protocol specific identification object VendorName

Table 97 – Structured device information data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdManufacturer	STRUCT			Manufacturer identification
	vendorName	M	[1..1]	
IdSoftwareRevision	STRUCT			Tool relevant version of the physical device – Firmware Version
	majorMinorRevision	M	[1..1]	
IdTypeID	STRUCT			Device type identification information of a single device
	productCode	M	[1..1]	
IdValue	STRUCT			FDT name, protocol specific name and value for one identification parameter without semantic information for the Frame Application
	name	M	[1..1]	
	protocolSpecificName	M	[1..1]	
	value	M	[1..1]	
IdValues	STRUCT			Collection of multiple identification objects
	IdValue	O	[0..*]	
ModbusDevice	STRUCT			Device identification information
	schemaVersion	O	[0..1]	
	protocolName	M	[1..1]	
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	

12.3 Scan identification data types

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

- Scan.

This subclause describes the information, which is used for the protocol specific scan identification. For a detailed description of the scan identification parameters, see Table 98 and Table 99.

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

Table 98 – Simple scan identification data types

Data type	Definition	Description
configuredState	enumeration (configuredAndPhysically Available configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	Shall contain the value “notApplicable”
resultState	enumeration (provisional final error)	Current state of the transmitted scan information

Table 99 – Structured scan identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdAddress	STRUCT			Address information of the target device
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
IdBusProtocol	STRUCT			Information about the used protocol
	modbusident: protocolName	M	[1..1]	
IdManufacturer	STRUCT			Manufacturer identification
	modbusident:vendorName	M	[1..1]	
ScanIdentification	STRUCT			Scan identification information of a single device
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	
ScanIdentifications	STRUCT			Collection of multiple device scan information
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	
IdSoftwareRevision	STRUCT			Tool relevant version of the physical device – Firmware Version
	modbusident: majorMinorRevision	M	[1..1]	
IdTypeID	STRUCT			Device type identification information of a single device
	modbusident:productCode	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdValue	STRUCT			FDT name, protocol specific name and value for one identification parameter without semantic information for the Frame Application
	modbusident:name	M	[1..1]	
	modbusident:value	M	[1..1]	
	modbusident:protocolSpecificName	M	[1..1]	
IdValues	STRUCT			Collection of multiple identification objects
	IdValue	O	[0..*]	

12.4 Device type identification data types – provided by DTM

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

- GetIdentificationInformation.

This subclause describes the information, which is used for the protocol specific device type identification. For a detailed description of the device type identification parameters, see Table 100.

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

Table 100 – Structured device type identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
DeviceIdentification	STRUCT			Single device identification information
	IdBusProtocol	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	
	modbusident:idDTMSupportLevel	M	[1..1]	
DeviceIdentifications	STRUCT			Collection of multiple device identification information
	DeviceIdentification	M	[1..*]	
IdBusProtocol	STRUCT			Information about the used protocol
	modbusident:protocolName	M	[1..1]	
IdManufacturer	STRUCT			Manufacturer identification
	modbusident:RegExpr	O	[0..*]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	modbusident:vendorName	M	[1..1]	
IdSoftwareRevision	STRUCT			Tool relevant version of the physical device – Firmware Version
	modbusident: majorMinorRevision	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
IdTypeID	STRUCT			Device type identification information of a single device
	modbusident:productCode	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
IdValue	STRUCT			FDT name, protocol specific name and value for one identification parameter without semantic information for the Frame Application
	modbusident: protocolSpecificName	M	[1..1]	
	modbusident:name	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
	modbusident:value	O	[0..1]	
IdValues	STRUCT			Collection of multiple identification objects
	IdValue	O	[0..*]	

12.5 Mapping of protocol specific device identification objects to FDT data types

The protocol specific device identification in FDT is based on the information, which can be retrieved from a device by the protocol specific Read Device Identification service (0x2B/0xE). For this reason, it is required for Modbus devices to support the protocol specific Read Device Identification service for all objects of the Basic Device Identification category. In order to clearly identify a device in FDT, it is recommended for Modbus devices also to support the protocol specific Read Device Identification service for the objects of the Regular Device Identification category.

The following description how the protocol specific device identification information is mapped to FDT data types applies to

- the FDT Device Type Identification (explained in 12.4),
- and to the FDT Scan (explained in 12.2).

All information provided by the protocol specific identification objects of the Basic Device Identification category shall be mapped to the specific data types as described in Table 101. All Information provided by the protocol specific identification objects of the Regular Device Identification category shall be mapped to the value data type of the structured data type IdValue, together with the additional information described in Table 102.

Table 101 – Mapping of protocol specific identification objects in FDT

Protocol specific device information				Protocol specific FDT information		
Object ID	Object name	Requirement for Modbus	Category	Semantics data type	Data type	Requirement for FDT
0x00	VendorName	Mandatory	Basic	IdManufacturer	vendorName	Mandatory
0x01	ProductCode	Mandatory		IdTypeID	productCode	Mandatory
0x02	MajorMinorRevision	Mandatory		IdSoftwareRevision	majorMinorRevision	Mandatory
0x03	VendorUrl	Optional	Regular	IdValue	value	Recommended
0x04	ProductName	Optional		IdValue	value	Recommended
0x05	ModelName	Optional		IdValue	value	Recommended
0x06	UserApplicationName	Optional		IdValue	value	Recommended

Table 102 – Additional information for optional identification objects

Protocol specific device information		Protocol specific FDT information		
Object ID	Object name	Semantics element	Name	protocolSpecificName
0x03	VendorUrl	IdValue	VendorURL	VendorURL
0x04	ProductName	IdValue	ProductName	ProductName
0x05	ModelName	IdValue	ModelName	ModelName
0x06	UserApplicationName	IdValue	UserApplicationName	UserApplicationName

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

Bibliography

- [1] *MODBUS Application Protocol Specification V1.1a*, Modbus IDA, June 4, 2004
 - [2] *MODBUS over Serial Line Specification & Implementation guide V1.0*, Modbus IDA, 12 December 2002
 - [3] *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0a*, Modbus IDA, June 4, 2004
 - [4] IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*
 - [5] ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*
-

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

SOMMAIRE

AVANT-PROPOS	61
INTRODUCTION.....	63
1 Domaine d'application	65
2 Références normatives	65
3 Termes, définitions, symboles, abréviations et conventions	66
3.1 Termes et définitions	66
3.2 Abréviations	66
3.3 Conventions	66
3.3.1 Dénominations des types de données et références aux types de données.....	66
3.3.2 Vocabulaire relatif aux exigences	66
4 Catégorie de bus	66
5 Accès à l'instance et aux données du dispositif	68
5.1 Objets de la voie de processus fournis par le DTM	68
5.2 Services du DTM pour accéder à l'instance et aux données du dispositif	68
6 Comportement spécifique à un protocole.....	69
6.1 Généralités.....	69
6.2 Diffusion.....	69
6.3 Demande privée non confirmée à Modbus	73
7 Utilisation spécifique à un protocole des types de données généraux	76
8 Types de données communs spécifiques à un protocole.....	77
8.1 Généralités.....	77
8.2 Information d'adresse	77
9 Types de données pour la gestion du réseau.....	78
10 Types de données de communication.....	79
10.1 Généralités.....	79
10.2 Types de données pour la gestion de la connexion.....	79
10.3 Types de données spécifiques au service transaction.....	80
10.3.1 Généralités	80
10.3.2 Adressage des éléments de données	80
10.3.3 Service transaction Read coils	81
10.3.4 Service transaction Read discrete inputs	82
10.3.5 Service transaction Read holding registers	84
10.3.6 Service transaction Read input registers.....	85
10.3.7 Service transaction Write single coil	86
10.3.8 Service transaction Write single register.....	87
10.3.9 Service transaction Read exception status.....	88
10.3.10 Service transaction Diagnostics.....	89
10.3.11 Service transaction Get Comm event counter	90
10.3.12 Service transaction Get Comm event log	91
10.3.13 Service transaction Write multiple coils.....	92
10.3.14 Service transaction Write multiple registers	93
10.3.15 Service transaction Report slave ID.....	94
10.3.16 Service transaction Read file record	96
10.3.17 Service transaction Write file record	97

10.3.18	Service transaction Mask write register.....	99
10.3.19	Service transaction Read/write holding registers.....	100
10.3.20	Service transaction Read FIFO queue.....	101
10.3.21	Service transaction Encapsulated interface transport.....	102
10.3.22	Service transaction Read device identification.....	103
10.3.23	Service transaction Private Modbus.....	107
10.3.24	Service transaction Unconfirmed private Modbus.....	108
10.3.25	Réponse d'exception de Modbus.....	109
11	Types de données pour les paramètres des voies.....	110
12	Identification du dispositif.....	113
12.1	Types de données pour l'identification commune d'un type de dispositif.....	113
12.2	Types de données pour le balayage topologique.....	114
12.3	Types de données pour l'identification lors du balayage.....	115
12.4	Types de données pour l'identification du type de dispositif – fourni par le DTM.....	117
12.5	Mise en correspondance des objets d'identification du dispositif spécifique à un protocole avec les types de données du FDT.....	119
	Bibliographie.....	120
	Figure 1 – Partie 315 de la série IEC 62453.....	64
	Figure 2 – Séquence de diffusion avec un DTM de Communication Modbus Serial Line.....	71
	Figure 3 – Séquence de diffusion avec un DTM de Communication Modbus TCP.....	72
	Figure 4 – Séquence de diffusion sans DTM de Communication Modbus.....	73
	Figure 5 – Demande non confirmée avec un DTM de Communication Modbus Serial Line.....	74
	Figure 6 – Demande non confirmée avec un DTM de Communication Modbus TCP.....	75
	Figure 7 – Demande non confirmée sans DTM de Communication Modbus.....	76
	Figure 8 – Adressage des éléments de données.....	81
	Tableau 1 – Identificateurs du protocole.....	66
	Tableau 2 – Information d'adresse pour le mode de diffusion.....	69
	Tableau 3 – Demandes de transaction en mode diffusion.....	70
	Tableau 4 – Utilisation spécifique à un protocole des types de données généraux de FDT.....	77
	Tableau 5 – Types de données simples pour l'information d'adresse.....	77
	Tableau 6 – Types de données structurés pour l'information d'adresse.....	78
	Tableau 7 – Types de données structurés pour la gestion du réseau.....	78
	Tableau 8 – Types de données communs de communication.....	79
	Tableau 9 – Types de données structurés pour la gestion du service connexion.....	80
	Tableau 10 – Types de données simples ReadCoilsReq.....	81
	Tableau 11 – Types de données structurés ReadCoilsReq.....	82
	Tableau 12 – Types de données simples ReadCoilsRsp.....	82
	Tableau 13 – Types de données structurés ReadCoilsRsp.....	82
	Tableau 14 – Types de données simples ReadDiscreteInputsReq.....	83
	Tableau 15 – Types de données structurés ReadDiscreteInputsReq.....	83
	Tableau 16 – Types de données simples ReadDiscreteInputsRsp.....	83

Tableau 17 – Types de données structurés ReadDiscreteInputsRsp.....	83
Tableau 18 – Types de données simples ReadHoldingRegistersReq	84
Tableau 19 – Types de données structurés ReadHoldingRegistersReq	84
Tableau 20 – Types de données simples ReadHoldingRegistersRsp	84
Tableau 21 – Types de données structurés ReadHoldingRegistersRsp	85
Tableau 22 – Types de données simples ReadInputRegistersReq.....	85
Tableau 23 – Types de données structurés ReadInputRegistersReq	85
Tableau 24 – Types de données simples ReadInputRegistersRsp	85
Tableau 25 – Types de données structurés ReadInputRegistersRsp	86
Tableau 26 – Types de données simples WriteSingleCoilReq	86
Tableau 27 – Types de données structurés WriteSingleCoilReq	86
Tableau 28 – Types de données structurés WriteSingleCoilRsp	87
Tableau 29 – Types de données simples WriteSingleRegisterReq.....	87
Tableau 30 – Types de données structurés WriteSingleRegisterReq	87
Tableau 31 – Types de données structurés WriteSingleRegisterRsp	88
Tableau 32 – Types de données structurés ReadExceptionStatusReq.....	88
Tableau 33 – Types de données simples ReadExceptionStatusRsp	88
Tableau 34 – Types de données structurés ReadExceptionStatusRsp.....	89
Tableau 35 – Types de données simples pour les diagnostics.....	89
Tableau 36 – Types de données structurés DiagnosticsReq.....	89
Tableau 37 – Types de données structurés DiagnosticsRsp	90
Tableau 38 – Types de données structurés GetCommEventCounterReq	90
Tableau 39 – Types de données simples GetCommEventCounterRsp	90
Tableau 40 – Types de données structurés GetCommEventCounterRsp	91
Tableau 41 – Types de données structurés GetCommEventLogReq.....	91
Tableau 42 – Types de données simples GetCommEventLogRsp	92
Tableau 43 – Types de données structurés GetCommEventLogRsp.....	92
Tableau 44 – Types de données simples WriteMultipleCoilsReq	93
Tableau 45 – Types de données structurés WriteMultipleCoilsReq.....	93
Tableau 46 – Types de données structurés WriteMultipleCoilsRsp	93
Tableau 47 – Types de données simples WriteMultipleRegistersReq.....	94
Tableau 48 – Types de données structurés WriteMultipleRegistersReq	94
Tableau 49 – Types de données structurés WriteMultipleRegistersRsp	94
Tableau 50 – Types de données structurés ReportSlaveIDReq	95
Tableau 51 – Types de données simples ReportSlaveIDRsp	95
Tableau 52 – Types de données structurés ReportSlaveIDRsp	95
Tableau 53 – Types de données simples ReadFileSubRequest.....	96
Tableau 54 – Types de données structurés ReadFileSubRequest	96
Tableau 55 – Types de données structurés ReadFileRecordReq.....	96
Tableau 56 – Types de données simples ReadFileSubResponse	97
Tableau 57 – Types de données structurés ReadFileSubResponse.....	97
Tableau 58 – Types de données structurés ReadFileRecordRsp	97
Tableau 59 – Types de données simples WriteFileSubRequest.....	98

Tableau 60 – Types de données structurés WriteFileSubRequest	98
Tableau 61 – Types de données structurés WriteFileRecordReq	98
Tableau 62 – Types de données structurés WriteFileRecordRsp	99
Tableau 63 – Types de données simples MaskWriteRegisterReq	99
Tableau 64 – Types de données structurés MaskWriteRegisterReq	99
Tableau 65 – Types de données structurés MaskWriteRegisterRsp	100
Tableau 66 – Types de données simples ReadWriteRegistersReq	100
Tableau 67 – Types de données structurés ReadWriteRegistersReq	100
Tableau 68 – Types de données simples ReadWriteRegistersRsp	101
Tableau 69 – Types de données structurés ReadWriteRegistersRsp	101
Tableau 70 – Types de données simples ReadFifoQueueReq	101
Tableau 71 – Types de données structurés ReadFifoQueueReq	101
Tableau 72 – Types de données simples ReadFifoQueueRsp	102
Tableau 73 – Types de données structurés ReadFifoQueueRsp	102
Tableau 74 – Types de données simples EncapsulatedInterfaceTransportReq	102
Tableau 75 – Types de données structurés EncapsulatedInterfaceTransportReq	103
Tableau 76 – Types de données simples EncapsulatedInterfaceTransportRsp	103
Tableau 77 – Types de données structurés EncapsulatedInterfaceTransportRsp	103
Tableau 78 – Types de données simples ReadDeviceIdentificationReq	104
Tableau 79 – Types de données structurés ReadDeviceIdentificationReq	104
Tableau 80 – Types de données simples IdentificationObject	104
Tableau 81 – Types de données structurés IdentificationObject	105
Tableau 82 – Types de données simples ReadDeviceIdentificationRsp	106
Tableau 83 – Types de données structurés ReadDeviceIdentificationRsp	107
Tableau 84 – Types de données simples PrivateModbusReq	107
Tableau 85 – Types de données structurés PrivateModbusReq	107
Tableau 86 – Types de données simples PrivateModbusRsp	108
Tableau 87 – Types de données structurés PrivateModbusRsp	108
Tableau 88 – Types de données simples UnconfirmedPrivateModbusReq	108
Tableau 89 – Types de données structurés UnconfirmedPrivateModbusReq	108
Tableau 90 – Types de données structurés UnconfirmedPrivateModbusRsp	109
Tableau 91 – Types de données Simple ModbusExceptionRsp	109
Tableau 92 – Types de données structurés ModbusExceptionRsp	110
Tableau 93 – Types de données simples pour les paramètres des voies	111
Tableau 94 – Types de données structurés pour les paramètres des voies	112
Tableau 95 – Types de données simples pour l'identification commune	113
Tableau 96 – Types de données simples pour l'information relative au dispositif	114
Tableau 97 – Types de données structurés pour l'information relative au dispositif	115
Tableau 98 – Types de données simples pour l'identification lors du balayage	116
Tableau 99 – Types de données structurés pour l'identification lors du balayage	116
Tableau 100 – Types de données pour l'identification du type de dispositif	118
Tableau 101 – Mise en correspondance des objets d'identification spécifique à un protocole dans le FDT	119

Tableau 102 – Informations supplémentaires pour les objets d'identification facultatifs.....	119
Tableau 103 – Identificateurs de couche physique pour le protocole Modbus TCP.....	67
Tableau 104 – Identificateurs de couche physique pour le protocole Modbus Serial	68

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

Partie 315: Intégration des profils de communication – IEC 61784 CPF 15

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(s) amendement(s) doivent être considérées comme les documents officiels.

Cette version consolidée de l'IEC 62453-315 porte le numéro d'édition 1.1. Elle comprend la première édition (2009-07) [documents 65E/131/FDIS et 65E/144/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 62453-315 a été établie par le sous-comité 65 E: 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.

Chacune des parties de la série IEC 62453-3xy est destinée à être lue conjointement avec l'IEC 62453-2.

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

Cette publication a été rédigée conformément aux 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)*, est disponible 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 "colour inside" 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.

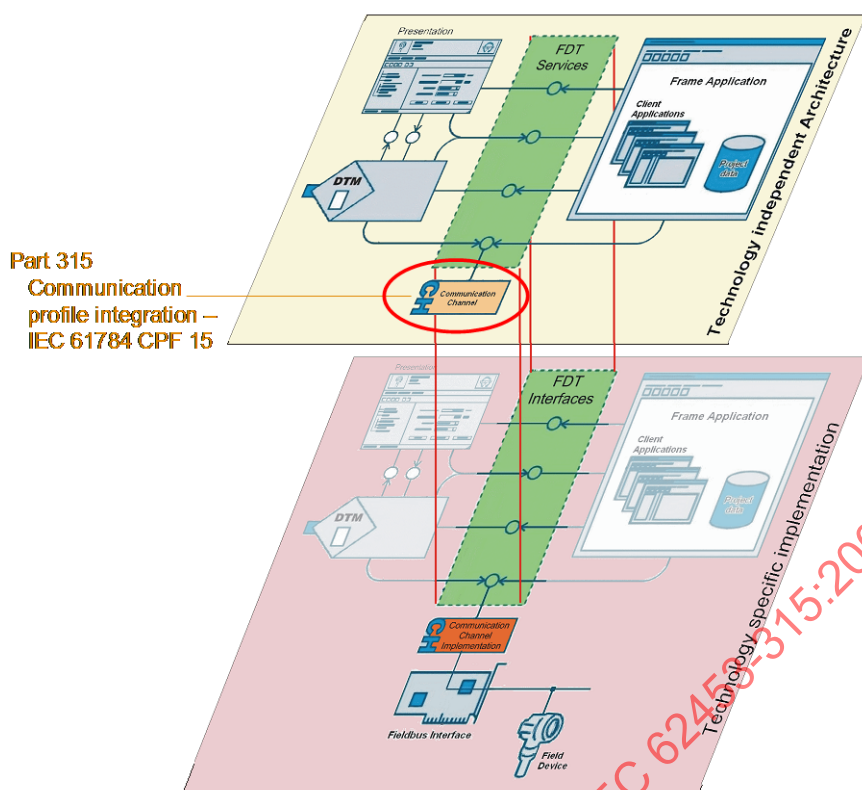
INTRODUCTION

Cette partie de l'IEC 62453 désigne une spécification de l'interface pour les développeurs des composants de FDT (Field Device Tool) (Outil pour Dispositifs de Terrain) pour la commande des fonctions et l'accès aux données au sein d'une architecture client/serveur. La spécification résulte d'une analyse et d'un processus de conception destiné à développer des interfaces normalisées afin de faciliter le développement de serveurs et de clients par de multiples vendeurs ayant besoin d'interagir sans couture.

Des bus de terrain étant intégrés aux systèmes de commande, quelques tâches supplémentaires doivent être effectuées. En plus des outils relatifs aux dispositifs ainsi qu'aux bus de terrain, il est nécessaire d'intégrer ces outils à des outils de planification à l'échelle du système à un niveau plus élevé ou à des outils d'études. En particulier, pour des utilisations dans des systèmes de commande vastes et hétérogènes, généralement dans le secteur de l'industrie de transformation, il est très important de définir clairement les interfaces d'ingénierie faciles d'utilisation pour toutes celles concernées.

Un composant logiciel spécifique à un dispositif appelé DTM (Device Type Manager) (Gestionnaire de type de dispositifs) est fourni par le fabricant du dispositif de terrain avec le dispositif auquel il est associé. Le DTM est intégré aux outils d'ingénierie via les interfaces du FDT définies dans la présente spécification. L'approche de l'intégration est généralement ouverte à tous les types de bus de terrain et par conséquent satisfait aux exigences relatives à l'intégration de différents types de dispositifs dans des systèmes de commande hétérogènes.

La Figure 1 présente la manière dont l'IEC 62453-315 est alignée dans la structure de la série IEC 62453.



IEC 1136/09

Légende

Anglais	Français
Part 315 Communication profile integration – IEC 61784 CPF 15	Partie 315 Intégration des profils de communication – IEC 61784 CPF 15
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	Service de FDT
FDT Interfaces	Interfaces de FDT
Fieldbus interface	Interface de bus de terrain

Figure 1 – Partie 315 de la série IEC 62453

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

Partie 315: Intégration des profils de communication – IEC 61784 CPF 15

1 Domaine d'application

La Famille de Profils de Communication 15 (généralement connue sous le nom de Modbus¹) définit les profils de communication basés sur l'IEC 61158-5-15 et l'IEC 61158-6-15. Le profil de base CP 15/1 (Modbus TCP) est défini dans l'IEC 61784-1. Un autre profil de communication (Modbus Serial Line) est défini en [2].

Cette partie de l'IEC 62453 fournit des informations pour intégrer le protocole Modbus TCP® et Modbus Serial Line® dans des systèmes basés sur FDT.

NOTE Cette partie de l'IEC 62453 ne spécifie que la mise en correspondance des paramètres Modbus avec les types de données de FDT. Pour connaître les restrictions des paramètres spécifiques à un protocole concernant les valeurs autorisées ainsi que les limites des matrices utilisées dans la définition des types de données de FDT, se référer à l'IEC 61158-5-15 ainsi qu'à la Spécification du Protocole pour l'Application de MODBUS.

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, *Automates programmables – Partie 3: Langages de programmation*

IEC 61158-5-15, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-15: Définition des services des couches d'application – Éléments de type 15*

IEC 61158-6-15, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-20: Spécification des services des couches d'application – Éléments de type 15*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles* (disponible en anglais uniquement)

IEC 61784-2, *Industrial communication networks – Profiles – Part -2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3* (disponible en anglais uniquement)

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

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

1) Modbus est la marque commerciale de Schneider Automation Inc. Elle est déposée aux Etats-Unis d'Amérique. 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 avec ce profil ne nécessite pas l'utilisation de la marque commerciale déposée. L'utilisation de la marque commerciale Modbus requiert l'autorisation de Schneider Automation Inc.

RFC 791, *Protocole Internet* (disponible sous <<http://www.ietf.org/rfc/rfc0791.txt>>)

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, l'IEC 62453-2 ainsi que les suivantes s'appliquent.

3.2 Abréviations

Pour les besoins du présent document, les abréviations données dans l'IEC 62453-1, l'IEC 62453-2 ainsi que les suivantes s'appliquent.

ASCII	American Standard Code for Information Interchange (Code Américain Normalisé pour l'Echange d'Informations)
DTM	Device Type Manager (Gestionnaire du type de dispositif)
FA	Frame Application (Application cadre)
IP	Internet Protocol (Protocole Internet)
RFC	Request For comment (Demande de commentaire)
TCP	Transmission Control Protocol (Protocole de commande de transport (protocoleTCP))

3.3 Conventions

3.3.1 Dénominations des types de données et références aux types de données

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

3.3.2 Vocabulaire relatif aux exigences

Les expressions suivantes sont utilisées lors de la spécification des exigences.

- Utilisation de "doit" ou de "obligatoire"
- Utilisation de "il convient de" ou de "recommandé"
- Utilisation de "peut" ou de "facultatif"
- Aucune exception autorisée.
- Recommandation forte. Il peut être légitime, dans des cas particuliers exceptionnels, de s'écarter du comportement décrit.
- La fonction ou le comportement peut être indiqué(e) selon les conditions définies.

4 Catégorie de bus

Le protocole Modbus est identifié dans l'élément protocolld du type de données structuré 'fdt:BusCategory' par les identificateurs uniques suivants (voir Tableau 1).

Tableau 1 – Identificateurs du protocole

Valeur de l'identificateur	Nom du Protocolld	Description
59629a40-285f-11db-a98b-0800200c9a66	'Modbus over Serial Line'	Modbus sur Ligne Sérielle
59629a41-285f-11db-a98b-0800200c9a66	'Modbus over TCP'	Modbus sur TCP

Le protocole Modbus TCP utilise les identificateurs uniques suivants dans les membres physicalLayer au sein du type de données PhysicalLayer (voir Tableau 103):

Tableau 103 – Identificateurs de couche physique pour le protocole Modbus TCP

Valeur d'identificateur	Description
307DD810-C010-11DB-90E7-0002B3ECDCE	100BaseTXFD (valeur par défaut pour Media Type Copper)
307DD812-C010-11DB-90E7-0002B3ECDCE	100BaseFXFD (valeur par défaut pour Media Type Fiber Optic)
307DD813-C010-11DB-90E7-0002B3ECDCE	100BaseLX10
307DD816-C010-11DB-90E7-0002B3ECDCE	1000BaseXFD
307DD818-C010-11DB-90E7-0002B3ECDCE	1000BaseLXFD
307DD81A-C010-11DB-90E7-0002B3ECDCE	1000BaseSXFD
307DD81C-C010-11DB-90E7-0002B3ECDCE	1000BaseTFD
307DD81D-C010-11DB-90E7-0002B3ECDCE	10GbaseFX
307DD81E-C010-11DB-90E7-0002B3ECDCE	10GbaseLX4
307DD81F-C010-11DB-90E7-0002B3ECDCE	10GBaseR
307DD820-C010-11DB-90E7-0002B3ECDCE	10GbaseER
307DD821-C010-11DB-90E7-0002B3ECDCE	10GbaseLR
307DD822-C010-11DB-90E7-0002B3ECDCE	10GbaseSR
307DD823-C010-11DB-90E7-0002B3ECDCE	10GbaseW
307DD824-C010-11DB-90E7-0002B3ECDCE	10GbaseEW
307DD824-C010-11DB-90E7-0002B3ECDCE	10GbaseLW
307DD825-C010-11DB-90E7-0002B3ECDCE	10GbaseSW
307DD826-C010-11DB-90E7-0002B3ECDCE	10GbaseCX4
307DD827-C010-11DB-90E7-0002B3ECDCE	2BaseTL
307DD828-C010-11DB-90E7-0002B3ECDCE	10PassTS
307DD829-C010-11DB-90E7-0002B3ECDCE	100BaseBX10D
307DD82A-C010-11DB-90E7-0002B3ECDCE	100BaseBX10U
307DD82B-C010-11DB-90E7-0002B3ECDCE	100BaseLX10
307DD82C-C010-11DB-90E7-0002B3ECDCE	1000BaseBX10D
307DD82D-C010-11DB-90E7-0002B3ECDCE	1000BaseBX10U
307DD82F-C010-11DB-90E7-0002B3ECDCE	1000BaseLX10
307DD830-C010-11DB-90E7-0002B3ECDCE	1000BasePX10D
307DD831-C010-11DB-90E7-0002B3ECDCE	1000BasePX10U
307DD832-C010-11DB-90E7-0002B3ECDCE	1000BasePX20D
307DD833-C010-11DB-90E7-0002B3ECDCE	1000BasePX20U
307DD834-C010-11DB-90E7-0002B3ECDCE	10GBaseT ou 100BasePXFD
307DD835-C010-11DB-90E7-0002B3ECDCE	10GBaseLRM
307DD836-C010-11DB-90E7-0002B3ECDCE	1000BaseKX
307DD837-C010-11DB-90E7-0002B3ECDCE	1000BaseKX4
307DD838-C010-11DB-90E7-0002B3ECDCE	1000BaseKR
307DD839-C010-11DB-90E7-0002B3ECDCE	10G1GBasePRXD1
307DD83A-C010-11DB-90E7-0002B3ECDCE	10G1GBasePXR2
307DD83B-C010-11DB-90E7-0002B3ECDCE	10G1GBasePXR3
307DD83C-C010-11DB-90E7-0002B3ECDCE	10G1GBasePRXU1
307DD83D-C010-11DB-90E7-0002B3ECDCE	10G1GBasePXR2
307DD83E-C010-11DB-90E7-0002B3ECDCE	10G1GBasePXR3
307DD83F-C010-11DB-90E7-0002B3ECDCE	10GBasePRD1

Valeur d'identificateur	Description
307DD840-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD2
307DD841-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD3
307DD842-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU1
307DD843-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU3
307DD844-C010-11DB-90E7-0002B3ECDCBE	40GbaseKR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseCR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseSR4
307DD846-C010-11DB-90E7-0002B3ECDCBE	40GbaseFR
307DD847-C010-11DB-90E7-0002B3ECDCBE	40GbaseLR4
307DD848-C010-11DB-90E7-0002B3ECDCBE	100GbaseCR10
307DD849-C010-11DB-90E7-0002B3ECDCBE	100GbaseSR10
307DD84A-C010-11DB-90E7-0002B3ECDCBE	100GbaseLR4
307DD84B-C010-11DB-90E7-0002B3ECDCBE	100GbaseER4
307DD84C-C010-11DB-90E7-0002B3ECDCBE	100BasePXFD
307DD84D-C010-11DB-90E7-0002B3ECDCBE	Radiocommunication
307DD84E-C010-11DB-90E7-0002B3ECDCBE	Vitesse à 100 Mbit/s (et plus) et duplex intégral

Le protocole Modbus Serial utilise les identificateurs uniques suivants dans les membres physicalLayer au sein du type de données PhysicalLayer (voir Tableau 104):

Tableau 104 – Identificateurs de couche physique pour le protocole Modbus Serial

Valeur d'identificateur	Description
C0458028-F240-45A5-8664-70DC84FDC6FE	RS-232
3BF008DC-5A44-4220-8C3E-3C46A589A0B4	RS-422
036D1591-387B-11D4-86E1-00E0987270B9	RS-485

La propriété DataLinkLayer n'est pas applicable pour Modbus et doit être mise à "null".

5 Accès à l'instance et aux données du dispositif

5.1 Objets de la voie de processus fournis par le DTM

Pour Modbus, il n'existe pas d'ensemble minimal de paramètres défini devant être fourni par un DTM. Si un DTM fournit des valeurs de processus, il convient que les variables du processus soient modélisées comme les Voies de Processus.

5.2 Services du DTM pour accéder à l'instance et aux données du dispositif

Non applicable.

6 Comportement spécifique à un protocole

6.1 Généralités

Cet article spécifie comment traiter dans le FDT les demandes Modbus non confirmées. Il existe deux types de demandes Modbus non confirmées prises en charge dans le FDT:

- les diffusions;
- les demandes Modbus privées non confirmées.

6.2 Diffusion

En mode de diffusion, un DTM peut envoyer une demande Modbus à l'ensemble des dispositifs connectés au bus. Ce mode n'est pris en charge que par les dispositifs connectés par l'intermédiaire de Modbus Serial Line. Il peut s'agir d'une connexion directe ou d'une connexion par l'intermédiaire d'une passerelle. Le mode de diffusion doit être initié par une `fdtmodbus:ConnectRequest` avec l'adresse d'esclave du dispositif cible mise à 0 (`slaveAddress=0`). Pour plus de détails concernant l'information d'adresse à fournir pour le mode de diffusion, voir Tableau 2.

Tableau 2 – Information d'adresse pour le mode de diffusion

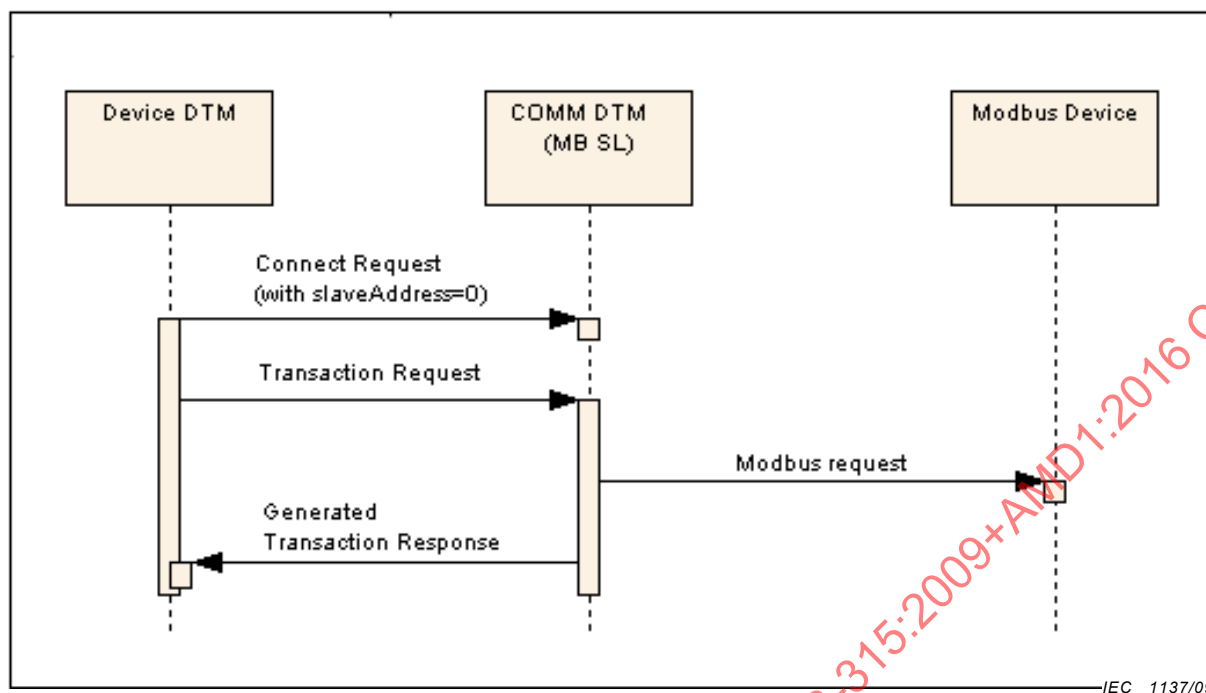
Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ModbusSerial	STRUCT			Information d'adresse pour Modbus Serial Line
	slaveAddress	M	[1..1]	Doit être 0 pour le mode de diffusion générale
ModbusTCP	STRUCT			Information d'adresse pour Modbus TCP:
	slaveAddress	M	[1..1]	Doit être 0 pour le mode de diffusion générale
	tcpAddress	M	[1..1]	Adresse IP de la passerelle Modbus TCP/ Modbus Serial Line
	tcpPort	O	[0..1]	Port pour la connexion Modbus TCP (si aucune information n'est fournie, le port par défaut 502 doit être utilisé)

Étant donné qu'aucune réponse n'est retournée par le dispositif en mode de diffusion, ce dernier ne doit être utilisé qu'avec les demandes de transaction présentées dans le Tableau 3:

Tableau 3 – Demandes de transaction en mode diffusion

Demandes de transaction en mode diffusion	Restrictions
WriteSingleCoilReq	Aucune
WriteSingleRegisterReq	Aucune
DiagnosticsReq	Cette demande de transaction doit être utilisée uniquement en mode diffusion avec les sous-fonctions suivantes: - 0x01: Restart Communication Option (Redémarrer l'option de communication) - 0x03: Change ASCII Input Delimiter (Modifier le délimiteur d'entrée ASCII) - 0x04: Force Listen Only Mode (Forcer le mode en écoute uniquement) - 0x0A: Clear Counters and Diagnostic Register (Effacer les compteurs et le registre des diagnostics) - 0x14: Clear Overrun Counter and Flag (Effacer le compteur de surcharge et le fanion)
WriteMultipleCoilsReq	Aucune
WriteMultipleRegistersReq	Aucune
WriteFileRecordReq	Aucune
MaskWriteRegisterReq	Aucune
PrivateModbusReq	Doit être utilisé uniquement avec des services privés où aucune réponse du dispositif n'est requise

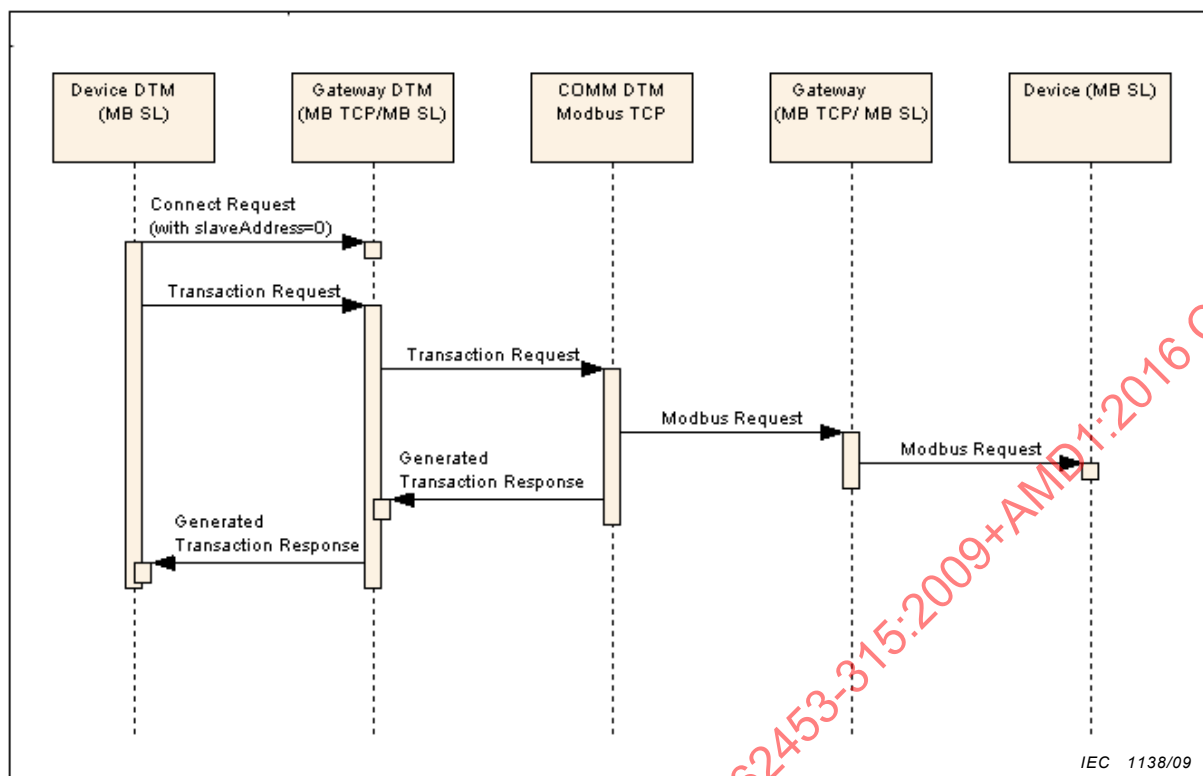
Bien qu'aucune réponse ne soit retournée par les dispositifs cibles en mode diffusion, une réponse de transaction correspondant à la demande de transaction doit être générée. Cette réponse de transaction doit être générée afin d'informer le DTM que la demande de diffusion a été envoyée sur le bus. Si un DTM de Communication Modbus est utilisé afin d'établir la communication, c'est lui qui fournit cette réponse de transaction générée (voir Figure 2 et Figure 3).



Légende

Anglais	Français
Device DTM	DTM du dispositif
COMM DTM	DTM de communication
Modbus Device	Dispositif Modbus
Connect request (with slave Address =0)	Demande de connexion (avec Adresse d'esclave = 0)
Transaction Request	Demande de transaction
Generated Transaction Response	Réponse de transaction générée
Modbus request	Demande Modbus

Figure 2 – Séquence de diffusion avec un DTM de Communication Modbus Serial Line

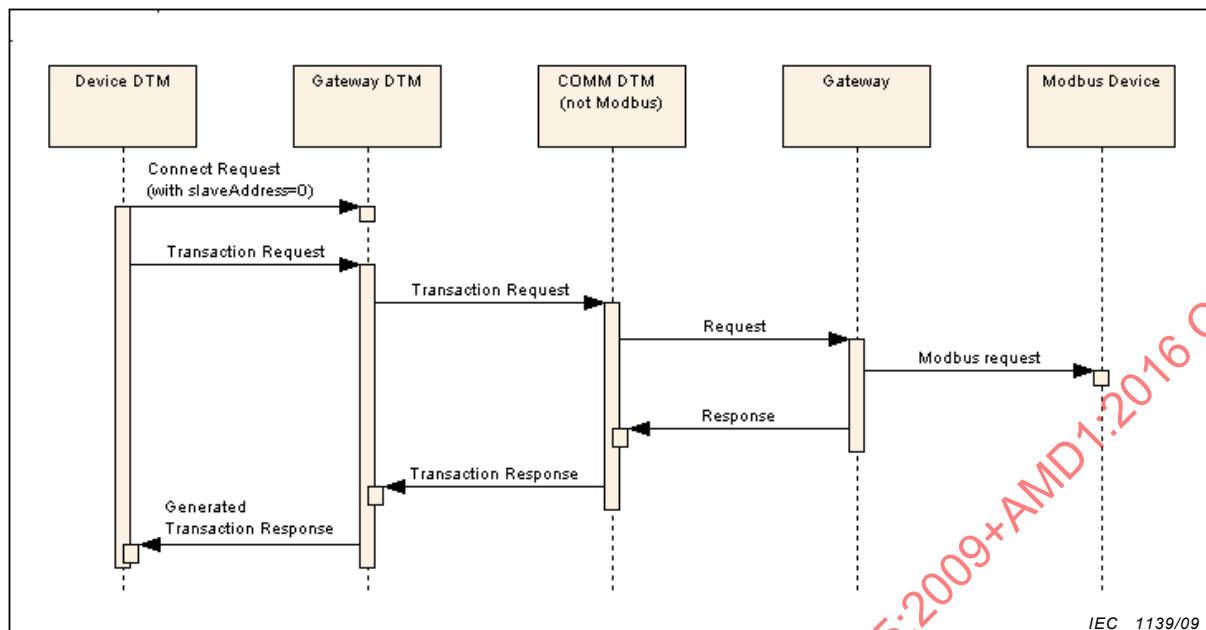


Légende

Anglais	Français
Device DTM	DTM de dispositif
Gateway DTM	DTM de passerelle
COMM DTM Modbus TCP	DTM de communication Modbus TCP
Gateway	Passerelle
Device MB SL	Dispositif MB SL
Connect request (with slave Address =0)	Demande de connexion (avec Adresse d'esclave = 0)
Transaction Request	Demande de transaction
Generated Transaction Response	Réponse de transaction générée
Modbus request	Demande Modbus

Figure 3 – Séquence de diffusion avec un DTM de Communication Modbus TCP

Si aucun DTM de Communication Modbus n'est utilisé pour établir la communication, le DTM de la passerelle Modbus, qui assure la liaison au niveau de communication le plus élevé différent de Modbus, doit fournir cette réponse de transaction (voir Figure 4).



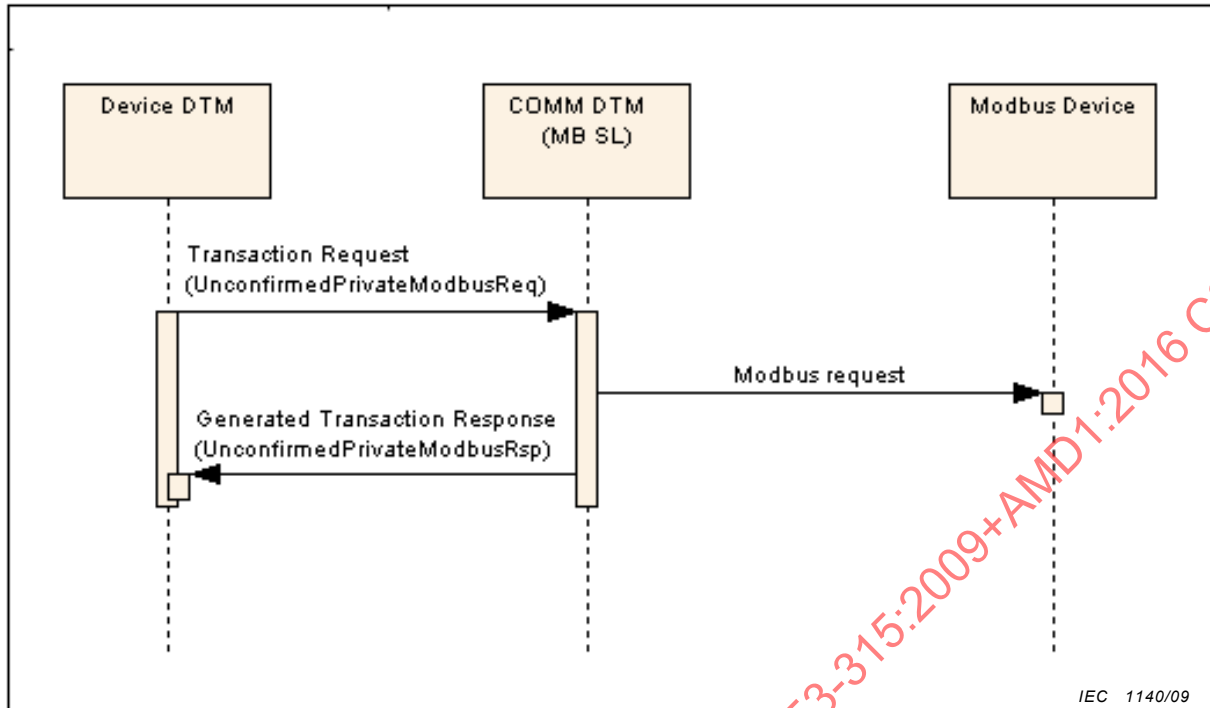
Légende

Anglais	Français
Device DTM	DTM de dispositif
Gateway DTM	DTM de passerelle
COMM DTM (not modbus)	DTM de communication (différent de Modbus)
Gateway	Passerelle
Modbus Device	Dispositif Modbus
Connect request (with slave Address =0)	Demande de connexion (avec Adresse d'esclave =0)
Transaction Request	Demande de transaction
Generated Transaction Response	Réponse de transaction générée
Modbus request	Demande Modbus

Figure 4 – Séquence de diffusion sans DTM de Communication Modbus

6.3 Demande privée non confirmée à Modbus

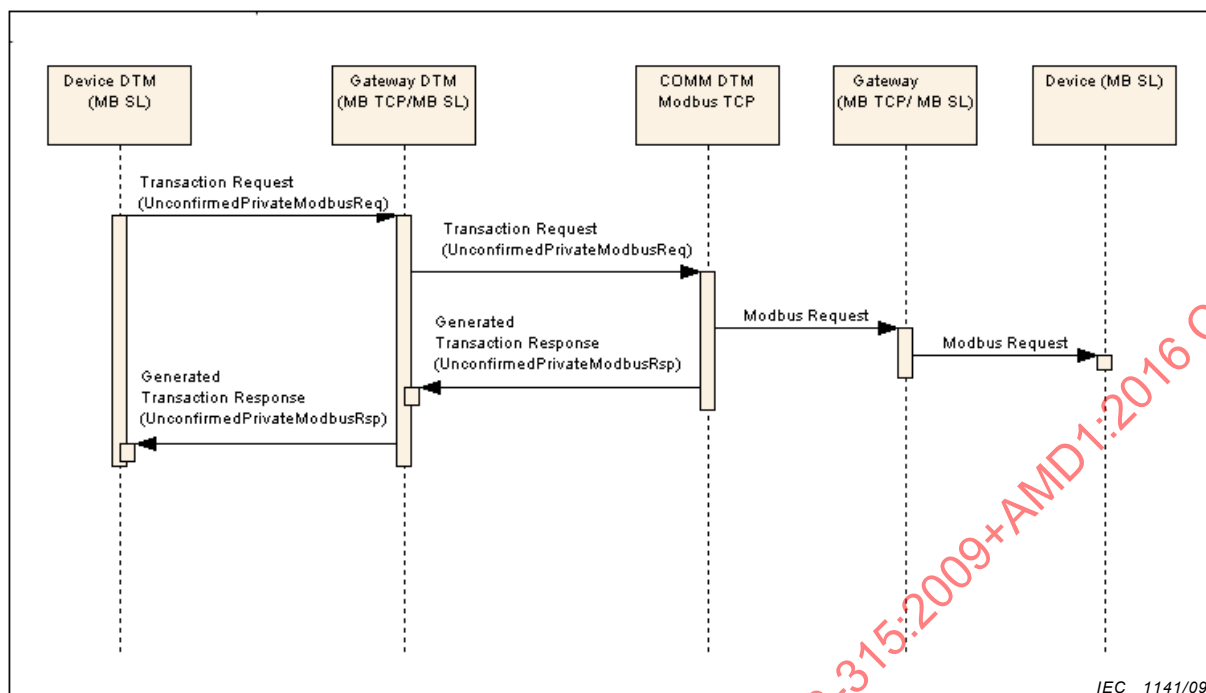
Cette demande de transaction peut être utilisée afin d'envoyer toute demande privée non confirmée à Modbus. Bien qu'aucune réponse ne soit retournée par le dispositif cible, une réponse de transaction correspondant à la demande de transaction doit être générée. Cette réponse de transaction doit être générée afin d'informer le DTM que la demande non confirmée a été envoyée sur le bus. Si un DTM de Communication Modbus est utilisé afin d'établir la communication, c'est lui qui doit fournir la réponse de transaction générée (voir Figure 5 et Figure 6).



Légende

Anglais	Français
Device DTM	DTM de dispositif
Gateway DTM	DTM de passerelle
COMM DTM Modbus TCP	DTM de communication Modbus TCP
Modbus Device	Dispositif Modbus
Transaction Request	Demande de transaction
(Unconfirmed Private Modbus Rsp)	(Réponse Modbus privé non confirmé)
Modbus request	Demande Modbus

Figure 5 – Demande non confirmée avec un DTM de Communication Modbus Serial Line

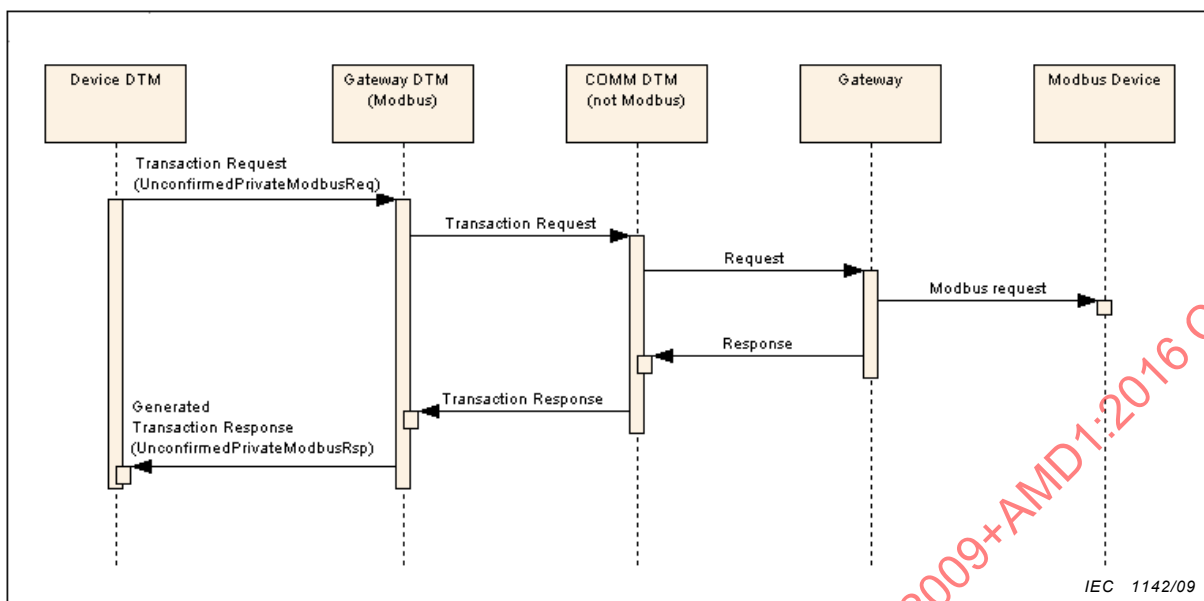


Légende

Anglais	Français
Device DTM	DTM de dispositif
Gateway DTM (MB TCP/MB SL)	DTM de passerelle (MB TCP/MB SL)
COMM DTM Modbus TCP	DTM de communication Modbus TCP
Gateway (MB TCP/MB SL)	Passerelle (MB TCP/MB SL)
Device MB SL	Dispositif MB SL
Transaction Request	Demande de transaction
(Unconfirmed Private Modbus Req)	(Demande Modbus privé non confirmé)
Generated Transaction Response	Réponse de transaction générée
(Unconfirmed Private Modbus Rsp)	(Réponse Modbus privé non confirmé)
Modbus request	Demande Modbus

Figure 6 – Demande non confirmée avec un DTM de Communication Modbus TCP

Si aucun DTM de Communication Modbus n'est utilisé afin d'établir la communication, le DTM de la passerelle Modbus, qui assure la liaison à un niveau de communication plus élevé, différent de Modbus, doit fournir cette réponse de transaction (voir Figure 7).



Légende

Anglais	Français
Device DTM	DTM de dispositif
Gateway DTM (Modbus)	DTM de passerelle (Modbus)
COMM DTM (Not Modbus)	DTM de communication (différent de Modbus)
Gateway	Passerelle
Modbus Device	Dispositif Modbus
Transaction Request	Demande de transaction
Transaction response	Réponse de transaction
(Unconfirmed Private Modbus Req)	Demande Modbus privé non confirmé
Generated Transaction Response	Réponse de transaction générée
(Unconfirmed Private Modbus Rsp)	Réponse Modbus privé non confirmé
Modbus request	Demande Modbus

Figure 7 – Demande non confirmée sans DTM de Communication Modbus

7 Utilisation spécifique à un protocole des types de données généraux

Le tableau suivant (Tableau 4) présente la manière dont les types de données généraux, définis dans l'IEC 62453-2 au sein de l'espace de noms 'fdt', sont utilisés avec les dispositifs de la CPF 15 de l'IEC 61784.

**Tableau 4 – Utilisation spécifique à un protocole
des types de données généraux de FDT**

Type de données	Description d'utilisation
fdt:dataType	Identificateur pour le type de données de FDT de l'objet de la voie
fdt:busCategory	Voir Article 4.
deviceTypeInfoInformation	Description du dispositif Modbus sous forme d'une chaîne lisible par l'homme, pouvant être fournie avec un DTM
deviceTypeInfoInformationPath	Chemin d'accès au fichier contenant une description du dispositif Modbus, si celle-ci est disponible
fdt:readAccess	Spécifie si, oui ou non, la valeur peut être lue à partir du dispositif: • "0", si la lecture n'est pas autorisée • "1", si la lecture est autorisée
fdt:systemTag	Identificateur unique d'une instance de dispositif au sein d'un projet d'une Application cadre
fdt:writeAccess	Spécifie si, oui ou non, la valeur peut être écrite au dispositif: • "0", si l'écriture n'est pas autorisée • "1", si l'écriture est autorisée

8 Types de données communs spécifiques à un protocole

8.1 Généralités

Cet article spécifie les types de données communs spécifiques à un protocole qui sont utilisés pour définir d'autres types de données.

8.2 Information d'adresse

Ce paragraphe spécifie la structure et la sémantique de l'information d'adresse spécifique à un protocole dans le FDT, qui est utilisée sur les différentes couches de communication sous-jacentes pour Modbus TCP et Modbus Serial Line.

Pour une description détaillée de la manière dont les informations d'adresse et les types de données sont mis en correspondance, voir Tableau 5 et Tableau 6.

Les types de données décrits dans ce paragraphe sont définis pour l'espace de noms suivant.
Espace de noms: mb

Tableau 5 – Types de données simples pour l'information d'adresse

Type de données	Définition	Description
slaveAddress	USINT	Adresse d'esclave d'un dispositif Modbus
tcpAddress	STRING	Représentation sous forme de chaîne de l'adresse IP comme décrit dans le Protocole Internet (IP) RFC 791
tcpPort	UINT	Port pour la connexion Modbus TCP (si aucune information relative au port n'est fournie, le port par défaut 502 doit être utilisé)

Tableau 6 – Types de données structurés pour l'information d'adresse

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ModbusSerial	STRUCT			Information d'adresse pour Modbus Serial Line
	slaveAddress	M	[1..1]	
ModbusTCP	STRUCT			Information d'adresse pour Modbus TCP: Adresse IP du serveur cible Modbus TCP Adresse d'esclave du dispositif cible si le serveur Modbus TCP fournit des services d'acheminement sur Modbus Serial Line
	slaveAddress	O	[0..1]	
	tcpAddress	M	[1..1]	
	tcpPort	O	[0..1]	

9 Types de données pour la gestion du réseau

Les types de données décrits dans cet article sont utilisés dans les services suivants:

- Service NetworkManagementInfoRead;
- Service NetworkManagementInfoWrite.

Cet article décrit les informations devant être fournies pour les dispositifs Modbus, dans le type de données UserDefinedBus (voir Tableau 7).

Tableau 7 – Types de données structurés pour la gestion du réseau

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
UserDefinedBus	STRUCT			Le type de données UserDefinedBus comporte l'information d'adresse du dispositif. Le type de données UserDefinedBus comporte au minimum un type de données pour l'information d'adresse du dispositif cible
	mb:ModbusSerial	O	[0..*]	
	mb:ModbusTCP	O	[0..*]	

Certains dispositifs et leurs modules requièrent plus d'une adresse sur le bus. Il est possible que le type de données UserDefinedBus puisse comporter plusieurs éléments de l'adresse.

Dans ce cas, le premier élément de l'adresse doit contenir l'information d'adresse principale du dispositif.

10 Types de données de communication

10.1 Généralités

Les types de données décrits dans cet article sont utilisés dans les services suivants:

- service connexion;
- service déconnexion;
- service transaction.

Cet article décrit la structure et la sémantique des données liées à la communication spécifique à un protocole, échangées entre un DTM et son composant parent.

Pour une description détaillée des types de données communs de communication utilisés dans ces services, voir Tableau 8.

Les types de données décrits dans cet article sont définis pour l'espace de noms suivant.
Espace de noms: fdtmodbus

Tableau 8 – Types de données communs de communication

Type de données	Définition	Description
communicationReference	UUID	Identification obligatoire pour une liaison de communication à un dispositif. Cet identificateur est attribué par le composant de communication durant la connexion. L'information d'adresse doit être utilisée pour l'ensemble des appels de communication suivants

10.2 Types de données pour la gestion de la connexion

Pour une description détaillée des informations utilisées dans les services connexion et déconnexion, voir Tableau 9.

Tableau 9 – Types de données structurés pour la gestion du service connexion

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
Abort	STRUCT			Si la commande Abot (arrêt prématuré) est envoyée au composant de communication suivant ou au dispositif connecté, cela met fin à l'ensemble des demandes et retours en cours, sans attendre de résultat. L'arrêt de la connexion ne sera pas confirmé
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Décrit la demande de communication: – pour établir une liaison de communication à un dispositif du serveur Modbus TCP – pour établir une liaison de communication à un dispositif esclave Modbus TCP La demande comporte l'information d'adresse du dispositif cible
	fdt:systemTag	M	[1..1]	
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
ConnectResponse	STRUCT			Décrit la réponse de communication à la demande de connexion
	communicationReference	M	[1..1]	
DisconnectRequest	STRUCT			Décrit la demande de communication pour libérer une liaison de communication
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Décrit la réponse de communication à une demande de déconnexion
	communicationReference	M	[1..1]	

10.3 Types de données spécifiques au service transaction

10.3.1 Généralités

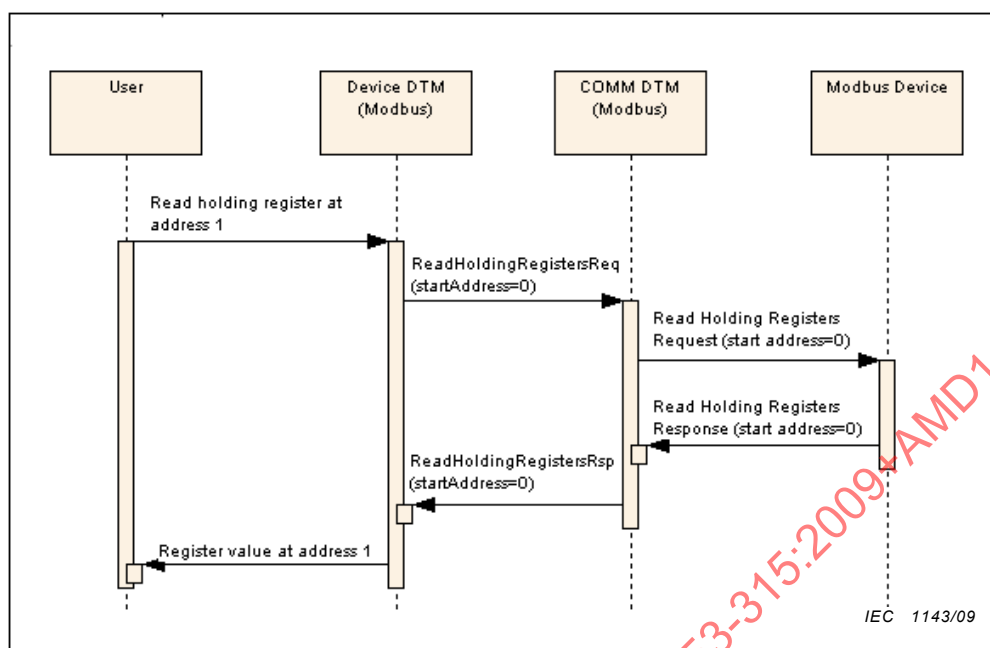
Ce paragraphe décrit les informations utilisées dans les services transaction.

10.3.2 Adressage des éléments de données

Le présent document spécifie la mise en correspondance des paramètres de Modbus et des types de données de FDT. Pour cette raison, toutes les informations d'adresse fournies comme paramètres et concernant les éléments de données Modbus

- bobines,
- entrées discrètes,
- registres d'exploitation et
- registres d'entrée,

doivent être identiques à celles envoyées ultérieurement sur le bus (voir Figure 8).



Légende

Anglais	Français
User	Utilisateur
Device DTM (Modbus)	DTM de dispositif (Modbus)
COMM DTM Modbus TCP	DTM de communication Modbus TCP
Modbus Device	Dispositif Modbus
Read holding register at adress 1	Lecture du registre d'exploitation à l'adresse 1
Register value at adress 1	Valeur du registre à l'adresse 1
Read Holding Registers Req (start adress = 0)	Demande de lecture des registres d'exploitation (adresse de départ = 0)
Read Holding Registers Rsp Request (start adress = 0)	Réponse de lecture des registres d'exploitation Demande (adresse de départ = 0)
Response (start adress = 0)	Réponse (adresse de départ = 0)

Figure 8 – Adressage des éléments de données

10.3.3 Service transaction Read coils

Les paramètres du service transaction, équivalent au service Read coils (lecture des bobines) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 10, Tableau 11, Tableau 12 et Tableau 13.

Tableau 10 – Types de données simples ReadCoilsReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
startAddress	UINT	Adresse de la première bobine lue	Adresse de la première bobine
quantity	UINT	Nombre de bobines lues	Quantité de bobines

Tableau 11 – Types de données structurés ReadCoilsReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadCoilsReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture des bobines définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Tableau 12 – Types de données simples ReadCoilsRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
multipleCoilValues	STRING	Chaîne ASCII avec l'état de chacune des bobines codé en un caractère, le premier caractère de la chaîne représentant la première bobine lue: – "0" = FALSE (FAUX) ou "OFF" – "1" = TRUE (VRAI) ou "ON"	Données

Tableau 13 – Types de données structurés ReadCoilsRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadCoilsRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture des bobines définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	multipleCoilValues	M	[1..1]	

10.3.4 Service transaction Read discrete inputs

Les paramètres du service transaction Read discrete inputs (lecture d'entrées discrètes), équivalent au service Read Discretes (lecture des entrées discrètes) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 14, Tableau 15, Tableau 16 et Tableau 17.

Tableau 14 – Types de données simples ReadDiscreteInputsReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
startAddress	UINT	Adresse de la première entrée discrète lue	Adresse de la première entrée discrète
quantity	UINT	Nombre d'entrées discrètes lues	Quantité d'entrées discrètes

Tableau 15 – Types de données structurés ReadDiscreteInputsReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadDiscreteInputsReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture des entrées discrètes définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Tableau 16 – Types de données simples ReadDiscreteInputsRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
discreteInputsStatus	STRING	Chaîne ASCII avec l'état de chacune des entrées discrètes codé en un caractère, le premier caractère de la chaîne représentant la première entrée discrète lue: – "0" = FALSE (FAUX) ou "OFF" – "1" = TRUE (VRAI) ou "ON"	Données

Tableau 17 – Types de données structurés ReadDiscreteInputsRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadDiscreteInputsRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture des entrées discrètes définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	discreteInputsStatus	M	[1..1]	

10.3.5 Service transaction Read holding registers

Les paramètres du service transaction, équivalent au service Read Holding Registers (lecture des registres d'exploitation) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 18, Tableau 19, Tableau 20 et Tableau 21.

Tableau 18 – Types de données simples ReadHoldingRegistersReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
startAddress	UINT	Adresse du premier registre d'exploitation lu	Adresse du premier registre d'exploitation à lire
quantity	UINT	Nombre de registres d'exploitation lus	Quantité de registres d'exploitation à lire

Tableau 19 – Types de données structurés ReadHoldingRegistersReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadHoldingRegistersReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture des registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	startAddress	M	[1..1]	
	quantity	M	[1..1]	

Tableau 20 – Types de données simples ReadHoldingRegistersRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
registerValues	ARRAY OF WORD	Valeurs de lecture des registres d'exploitation	Données

Tableau 21 – Types de données structurés ReadHoldingRegistersRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadHoldingRegistersRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture des registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	registerValues	M	[1..1]	

10.3.6 Service transaction Read input registers

Les paramètres du service transaction, équivalent au service Read Input Registers (lecture des registres d'entrée) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 22, Tableau 23, Tableau 24 et Tableau 25.

Tableau 22 – Types de données simples ReadInputRegistersReq

Type de données	Définition	Description	Paramètre de service dans l'IEC 61158-5-15
startAddress	UINT	Adresse du premier registre d'entrée lu	Adresse du premier registre d'entrée
quantity	UINT	Nombre de registres d'entrée lus	Quantité de registres d'entrée

Tableau 23 – Types de données structurés ReadInputRegistersReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadInputRegistersReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture des registres d'entrée définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Tableau 24 – Types de données simples ReadInputRegistersRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
registerValues	ARRAY OF WORD	Valeurs de lecture des registres d'entrée	Données

Tableau 25 – Types de données structurés ReadInputRegistersRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadInputRegistersRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture des registres d'entrée définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	registerValues	M	[1..1]	

10.3.7 Service transaction Write single coil

Les paramètres du service transaction, équivalent au service Write Single Coil (écriture d'une seule bobine) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 26, Tableau 27 et Tableau 28.

Tableau 26 – Types de données simples WriteSingleCoilReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
outputAddress	UINT	Adresse de la bobine forcée	Adresse de la première bobine
singleCoilValue	BOOL	État de la bobine forcée, avec: – "0" = FALSE (FAUX) ou "OFF" – "1" = TRUE (VRAI) ou "ON"	Données d'une seule bobine

Tableau 27 – Types de données structurés WriteSingleCoilReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteSingleCoilReq	STRUCT			Demande de communication équivalente à la demande au service pour l'écriture d'une seule bobine définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	singleCoilValue	M	[1..1]	

Tableau 28 – Types de données structurés WriteSingleCoilRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteSingleCoilRsp	STRUCT			Réponse de communication équivalente à la réponse du service d'écriture d'une seule bobine définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.8 Service transaction Write single register

Les paramètres du service transaction Write single register (écriture d'un seul registre), équivalent au service Write Single Holding Register (écriture d'un seul registre d'exploitation) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 29, Tableau 30 et dans le Tableau 31.

Tableau 29 – Types de données simples WriteSingleRegisterReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
outputAddress	UINT	Adresse du registre d'exploitation écrit	Adresse du premier registre d'exploitation à écrire
singleRegister	WORD	Valeur écrite sur le registre	Données

Tableau 30 – Types de données structurés WriteSingleRegisterReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteSingleRegisterReq	STRUCT			Demande de communication équivalente à la demande au service pour l'écriture d'un seul registre d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	singleRegister	M	[1..1]	

Tableau 31 – Types de données structurés WriteSingleRegisterRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteSingleRegisterRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour l'écriture d'un seul registre d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.9 Service transaction Read exception status

Les paramètres du service transaction, équivalent au service Read Exception Status (lecture des statuts d'exception) défini dans la spécification du Protocole pour l'Application de MODBUS, sont présentés dans les Tableau 32, Tableau 33 et Tableau 34.

Tableau 32 – Types de données structurés ReadExceptionStatusReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadExceptionStatusReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture des statuts d'exception définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	

NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série

Tableau 33 – Types de données simples ReadExceptionStatusRsp

Type de données	Définition	Description	Paramètre de service équivalent dans le protocole pour l'Application de MODBUS
exceptionStatus	BYTE	Statut d'exception du dispositif Modbus Serial Line	Données de sortie

Tableau 34 – Types de données structurés ReadExceptionStatusRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadExceptionStatusRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture des statuts d'exception définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
	exceptionStatus	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

10.3.10 Service transaction Diagnostics

Les paramètres du service transaction, équivalent au service Diagnostics (diagnostics) défini dans la spécification du Protocole pour l'Application de MODBUS sont présentés dans les Tableau 35, Tableau 36 et Tableau 37.

Tableau 35 – Types de données simples pour les diagnostics

Type de données	Définition	Description	Paramètre de service équivalent dans le Protocole pour l'Application de MODBUS
diagnosticsData	ARRAY OF WORD	Données requises, selon le type de sous-fonction	Données
diagnosticsSubFct	UINT	Code de sous-fonction	Sous-fonction

Tableau 36 – Types de données structurés DiagnosticsReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
DiagnosticsReq	STRUCT			Demande de communication équivalente à la demande au service pour les diagnostics définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
	diagnosticsData	M	[1..1]	
	diagnosticsSubFct	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

Tableau 37 – Types de données structurés DiagnosticsRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
DiagnosticsRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour les diagnostics définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
	diagnosticsData	M	[1..1]	
	diagnosticsSubFct	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

10.3.11 Service transaction Get Comm event counter

Les paramètres du service transaction, équivalent au service Get Comm Event Counter (récupération du compteur des événements de communication) défini dans la spécification du Protocole pour l'Application de MODBUS, sont présentés dans les Tableau 38, Tableau 39 et Tableau 40.

Tableau 38 – Types de données structurés GetCommEventCounterReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
GetCommEventCounterReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture du compteur des événements de communication définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

Tableau 39 – Types de données simples GetCommEventCounterRsp

Type de données	Définition	Description	Paramètre de service équivalent dans le Protocole pour l'Application de MODBUS
commStatus	WORD	Information de statut de 2 octets. L'information de statut sera 0xFFFF si un programme mis en place précédemment est encore utilisé par le dispositif à distance (état occupé). Dans le cas contraire, elle sera	Statut

Type de données	Définition	Description	Paramètre de service équivalent dans le Protocole pour l'Application de MODBUS
		0x0000.	
eventCount	UINT	Nombre d'événements comptés par le dispositif	Décompte des événements

Tableau 40 – Types de données structurés GetCommEventCounterRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
GetCommEventCounterRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture du compteur des événements de communication définie dans la spécification du Protocole pour l'Application de MODBUS
	commStatus	M	[1..1]	
	communicationReference	M	[1..1]	
	eventCount	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

10.3.12 Service transaction Get Comm event log

Les paramètres du service transaction, équivalent au service Get Comm Event Log (récupérer le journal des événements de communication) défini dans la spécification du Protocole pour l'Application de MODBUS sont présentés dans les Tableau 41, Tableau 42 et Tableau 43.

Tableau 41 – Types de données structurés GetCommEventLogReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
GetCommEventLogReq	STRUCT			Demande de communication équivalente à la demande au service pour le registre d'historique des événements définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

Tableau 42 – Types de données simples GetCommEventLogRsp

Type de données	Définition	Description	Paramètre de service équivalent dans le Protocole pour l'Application de MODBUS
commStatus	WORD	Information de statut de 2 octets. L'information de statut sera 0xFFFF si un programme mis en place précédemment est encore utilisé par le dispositif à distance (état occupé). Dans le cas contraire, elle sera 0x0000	Statut
eventCount	UINT	Nombre d'événements comptés par le dispositif	Décompte des événements
events	ARRAY OF BYTE	Chaque octet correspond au statut d'envoi ou de réception de MODBUS pour le dispositif à distance	Événements
messageCount	UINT	Quantité de messages traités par le dispositif à distance depuis son dernier redémarrage, la remise à zéro des compteurs ou la procédure de mise sous tension	Décompte des messages

Tableau 43 – Types de données structurés GetCommEventLogRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
GetCommEventCounter Rsp	STRUCT			Réponse de communication équivalente à la réponse du service pour le registre d'historique des événements définie dans la spécification du Protocole pour l'Application de MODBUS
	commStatus	M	[1..1]	
	communicationReference	M	[1..1]	
	eventCount	M	[1..1]	
	events	O	[0..1]	
	messageCount	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

10.3.13 Service transaction Write multiple coils

Les paramètres du service transaction, équivalent au service Write Multiple Coils (écriture de plusieurs bobines) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 44, Tableau 45 et Tableau 46.

Tableau 44 – Types de données simples WriteMultipleCoilsReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
multipleCoilValues	STRING	Chaîne ASCII avec chaque état de bobine codé en un caractère, le premier caractère de la chaîne représentant la première bobine dans laquelle est effectuée l'écriture: – "0" = FALSE (FAUX) ou "OFF" – "1" = TRUE (VRAI) ou "ON"	Données
outputAddress	UINT	Adresse de la première bobine forcée	Adresse de la première bobine

Tableau 45 – Types de données structurés WriteMultipleCoilsReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteMultipleCoilsReq	STRUCT			Demande de communication équivalente à la demande au service pour l'écriture de plusieurs bobines définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	multipleCoilValues	M	[1..1]	
	outputAddress	M	[1..1]	

Tableau 46 – Types de données structurés WriteMultipleCoilsRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteMultipleCoilsRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour l'écriture de plusieurs bobines définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.14 Service transaction Write multiple registers

Les paramètres du service transaction, équivalent au service Write Multiple Holding Registers (écriture de plusieurs registres d'exploitation) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 47, Tableau 48 et Tableau 49.

Tableau 47 – Types de données simples WriteMultipleRegistersReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
outputAddress	UINT	Adresse du premier registre d'exploitation écrit	Adresse du premier registre d'exploitation à écrire
registerValues	ARRAY OF WORD	Valeurs du registre écrites	Données

Tableau 48 – Types de données structurés WriteMultipleRegistersReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteMultipleRegistersReq	STRUCT			Demande de communication équivalente à la demande au service pour l'écriture de plusieurs registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	registerValues	M	[1..1]	

Tableau 49 – Types de données structurés WriteMultipleRegistersRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteMultipleRegistersRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour l'écriture de plusieurs registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.15 Service transaction Report slave ID

Les paramètres du service transaction, équivalent au service Report Slave ID (rapporter l'identificateur d'esclave) défini dans la spécification du Protocole pour l'Application de MODBUS sont présentés dans les Tableau 50, Tableau 51 et Tableau 52.

Tableau 50 – Types de données structurés ReportSlaveIDReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReportSlaveIDReq	STRUCT			Demande de Communication équivalente à la demande au service pour l'état de l'identifiant de l'esclave définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

Tableau 51 – Types de données simples ReportSlaveIDRsp

Type de données	Définition	Description	Paramètre de service équivalent dans le protocole pour l'Application de MODBUS
data	ARRAY OF BYTE	Cet attribut comporte: • l'identifiant de l'esclave, • le Statut de l'Indicateur d'Activité (0x00 ou 0xFF) • et les données supplémentaires spécifiques au dispositif dans le même format et le même ordre que ceux définis dans la Spécification du Protocole pour l'Application de MODBUS	Identifiant de l'esclave, Statut de l'Indicateur d'Activité Données supplémentaires

Tableau 52 – Types de données structurés ReportSlaveIDRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReportSlaveIDRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour l'état de l'identifiant de l'esclave définie dans la spécification du Protocole pour l'Application de MODBUS
	communicationReference	M	[1..1]	
	data	M	[1..1]	
NOTE Conformément à la Spécification du Protocole pour l'Application de MODBUS V1.1a, ce service de Modbus ne peut être utilisé qu'avec Modbus sur ligne série				

10.3.16 Service transaction Read file record

Les paramètres du service transaction, équivalent au service Read File Record (lecture d'enregistrements de fichiers) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 53 à Tableau 58.

Tableau 53 – Types de données simples ReadFileSubRequest

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
fileNumber	UINT	Numéro du fichier	Numéro du fichier
quantity	UINT	Longueur des enregistrements à lire comme la quantité de registres	Longueur des enregistrements
recordNumber	UINT	Numéro d'enregistrement de départ au sein du fichier	Numéro d'enregistrement
referenceType	USINT	Type de référence	Type de référence
NOTE Conformément à l'IEC 61158-5-15, le type de référence doit être spécifié en tant que 0x06.			

Tableau 54 – Types de données structurés ReadFileSubRequest

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadFileSubRequest	STRUCT			Sous-demande équivalente à la sous-demande de lecture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	fileNumber	M	[1..1]	
	quantity	M	[1..1]	
	recordNumber	M	[1..1]	
	referenceType	M	[1..1]	

Tableau 55 – Types de données structurés ReadFileRecordReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadFileRecordReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	ReadFileSubRequest	M	[1..*]	

Tableau 56 – Types de données simples ReadFileSubResponse

Type de données	Définition	Description	Equivalent IEC 61158-5-15 service parameter
recordData	ARRAY OF WORD	Données d'enregistrement	Matrice de données d'enregistrement

Tableau 57 – Types de données structurés ReadFileSubResponse

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadFileSubResponse	STRUCT			Sous-réponse équivalente à la sous-réponse de lecture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	recordData	M	[1..1]	

Tableau 58 – Types de données structurés ReadFileRecordRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadFileRecordRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	ReadFileSubResponse	M	[1..*]	

10.3.17 Service transaction Write file record

Les paramètres du service transaction, équivalent au service Write File Record (écriture d'enregistrement de fichiers) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 59 à Tableau 62.

Tableau 59 – Types de données simples WriteFileSubRequest

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
fileNumber	UINT	Numéro du fichier	Numéro du fichier
recordData	ARRAY OF WORD	Longueur des enregistrements à lire comme la quantité de registres	Matrice de données d'enregistrement
recordNumber	UINT	Numéro d'enregistrement de départ au sein du fichier	Numéro d'enregistrement
referenceType	USINT	Type de référence	Type de référence
NOTE Conformément à l'IEC 61158-5-15, le type de référence doit être spécifié en tant que 0x06.			

Tableau 60 – Types de données structurés WriteFileSubRequest

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteFileSubRequest	STRUCT			Sous-demande équivalente à la sous-demande d'écriture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	fileNumber	M	[1..1]	
	recordData	M	[1..1]	
	RecordNumber	M	[1..1]	
	ReferenceType	M	[1..1]	

Tableau 61 – Types de données structurés WriteFileRecordReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteFileRecordReq	STRUCT			Demande de communication équivalente à la demande au service pour l'écriture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	WriteFileSubRequest	M	[1..*]	

Tableau 62 – Types de données structurés WriteFileRecordRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
WriteFileRecordRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour l'écriture des fichiers d'enregistrement définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.18 Service transaction Mask write register

Les paramètres du service transaction Mask write register (écriture avec masque dans un registre), équivalent au service Mask Write Holding Register (écriture des registres d'exploitation en utilisant des masques) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 63, Tableau 64 et Tableau 65.

Tableau 63 – Types de données simples MaskWriteRegisterReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
andMask	WORD	Masque AND (ET)	Masque AND (ET)
orMask	WORD	Masque OR (OU)	Masque OR (OU)
referenceAddress	UINT	Adresse du registre d'exploitation auquel le masque doit être appliqué	Adresse du premier registre d'exploitation à écrire

Tableau 64 – Types de données structurés MaskWriteRegisterReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
MaskWriteRegisterReq	STRUCT			Demande de communication équivalente à la demande au service pour le masque d'écriture des registres d'exploitation définie dans l'IEC 61158-5-15
	andMask	M	[1..1]	
	communicationReference	M	[1..1]	
	orMask	M	[1..1]	
	referenceAddress	M	[1..1]	

Tableau 65 – Types de données structurés MaskWriteRegisterResp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
MaskWriteRegisterResp	STRUCT			Réponse de communication équivalente à la réponse du service pour le masque d'écriture des registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.19 Service transaction Read/write holding registers

Les paramètres du service transaction, équivalent au service Read/Write Holding Registers (lecture/écriture des registres d'exploitation) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 66, Tableau 67, Tableau 68 et Tableau 69.

Tableau 66 – Types de données simples ReadWriteRegistersReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
readQuantity	UINT	Nombre de registres d'exploitation lus	Quantité de registres d'exploitation à lire
readStartAddress	UINT	Adresse du premier registre d'exploitation lu	Adresse du premier registre d'exploitation à lire
writeRegisterValues	ARRAY OF WORD	Valeurs du registre écrites	Données
writeStartAddress	UINT	Adresse du premier registre d'exploitation écrit	Adresse du premier registre d'exploitation à écrire

Tableau 67 – Types de données structurés ReadWriteRegistersReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadWriteRegistersReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture/écriture des registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	readQuantity	M	[1..1]	
	readStartAddress	M	[1..1]	
	writeRegisterValues	M	[1..1]	
	writeStartAddress	M	[1..1]	

Tableau 68 – Types de données simples ReadWriteRegistersRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
readRegisterValues	ARRAY OF WORD	Valeurs de lecture des registres d'exploitation	Données

Tableau 69 – Types de données structurés ReadWriteRegistersRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadWriteRegistersRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture/écriture des registres d'exploitation définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	readRegisterValues	M	[1..1]	

10.3.20 Service transaction Read FIFO queue

Les paramètres du service transaction, équivalent au service Read FIFO (la lecture de file d'attente FIFO) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 70, Tableau 71, Tableau 72 et Tableau 73.

Tableau 70 – Types de données simples ReadFifoQueueReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
fifoPointerAddress	UINT	Adresse du premier registre lu dans une file d'attente First-In-First-Out (FIFO) (Premier entré premier sorti) de registres	Adresse de la file d'attente FIFO

Tableau 71 – Types de données structurés ReadFifoQueueReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadFifoQueueReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture de FIFO définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	fifoPointerAddress	M	[1..1]	

Tableau 72 – Types de données simples ReadFifoQueueRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
fifoRegisterValues	ARRAY OF WORD	Valeurs de registre lues à partir de la file d'attente FIFO	Données

Tableau 73 – Types de données structurés ReadFifoQueueRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadFifoQueueRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture de FIFO définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	fifoRegisterValues	M	[1..1]	

10.3.21 Service transaction Encapsulated interface transport

Les paramètres du service transaction Encapsulated interface transport (transport par l'interface encapsulée), équivalent au service fourni par l'interface encapsulée (Encapsulated Interface) défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 74, Tableau 75, Tableau 76 et Tableau 77.

Tableau 74 – Types de données simples EncapsulatedInterfaceTransportReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
meiData	ARRAY OF BYTE	Données spécifiques à chaque interface encapsulée de Modbus	Données spécifiques au type de MEI (Interface encapsulée de Modbus)
meiType	USINT	Type d'interface encapsulée de Modbus (type de MEI)	Type de MEI

Tableau 75 – Types de données structurés EncapsulatedInterfaceTransportReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
EncapsulatedInterfaceTransportReq	STRUCT			Demande de communication équivalente à la demande au service fourni par l'interface encapsulée définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	meiData	M	[1..1]	
	meiType	M	[1..1]	

Tableau 76 – Types de données simples EncapsulatedInterfaceTransportRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
meiData	ARRAY OF BYTE	Données spécifiques à chaque interface encapsulée de Modbus	Données spécifiques au type de MEI (Interface encapsulée de Modbus)
meiType	USINT	Type d'interface encapsulée de Modbus (type de MEI)	Type de MEI

Tableau 77 – Types de données structurés EncapsulatedInterfaceTransportRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
EncapsulatedInterfaceTransportRsp	STRUCT			Réponse de communication équivalente à la réponse du service fourni par l'interface encapsulée définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	meiData	M	[1..1]	
	meiType	M	[1..1]	

10.3.22 Service transaction Read device identification

Les paramètres du service transaction, équivalent au service Read Device Identification (lecture de l'identification du dispositif) fourni par l'interface encapsulée et défini dans l'IEC 61158-5-15, sont présentés dans les Tableau 78 à Tableau 83.

Tableau 78 – Types de données simples ReadDeviceIdentificationReq

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
readDeviceIdCode	USINT	Le paramètre " Code de lecture de l'identifiant du dispositif " permet de définir quatre types d'accès: 01: demande pour obtenir l'identification du dispositif de base (accès par flux) 02: demande pour obtenir l'identification ordinaire du dispositif (accès par flux) 03: demande pour obtenir l'identification étendue du dispositif (accès par flux) 04: demande pour obtenir un objet pour l'identification spécifique (accès individuel)	Code de lecture de l'identifiant du dispositif
objectId	USINT	Identification du premier objet à obtenir	Identifiant de l'objet requis

Tableau 79 – Types de données structurés ReadDeviceIdentificationReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadDeviceIdentificationReq	STRUCT			Demande de communication équivalente à la demande au service pour la lecture de l'identification du dispositif fourni par l'interface encapsulée définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	readDeviceIdCode	M	[1..1]	
	objectId	M	[1..1]	

Tableau 80 – Types de données simples IdentificationObject

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
objectId	USINT	Identification de l'objet retourné	Identifiant de l'objet retourné
objectValue	ARRAY OF BYTES	Chaîne ASCII représentant la valeur de l'objet	Valeur de l'objet

Tableau 81 – Types de données structurés IdentificationObject

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
IdentificationObject	STRUCT			Objets d'identification de la réponse du service pour la lecture de l'identification du dispositif fournie par l'interface encapsulée définie dans l'IEC 61158-5-15
	objectId	M	[1..1]	
	objectValue	M	[1..1]	

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

Tableau 82 – Types de données simples ReadDeviceIdentificationRsp

Type de données	Définition	Description	Paramètre de service équivalent dans l'IEC 61158-5-15
conformityLevel	BYTE	Niveau de conformité pour l'identification du dispositif et le type d'accès pris en charge 01 : identification de base (accès par flux uniquement) 02 : identification ordinaire (accès par flux uniquement) 03 : identification étendue (accès par flux uniquement) 81 : identification de base (accès par flux et accès individuel) 82 : identification ordinaire (accès par flux et accès individuel) 83 : identification étendue (accès par le flux et accès individuel)	Niveau de conformité
moreFollows	BOOL	Dans le cas d'un readDeviceIdCode 01, 02 ou 03 (accès par flux), si les données d'identification ne rentrent pas dans une seule réponse et plusieurs demandes/réponses, des transactions peuvent être requises: – "0" : plus aucun objet n'est disponible – "1" : d'autres objets d'identification sont disponibles et des transactions Modbus ultérieures sont requises Dans le cas d'un ReadDevId code 04 (accès individuel), ce champ doit être mis à "0"	Augmentation des fanions disponibles
nextObjectId	USINT	– "0", si aucun autre objet d'identification n'est disponible (moreFollows="0") – Identification du prochain objet obtenu, si d'autres objets d'identification sont disponibles (moreFollows="1")	Identifiant du prochain objet
numberOfObjects	USINT	Nombre d'objets d'identification retournés dans cette réponse (pour un accès individuel, numberOfObjects="1")	Nombre d'objets
readDeviceIdCode	USINT	Le paramètre " Code de lecture de l'identifiant du dispositif " permet de définir quatre types d'accès: 01: demande pour obtenir l'identification de base du dispositif (accès par flux) 02: demande pour obtenir l'identification ordinaire du dispositif (accès par flux) 03: demande pour obtenir l'identification étendue du dispositif (accès par flux) 04: demande pour obtenir un objet pour l'identification spécifique (accès individuel)	Code de lecture de l'identifiant du dispositif

Tableau 83 – Types de données structurés ReadDeviceIdentificationRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ReadDeviceIdentificationRsp	STRUCT			Réponse de communication équivalente à la réponse du service pour la lecture de l'identification du dispositif fournie par l'interface encapsulée définie dans l'IEC 61158-5-15
	communicationReference	M	[1..1]	
	conformityLevel	M	[1..1]	
	IdentificationObject	M	[1..*]	
	moreFollows	M	[1..1]	
	nextObjectId	M	[1..1]	
	numberOfObjects	M	[1..1]	
	readDeviceIdCode	M	[1..1]	

10.3.23 Service transaction Private Modbus

Ce service transaction Private Modbus (Modbus privé) peut être utilisé afin d'envoyer n'importe quelle demande privée à Modbus. Les paramètres de ce service de transaction sont présentés dans les Tableau 84, Tableau 85, Tableau 86 et Tableau 87.

Tableau 84 – Types de données simples PrivateModbusReq

Type de données	Définition	Description
privateRequest	ARRAY OF BYTE	Suite de chiffres hexadécimaux représentant la demande privée de Modbus

Tableau 85 – Types de données structurés PrivateModbusReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
PrivateModbusReq	STRUCT			Demande de communication du service transaction de Modbus en mode privé
	communicationReference	M	[1..1]	
	privateRequest	M	[1..1]	

Tableau 86 – Types de données simples PrivateModbusRsp

Type de données	Définition	Description
privateResponse	ARRAY OF BYTE	Suite de chiffres hexadécimaux représentant la réponse privée de Modbus

Tableau 87 – Types de données structurés PrivateModbusRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
PrivateModbusRsp	STRUCT			Réponse de communication du service transaction de Modbus en mode privé
	communicationReference	M	[1..1]	
	privateResponse	M	[1..1]	

10.3.24 Service transaction Unconfirmed private Modbus

Ce service transaction Unconfirmed private Modbus (Modbus privé non confirmé) peut être utilisé afin d'envoyer n'importe quelle demande privée non confirmée à Modbus. Les paramètres de ce service transaction sont présentés dans les Tableau 88, Tableau 89 et Tableau 90. Bien que ce service fasse état d'un service de Modbus non confirmé, une réponse de transaction doit être retournée au niveau du FDT afin d'informer le DTM que la demande a été envoyée. Pour de plus amples informations sur cette action, voir 6.3.

Tableau 88 – Types de données simples UnconfirmedPrivateModbusReq

Type de données	Définition	Description
privateRequest	ARRAY OF BYTE	Suite de chiffres hexadécimaux représentant la demande privée non confirmée à Modbus

Tableau 89 – Types de données structurés UnconfirmedPrivateModbusReq

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
UnconfirmedPrivateModbusReq	STRUCT			Demande de communication au service transaction de Modbus en mode privé non confirmé
	communicationReference	M	[1..1]	
	privateRequest	M	[1..1]	

Tableau 90 – Types de données structurés UnconfirmedPrivateModbusRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
UnconfirmedPrivateModbusRsp	STRUCT			Réponse de communication du service transaction de Modbus en mode privé non confirmé
	communicationReference	M	[1..1]	

10.3.25 Réponse d'exception de Modbus

Les paramètres de la réponse du service transaction, équivalente à la réponse d'exception définie dans l'IEC 61158-5-15, sont présentés dans le Tableau 91 et le Tableau 92.

Tableau 91 – Types de données Simple ModbusExceptionRsp

Typ de données	Définition	Description	Paramètre de service IEC 61158-5-15 équivalent
modbusExceptionCode	BYTE	Modbus Exception Code (Code d'exception Modbus)	Code d'exception (Exception code)
modbusService	énumération (ReadCoils ReadDiscreteInputs ReadHoldingRegisters ReadInputRegisters WriteSingleCoil WriteSingleRegister ReadExceptionStatus Diagnostics GetCommEventCounter GetCommEventLog WriteMultipleCoils WriteMultipleRegisters ReportSlaveID ReadFileRecord WriteFileRecord MaskWriteRegister ReadWriteRegisters ReadFifoQueue EncapsulatedInterfaceTransport ReadDeviceIdentification PrivateModbus)	Nom du service Modbus qui a échoué: – ReadCoils – ReadDiscreteInputs – ReadHoldingRegisters – ReadInputRegisters – WriteSingleCoil – WriteSingleRegister – ReadExceptionStatus – Diagnostics: Diagnostics – GetCommEventCounter – GetCommEventLog – WriteMultipleCoils – WriteMultipleRegisters – ReportSlaveID – ReadFileRecord – WriteFileRecord – MaskWriteRegister – ReadWriteRegisters – ReadFifoQueue – EncapsulatedInterfaceTransport – ReadDeviceIdentification – PrivateModbus	Non défini

Tableau 92 – Types de données structurés ModbusExceptionRsp

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ModbusExceptionRsp	STRUCT			Réponse de communication du service transaction de Modbus en mode privé
	communicationReference	M	[1..1]	
	modbusExceptionCode	M	[1..1]	
	modbusService	M	[1..1]	

11 Types de données pour les paramètres des voies

Les types de données décrits dans cet article sont utilisés dans les services suivants:

- ReadChannelInformation;
- WriteChannelInformation.

Un DTM peut fournir des Voies de Processus, afin d'exposer de manière détaillée aux systèmes externes tels que les contrôleurs, les applications IHM ou à l'Application cadre elle-même, les valeurs du processus disponibles sur les dispositifs. Les systèmes externes peuvent utiliser ces informations afin d'interpréter ou d'accéder directement aux valeurs d'exécution du dispositif avec les services normalisés du protocole.

Cet article décrit la structure et la sémantique des informations pouvant être exposées par les voies de processus fournies par les DTM. Un DTM choisit ou non de fournir des voies. Pour une description détaillée des paramètres des voies de processus, se référer au Tableau 93 et au Tableau 94.

Les types de données décrits dans cet article sont définis pour l'espace de noms suivant.
Espace de noms: modbuschannel

Tableau 93 – Types de données simples pour les paramètres des voies

Type de données	Définition	Description
address	UINT	Adresse de l'entrée discrète, de la bobine ou du registre auxquels il faut accéder. Lorsque l'accès doit se faire à une gamme de ces éléments de données Modbus, le type de données de l'adresse comporte l'adresse du premier élément de données Modbus de la gamme.
frameApplicationTag	STRING	Marqueur spécifique à l'Application cadre utilisé pour l'identification et la navigation. Il convient que le DTM affiche ce marqueur au niveau des interfaces utilisateur spécifiques à chaque voie.
gatewayBusCategory	UUID	L'unique identificateur pour un type de bus pris en charge tel que Profibus ou HART conformément au CATID spécifique au FDT.
iecDataType	énumération (BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TimeOfDay DateAndTime STRING BYTE WORD DWORD LWORD WSTRING)	Identificateur pour le type de données de l'IEC 61131-3 de l'objet de la voie
modbusDataTypes	énumération (coil discreteInput holdingRegister inputRegister)	Identificateur pour l'élément de données de Modbus de l'objet de la voie
protectedByChannelAssignment	BOOL	– “0”, aucune restriction pour accéder aux valeurs des paramètres. – “1”, si la voie est configurée afin de n'être lue que par l'Application cadre et ni l'Application cadre ni le DTM lui-même ne sont autorisés à modifier les valeurs des paramètres. Généralement configurée sur “1” si l'attribution d'une voie existe
quantity	UINT	Nombre d'entrées discrètes, de bobines ou de registres représentant l'objet de la voie dans le dispositif
scaleValue	STRING	Valeur REAL (RÉELLE) codée ASCII avec un point en tant que séparateur décimal. Cette valeur représente le facteur d'échelle par lequel une valeur actuelle doit être multipliée afin d'obtenir une valeur étalonnée
statusChannel	BOOL	TRUE (VRAI) si la voie est uniquement pour l'information de statut

Tableau 94 – Types de données structurés pour les paramètres des voies

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
FDTChannel	STRUCT			Description de la voie
	appld:applicationId	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:dataType	M	[1..1]	
	fdt:Deadband	O	[0..1]	
	fdt:descriptor	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:id	M	[1..1]	
	fdt:Ranges	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:signalType	M	[1..1]	
	fdt:StructuredElements	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:tag	M	[1..1]	
	fdt:Unit	O	[0..1]	
	frameApplicationTag	O	[0..1]	
	iecDataType	O	[0..1]	
	ModbusAccessData	O	[0..1]	
	protectedByChannel Assignment	M	[1..1]	
	UnitScaling	O	[0..1]	
FDTChannelType	STRUCT			Description des composants pour les voies dotées d'une fonctionnalité de passerelle
	fdt:VersionInformation	M	[1..1]	
ModbusAccessData	STRUCT			Ensemble des informations requises afin d'accéder à la valeur de processus avec des services normalisés de Modbus pour l'accès à la lecture ou à l'écriture
	address	M	[1..1]	
	modbusDataTypes	M	[1..1]	
	quantity	M	[1..1]	
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	
UnitScaling	STRUCT			Information relative à la mise à l'échelle de la valeur actuelle du paramètre fournie par l'objet de voie
	scaleValue	M	[1..1]	

12 Identification du dispositif

Les articles suivants décrivent les types d'informations relatives à l'identification du dispositif qui doivent être fournies par un dispositif et la manière dont ces informations sont interprétées au niveau du FDT afin de clairement identifier les dispositifs Modbus dans le FDT.

12.1 Types de données pour l'identification commune d'un type de dispositif

Cet paragraphe décrit les informations utilisées pour l'identification du dispositif spécifique à un protocole. Pour une description détaillée des paramètres d'identification du dispositif commun, voir Tableau 95.

Les types de données décrits dans cet article sont définis pour l'espace de noms suivant.
Espace de noms: mbident

Tableau 95 – Types de données simples pour l'identification commune

Type de données	Définition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	Définit le niveau de prise en charge d'un DTMDeviceType pour un dispositif physique. Ce type de données peut être utilisé par une Application cadre pour afficher le niveau de prise en charge d'un DTM à l'utilisateur. Les utilisateurs peuvent se servir de ces informations pour prendre une décision concernant une attribution. – genericSupport: DTMDeviceType s'applique à tous les types de dispositifs physiques du protocole de bus de terrain correspondant; – profileSupport: DTMDeviceType s'applique à tous les dispositifs physiques d'un certain profil de protocole de bus de terrain; – blockspecificProfileSupport DTMDeviceType s'applique aux blocs inclus dans les types de dispositifs physiques (ex: Pressure TransducerBlock dans Profibus PA); – specificSupport (DTMDeviceType a été développé pour ce type de dispositif physique; – identSupport: The DTMDeviceType est capable d'identifier le dispositif physique d'une façon spécifique au vendeur et de proposer un meilleur DTMDeviceType.
majorMinorRevision	STRING	Information de l'objet d'identification MajorMinorRevision spécifique au protocole
match	STRING	Chaîne d'expression ordinaire, qui doit correspondre à un type de données fourni par le résultat du balayage
name	enumeration (ProductName ModelName VendorURL UserApplicationName)	Nom de l'objet d'identification spécifique à un protocole (catégorie ordinaire)
nomatch	STRING	Chaîne d'expression ordinaire, qui ne doit pas correspondre à un type de données fourni par le résultat du balayage
productCode	STRING	Information de l'objet d'identification ProductCode spécifique au protocole
protocolName	STRING	Nom du protocole de communication pris en charge. Ce type de données doit comporter l'information "protocol_Modbus"

Type de données	Définition	Description
protocolSpecificName	enumeration (ProductName ModelName VendorURL UserApplicationName)	Nom spécifique au protocole de bus de terrain de l'information pour une seule identification
value	STRING	Information relative à une seule identification
vendorName	STRING	Information de l'objet d'identification VendorName spécifique à un protocole

12.2 Types de données pour le balayage topologique

Les types de données décrits dans cet article sont utilisés dans le service suivant:

- Balayage.

Cet article décrit les informations pour un des dispositifs présent dans la liste des dispositifs balayés. Pour une description détaillée des paramètres d'information du dispositif, voir Tableau 96 et Tableau 97.

Les types de données décrits dans ce paragraphe sont définis pour l'espace de noms suivant.
Espace de noms: fdtmodbusdevice

Tableau 96 – Types de données simples pour l'information relative au dispositif

Type de données	Définition	Description
majorMinorRevision	STRING	Information de l'objet d'identification MajorMinorRevision spécifique à un protocole
name	énumération (ProductName ModelName VendorURL UserApplicationName)	Nom de l'objet d'identification spécifique à un protocole (catégorie ordinaire)
productCode	STRING	Information de l'objet d'identification ProductCode spécifique à un protocole
protocolName	STRING	Nom du protocole de communication pris en charge. Ce type de données doit comporter l'information "protocol_Modbus"
protocolSpecificName	énumération (ProductName ModelName VendorURL UserApplicationName)	Nom spécifique à un protocole de bus de terrain des informations pour une seule identification
value	STRING	Information relative à une seule identification
vendorName	STRING	Information de l'objet d'identification VendorName spécifique à un protocole

Tableau 97 – Types de données structurés pour l'information relative au dispositif

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
IdManufacturer	STRUCT			Identification du fabricant
	vendorName	M	[1..1]	
IdSoftwareRevision	STRUCT			Version correspondante à l'outil du dispositif physique – Version du micrologiciel («firmware»)
	majorMinorRevision	M	[1..1]	
IdTypeID	STRUCT			Information pour l'identification du type de dispositif d'un seul dispositif
	productCode	M	[1..1]	
IdValue	STRUCT			Nom du FDT, nom spécifique à un protocole et valeur pour un paramètre d'identification sans information sémantique pour l'Application cadre
	name	M	[1..1]	
	protocolSpecificName	M	[1..1]	
	value	M	[1..1]	
IdValues	STRUCT			Ensemble d'objets à plusieurs identifications
	IdValue	O	[0..*]	
ModbusDevice	STRUCT			Information pour l'identification du dispositif
	schemaVersion	O	[0..1]	
	protocolName	M	[1..1]	
	choice.of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	

12.3 Types de données pour l'identification lors du balayage

Les types de données décrits dans ce paragraphe sont utilisés dans le service suivant:

- Balayage.

Ce paragraphe décrit les informations utilisées pour l'identification lors du balayage spécifique à un protocole. Pour une description détaillée des paramètres d'identification lors du balayage, voir Tableau 98 et Tableau 99.

Les types de données décrits dans ce paragraphe sont définis pour l'espace de noms suivant.
Espace de noms: mbscan

Tableau 98 – Types de données simples pour l'identification lors du balayage

Type de données	Définition	Description
configuredState	enumeration (configuredAndPhysically Available configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	Doit contenir la valeur "notApplicable"
resultState	enumeration (provisional final error)	État actuel des informations transmises lors du balayage

Tableau 99 – Types de données structurés pour l'identification lors du balayage

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
IdAddress	STRUCT			Information d'adresse du dispositif cible
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
IdBusProtocol	STRUCT			Information relative au protocole utilisé
	modbusident: protocolName	M	[1..1]	
IdManufacturer	STRUCT			Identification du fabricant
	modbusident:vendorName	M	[1..1]	
ScanIdentification	STRUCT			Information pour l'identification lors du balayage d'un seul dispositif
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	
ScanIdentifications	STRUCT			Ensemble d'informations pour le balayage de plusieurs dispositifs
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	
IdSoftwareRevision	STRUCT			Version correspondante à l'outil du dispositif physique – Version du micrologiciel («firmware»)
	modbusident: majorMinorRevision	M	[1..1]	
IdTypeID	STRUCT			Information pour l'identification du type de

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
				dispositif d'un seul dispositif
	modbusident:productCode	M	[1..1]	
IdValue	STRUCT			Nom du FDT, nom spécifique à un protocole et valeur pour un paramètre d'identification sans information sémantique pour l'Application cadre
	modbusident:name	M	[1..1]	
	modbusident:value	M	[1..1]	
	modbusident:protocolSpecificName	M	[1..1]	
IdValues	STRUCT			Ensemble d'objets à plusieurs identifications
	IdValue	O	[0..*]	

12.4 Types de données pour l'identification du type de dispositif – fourni par le DTM

Les types de données décrits dans cet article sont utilisés dans le service suivant:

- GetIdentificationInformation.

Ce paragraphe décrit les informations utilisées pour l'identification du type de dispositif spécifique à un protocole. Pour une description détaillée des paramètres d'identification du type de dispositif, voir Tableau 100.

Les types de données décrits dans ce paragraphe sont définis pour l'espace de noms suivant.
Espace de noms: mbdevice

Tableau 100 – Types de données pour l'identification du type de dispositif

Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
DeviceIdentification	STRUCT			Information relative à l'identification d'un seul dispositif
	IdBusProtocol	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	
	modbusident: idDTMSupportLevel	M	[1..1]	
DeviceIdentifications	STRUCT			Ensemble d'informations pour l'identification de plusieurs dispositifs
	DeviceIdentification	M	[1..*]	
IdBusProtocol	STRUCT			Information relative au protocole utilisé
	modbusident: protocolName	M	[1..1]	
IdManufacturer	STRUCT			Identification du fabricant
	modbusident:RegExpr	O	[0..*]	
	modbusident:vendorName	M	[1..1]	
IdSoftwareRevision	STRUCT			Version correspondante à l'outil du dispositif physique – Version du micrologiciel («firmware»)
	modbusident: majorMinorRevision	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
IdTypeID	STRUCT			Information pour l'identification du type de dispositif d'un seul dispositif
	modbusident:productCode	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
IdValue	STRUCT			Nom du FDT, nom spécifique à un protocole et valeur pour un paramètre d'identification sans information sémantique pour l'Application cadre
	modbusident: protocolSpecificName	M	[1..1]	
	modbusident:name	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
	modbusident:value	O	[0..1]	
IdValues	STRUCT			Ensemble d'objets à plusieurs identifications
	IdValue	O	[0..*]	

12.5 Mise en correspondance des objets d'identification du dispositif spécifique à un protocole avec les types de données du FDT

L'identification du dispositif spécifique à un protocole dans le FDT est basée sur les informations pouvant être extraites d'un dispositif par le service pour la lecture de l'identification du dispositif spécifique à un protocole (0x2B/0xE). Pour cette raison, il est requis que les dispositifs Modbus prennent en charge le service pour la lecture de l'identification du dispositif pour tous les objets de la catégorie d'identification de base des dispositifs. Afin de clairement identifier un dispositif dans un FDT, il est recommandé pour les dispositifs Modbus de prendre en charge le service pour la lecture de l'identification du dispositif pour les objets de la catégorie d'identification ordinaire des dispositifs.

La description suivante de la manière dont les informations d'identification du dispositif spécifique à un protocole sont mises en correspondance avec les types de données du FDT s'applique

- à l'identification du type de dispositif du FDT (présenté en 12.4),
- et au balayage du FDT (présenté en 12.2).

Toutes les informations fournies par les objets d'identification spécifique à un protocole de la catégorie d'identification du dispositif de base doivent être mises en correspondance avec les types de données comme décrit dans le Tableau 101. Toutes les informations fournies par les objets d'identification spécifique à un protocole de la catégorie d'identification du dispositif régulier doivent être mises en correspondance avec le type de données de la valeur et avec le type de données structuré IdValue, ensemble avec les informations supplémentaires décrites dans le Tableau 102.

Tableau 101 – Mise en correspondance des objets d'identification spécifique à un protocole dans le FDT

Information du dispositif spécifique à un protocole				Information du FDT spécifique à un protocole		
Identifiant de l'objet	Nom de l'objet	Exigence pour Modbus	Catégorie	Type de données pour la sémantique	Type de données	Exigence pour FDT
0x00	VendorName	Obligatoire	De base	IdManufacturer	vendorName	Obligatoire
0x01	ProductCode	Obligatoire		IdTypeID	productCode	Obligatoire
0x02	MajorMinorRevision	Obligatoire		IdSoftwareRevision	majorMinorRevision	Obligatoire
0x03	VendorUrl	Facultative	Ordinaire	IdValue	value	Recommandée
0x04	ProductName	Facultative		IdValue	value	Recommandée
0x05	ModelName	Facultative		IdValue	value	Recommandée
0x06	UserApplicationName	Facultative		IdValue	value	Recommandée

Tableau 102 – Informations supplémentaires pour les objets d'identification facultatifs

Information du dispositif spécifique à un protocole		Information du FDT spécifique à un protocole		
Identifiant de l'objet	Nom de l'objet	Élément de la sémantique	Nom	protocolSpecificName
0x03	VendorUrl	IdValue	VendorURL	VendorURL
0x04	ProductName	IdValue	ProductName	ProductName
0x05	ModelName	IdValue	ModelName	ModelName
0x06	UserApplicationName	IdValue	UserApplicationName	UserApplicationName

Bibliographie

- [1] *MODBUS Application Protocol Specification V1.1a*, Modbus IDA, 4 juin 2004 (disponible en anglais uniquement)
 - [2] *MODBUS over Serial Line Specification & Implementation guide V1.0*, Modbus IDA, 12 décembre 2002 (disponible en anglais uniquement)
 - [3] *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0a*, Modbus IDA, 4 juin 2004
 - [4] IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description* (disponible en anglais uniquement)
 - [5] ISO/IEC 19501:2005, *Technologie de l'information – Traitement distribué ouvert – Langage de Modélisation Unifié (UML) Version 1.4.2*
-

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

FINAL VERSION

VERSION FINALE



**Field device tool (FDT) Interface specification –
Part 315: Communication profile integration – IEC 61784 CPF 15**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 315: Intégration des profils de communication – IEC 61784 CPF 15**

CONTENTS

FOREWORD.....	6
INTRODUCTION.....	8
1 Scope	9
2 Normative references	9
3 Terms, definitions, symbols, abbreviated terms and conventions	10
3.1 Terms and definitions	10
3.2 Abbreviated terms	10
3.3 Conventions	10
3.3.1 Data type names and references to data types	10
3.3.2 Vocabulary for requirements.....	10
4 Bus category	10
5 Access to instance and device data	12
5.1 Process Channel objects provided by DTM.....	12
5.2 DTM services to access instance and device data	12
6 Protocol specific behavior	12
6.1 General.....	12
6.2 Broadcasting	13
6.3 Unconfirmed private Modbus request	15
8 Protocol specific common data types	17
8.1 General.....	17
8.2 Address information.....	18
7 Protocol specific usage of general data types.....	17
9 Network management data types	18
10 Communication data types.....	19
10.1 General.....	19
10.2 Connection management data types.....	19
10.3 Transaction service specific data types	20
10.3.1 General	20
10.3.2 Data item addressing	20
10.3.3 Read coils transaction service.....	21
10.3.4 Read discrete inputs transaction service.....	22
10.3.5 Read holding registers transaction service.....	23
10.3.6 Read input registers transaction service	24
10.3.7 Write single coil transaction service.....	25
10.3.8 Write single register transaction service	26
10.3.9 Read exception status transaction service	27
10.3.10Diagnostics transaction service	27
10.3.11Get Comm event counter transaction service	28
10.3.12Get Comm event log transaction service.....	29
10.3.13Write multiple coils transaction service	31
10.3.14Write multiple registers transaction service.....	31
10.3.15Report slave ID transaction service	32
10.3.16Read file record transaction service.....	33
10.3.17Write file record transaction service.....	35
10.3.18Mask write register transaction service	36

10.3.19	Read/write holding registers transaction service	37
10.3.20	Read FIFO queue transaction service	38
10.3.21	Encapsulated interface transport transaction service	39
10.3.22	Read device identification transaction service	40
10.3.23	Private Modbus transaction service	43
10.3.24	Unconfirmed private Modbus transaction service	44
10.3.25	Modbus exception response	45
11	Channel parameter data types	46
12	Device Identification	48
12.1	Common device type identification data types	48
12.2	Topology scan data types	49
12.3	Scan identification data types	50
12.4	Device type identification data types – provided by DTM	52
12.5	Mapping of protocol specific device identification objects to FDT data types	53
	Bibliography	55
	Figure 1 – Part 315 of the IEC 62453 series	8
	Figure 2 – Broadcast sequence with Modbus Serial Line Communication DTM	14
	Figure 3 – Broadcast sequence with Modbus TCP Communication DTM	14
	Figure 4 – Broadcast sequence without Modbus Communication DTM	15
	Figure 5 – Unconfirmed request with Modbus Serial Line Communication DTM	16
	Figure 6 – Unconfirmed request with Modbus TCP Communication DTM	16
	Figure 7 – Unconfirmed request without Modbus Communication DTM	17
	Figure 8 – Data item addressing	21
	Table 1 – Protocol identifiers	10
	Table 2 – Address information for broadcast mode	13
	Table 3 – Broadcast transaction requests	13
	Table 4 – Protocol specific usage of general FDT data types	17
	Table 5 – Simple address information data types	18
	Table 6 – Structured address information data types	18
	Table 7 – Structured network management data types	19
	Table 8 – Simple common communication data types	19
	Table 9 – Structured connection management service data types	20
	Table 10 – Simple ReadCoilsReq data types	21
	Table 11 – Structured ReadCoilsReq data types	21
	Table 12 – Simple ReadCoilsRsp data types	22
	Table 13 – Structured ReadCoilsRsp data types	22
	Table 14 – Simple ReadDiscreteInputsReq data types	22
	Table 15 – Structured ReadDiscreteInputsReq data types	22
	Table 16 – Simple ReadDiscreteInputsRsp data types	23
	Table 17 – Structured ReadDiscreteInputsRsp data types	23
	Table 18 – Simple ReadHoldingRegistersReq data types	23
	Table 19 – Structured ReadHoldingRegistersReq data types	23
	Table 20 – Simple ReadHoldingRegistersRsp data types	24

Table 21 – Structured ReadHoldingRegistersRsp data types.....	24
Table 22 – Simple ReadInputRegistersReq data types.....	24
Table 23 – Structured ReadInputRegistersReq data types	24
Table 24 – Simple ReadInputRegistersRsp data types	25
Table 25 – Structured ReadInputRegistersRsp data types.....	25
Table 26 – Simple WriteSingleCoilReq data types	25
Table 27 – Structured WriteSingleCoilReq data types	25
Table 28 – Structured WriteSingleCoilRsp data types	26
Table 29 – Simple WriteSingleRegisterReq data types.....	26
Table 30 – Structured WriteSingleRegisterReq data types	26
Table 31 – Structured WriteSingleRegisterRsp data types	26
Table 32 – Structured ReadExceptionStatusReq data types.....	27
Table 33 – Simple ReadExceptionStatusRsp data types	27
Table 34 – Structured ReadExceptionStatusRsp data types	27
Table 35 – Simple diagnostics data types.....	28
Table 36 – Structured DiagnosticsReq data types.....	28
Table 37 – Structured DiagnosticsRsp data types	28
Table 38 – Structured GetCommEventCounterReq data types	29
Table 39 – Simple GetCommEventCounterRsp data types	29
Table 40 – Structured GetCommEventCounterRsp data types.....	29
Table 41 – Structured GetCommEventLogReq data types	30
Table 42 – Simple GetCommEventLogRsp data types	30
Table 43 – Structured GetCommEventLogRsp data types	30
Table 44 – Simple WriteMultipleCoilsReq data types	31
Table 45 – Structured WriteMultipleCoilsReq data types	31
Table 46 – Structured WriteMultipleCoilsRsp data types	31
Table 47 – Simple WriteMultipleRegistersReq data types.....	32
Table 48 – Structured WriteMultipleRegistersReq data types	32
Table 49 – Structured WriteMultipleRegistersRsp data types	32
Table 50 – Structured ReportSlaveIDReq data types	32
Table 51 – Simple ReportSlaveIDRsp data types.....	33
Table 52 – Structured ReportSlaveIDRsp data types.....	33
Table 53 – Simple ReadFileSubRequest data types.....	33
Table 54 – Structured ReadFileSubRequest data types	34
Table 55 – Structured ReadFileRecordReq data types	34
Table 56 – Simple ReadFileSubResponse data types	34
Table 57 – Structured ReadFileSubResponse data types.....	34
Table 58 – Structured ReadFileRecordRsp data types	35
Table 59 – Simple WriteFileSubRequest data types.....	35
Table 60 – Structured WriteFileSubRequest data types	35
Table 61 – Structured WriteFileRecordReq data types	36
Table 62 – Structured WriteFileRecordRsp data types	36
Table 63 – Simple MaskWriteRegisterReq data types	36

Table 64 – Structured MaskWriteRegisterReq data types.....	37
Table 65 – Structured MaskWriteRegisterRsp data types.....	37
Table 66 – Simple ReadWriteRegistersReq data types	37
Table 67 – Structured ReadWriteRegistersReq data types	38
Table 68 – Simple ReadWriteRegistersRsp data types.....	38
Table 69 – Structured ReadWriteRegistersRsp data types	38
Table 70 – Simple ReadFifoQueueReq data types	38
Table 71 – Structured ReadFifoQueueReq data types.....	38
Table 72 – Simple ReadFifoQueueRsp data types	39
Table 73 – Structured ReadFifoQueueRsp data types.....	39
Table 74 – Simple EncapsulatedInterfaceTransportReq data types	39
Table 75 – Structured EncapsulatedInterfaceTransportReq data types	40
Table 76 – Simple EncapsulatedInterfaceTransportRsp data types	40
Table 77 – Structured EncapsulatedInterfaceTransportRsp data types.....	40
Table 78 – Simple ReadDeviceIdentificationReq data types	41
Table 79 – Structured ReadDeviceIdentificationReq data types.....	41
Table 80 – Simple IdentificationObject data types.....	41
Table 81 – Structured IdentificationObject data types	41
Table 82 – Simple ReadDeviceIdentificationRsp data types	42
Table 83 – Structured ReadDeviceIdentificationRsp data types.....	43
Table 84 – Simple PrivateModbusReq data types	43
Table 85 – Structured PrivateModbusReq data types.....	43
Table 86 – Simple PrivateModbusRsp data types	43
Table 87 – Structured PrivateModbusRsp data types	44
Table 88 – Simple UnconfirmedPrivateModbusReq data types	44
Table 89 – Structured UnconfirmedPrivateModbusReq data types.....	44
Table 90 – Structured UnconfirmedPrivateModbusRsp data types.....	44
Table 91 – Simple ModbusExceptionRsp data types	45
Table 92 – Structured ModbusExceptionRsp data types.....	45
Table 93 – Simple channel parameter data types.....	46
Table 94 – Structured channel parameter data types	47
Table 95 – Simple common identification data types	48
Table 96 – Simple device information data types	49
Table 97 – Structured device information data types.....	50
Table 98 – Simple scan identification data types.....	51
Table 99 – Structured scan identification data types	51
Table 100 – Structured device type identification data types	52
Table 101 – Mapping of protocol specific identification objects in FDT	54
Table 102 – Additional information for optional identification objects	54
Table 103 – Physical layer identifiers for Modbus TCP.....	11
Table 104 – Physical layer identifiers for Modbus Serial	12

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

**Part 315: Communication profile integration –
IEC 61784 CPF 15**

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 62453-315 bears the edition number 1.1. It consists of the first edition (2009-07) [documents 65E/131/FDIS and 65E/144/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 62453-315 has 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 document using a colour printer.

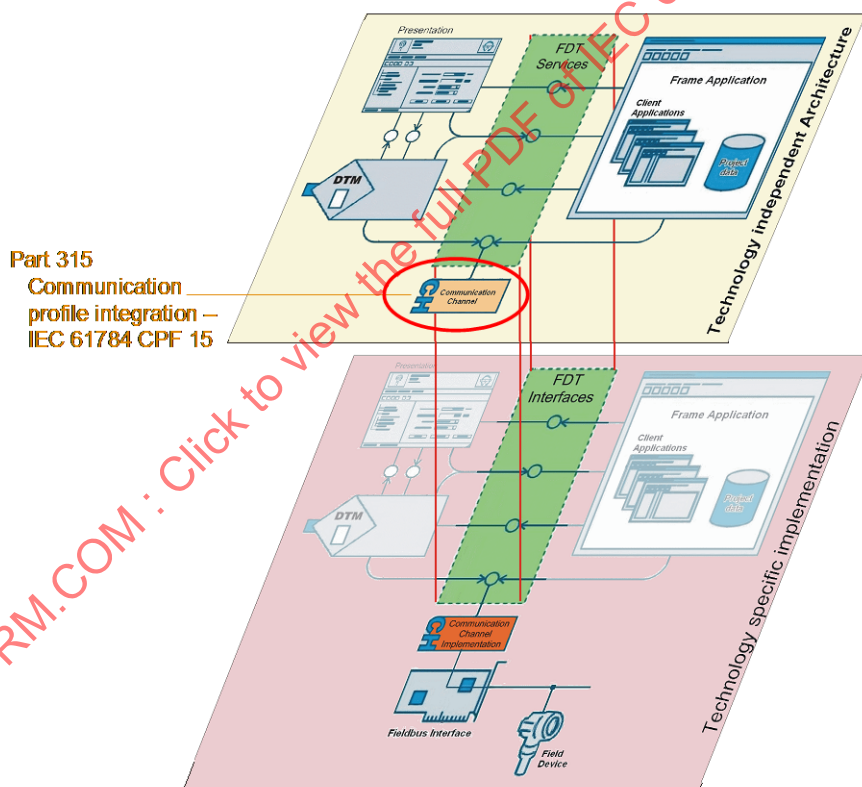
INTRODUCTION

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 62453-315 is aligned in the structure of the IEC 62453 series.



IEC 1136/09

Figure 1 – Part 315 of the IEC 62453 series

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 315: Communication profile integration – IEC 61784 CPF 15

1 Scope

Communication Profile Family 15 (commonly known as Modbus¹) defines communication profiles based on IEC 61158-5-15 and IEC 61158-6-15. The basic profile CP 15/1 (Modbus TCP) is defined in IEC 61784-1. An additional communication profile (Modbus Serial Line) is defined in [2].

This part of the IEC 62453 provides information for integrating Modbus TCP® and Modbus Serial Line® protocol support into FDT based systems.

NOTE This part of IEC 62453 only specifies the mapping of Modbus parameters to FDT data types. For restrictions of protocol specific parameters concerning allowed values and concerning limitations of arrays used in the definition of FDT data types, refer to IEC 61158-5-15 and the MODBUS Application Protocol Specification.

2 Normative references

The following referenced documents are indispensable for the application of this document. 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, *Programmable controllers – Part 3: Programming languages*

IEC 61158-5-15, *Industrial communication networks – Fieldbus specifications – Part 5-15: Application layer service definition – Type 15 elements*

IEC 61158-6-15, *Industrial communication networks – Fieldbus specifications – Part 6-20: Application layer protocol specification – Type 15 elements*

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

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

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*

RFC 791, *Internet Protocol* (available at <<http://www.ietf.org/rfc/rfc0791.txt>>)

1) Modbus is the trademark of Schneider Automation Inc. It is registered in the United States of America. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trademark Modbus. Use of the trademark Modbus requires permission from Schneider Automation Inc.

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, IEC 62453-2 and the following apply.

3.2 Abbreviated terms

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

ASCII	American Standard Code for Information Interchange
DTM	Device Type Manager
FA	Frame Application
IP	Internet Protocol
RFC	Request For comment
TCP	Transmission Control Protocol

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

4 Bus category

The Modbus protocol is identified in the protocolId element of the structured data type 'fdt:BusCategory' by the following unique identifiers (see Table 1).

Table 1 – Protocol identifiers

Identifier value	ProtocolId name	Description
59629a40-285f-11db-a98b-0800200c9a66	'Modbus over Serial Line'	Modbus over Serial Line
59629a41-285f-11db-a98b-0800200c9a66	'Modbus over TCP'	Modbus over TCP

Modbus TCP is using the following unique identifiers in physicalLayer members within PhysicalLayer data type (see Table 103):

Table 103 – Physical layer identifiers for Modbus TCP

Identifier value	Description
307DD810-C010-11DB-90E7-0002B3ECDCBE	100BaseTXFD (default for Media Type Copper)
307DD812-C010-11DB-90E7-0002B3ECDCBE	100BaseFXFD (default for Media Type Fiber Optic)
307DD813-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD816-C010-11DB-90E7-0002B3ECDCBE	1000BaseXFD
307DD818-C010-11DB-90E7-0002B3ECDCBE	1000BaseLXFD
307DD81A-C010-11DB-90E7-0002B3ECDCBE	1000BaseSXFD
307DD81C-C010-11DB-90E7-0002B3ECDCBE	1000BaseTFD
307DD81D-C010-11DB-90E7-0002B3ECDCBE	10GbaseFX
307DD81E-C010-11DB-90E7-0002B3ECDCBE	10GbaseLX4
307DD81F-C010-11DB-90E7-0002B3ECDCBE	10GBaseR
307DD820-C010-11DB-90E7-0002B3ECDCBE	10GbaseER
307DD821-C010-11DB-90E7-0002B3ECDCBE	10GbaseLR
307DD822-C010-11DB-90E7-0002B3ECDCBE	10GbaseSR
307DD823-C010-11DB-90E7-0002B3ECDCBE	10GbaseW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseEW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseLW
307DD825-C010-11DB-90E7-0002B3ECDCBE	10GbaseSW
307DD826-C010-11DB-90E7-0002B3ECDCBE	10GbaseCX4
307DD827-C010-11DB-90E7-0002B3ECDCBE	2BaseTL
307DD828-C010-11DB-90E7-0002B3ECDCBE	10PassTS
307DD829-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10D
307DD82A-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10U
307DD82B-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD82C-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10D
307DD82D-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10U
307DD82F-C010-11DB-90E7-0002B3ECDCBE	1000BaseLX10
307DD830-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10D
307DD831-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10U
307DD832-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20D
307DD833-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20U
307DD834-C010-11DB-90E7-0002B3ECDCBE	10GBaseT or 100BasePXFD
307DD835-C010-11DB-90E7-0002B3ECDCBE	10GBaseLRM
307DD836-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX
307DD837-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX4
307DD838-C010-11DB-90E7-0002B3ECDCBE	1000BaseKR
307DD839-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXD1
307DD83A-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR2
307DD83B-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR3
307DD83C-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXU1
307DD83D-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR2U
307DD83E-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR3U
307DD83F-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD1
307DD840-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD2

Identifier value	Description
307DD841-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD3
307DD842-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU1
307DD843-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU3
307DD844-C010-11DB-90E7-0002B3ECDCBE	40GbaseKR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseCR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseSR4
307DD846-C010-11DB-90E7-0002B3ECDCBE	40GbaseFR
307DD847-C010-11DB-90E7-0002B3ECDCBE	40GbaseLR4
307DD848-C010-11DB-90E7-0002B3ECDCBE	100GbaseCR10
307DD849-C010-11DB-90E7-0002B3ECDCBE	100GbaseSR10
307DD84A-C010-11DB-90E7-0002B3ECDCBE	100GbaseLR4
307DD84B-C010-11DB-90E7-0002B3ECDCBE	100GbaseER4
307DD84C-C010-11DB-90E7-0002B3ECDCBE	100BasePXFD
307DD84D-C010-11DB-90E7-0002B3ECDCBE	Radio Communication
307DD84E-C010-11DB-90E7-0002B3ECDCBE	Speed of 100 Mbit/s (and more) and full duplexity

Modbus Serial is using the following unique identifiers in physicalLayer members within PhysicalLayer data type (see Table 104):

Table 104 – Physical layer identifiers for Modbus Serial

Identifier value	Description
C0458028-F240-45A5-8664-70DC84FDC6FE	RS-232
3BF008DC-5A44-4220-8C3E-3C46A589A0B4	RS-422
036D1591-387B-11D4-86E1-00E0987270B9	RS-485

The DataLinkLayer property is not applicable for Modbus and has to be set to null.

5 Access to instance and device data

5.1 Process Channel objects provided by DTM

For Modbus no minimum set of parameters is defined which must be provided by a DTM. If a DTM provides process values, the process variables should be modeled as Process Channels.

5.2 DTM services to access instance and device data

Not applicable.

6 Protocol specific behavior

6.1 General

This clause specifies how unconfirmed Modbus requests shall be handled in FDT. Two types of unconfirmed Modbus requests are supported in FDT:

- broadcasts;
- unconfirmed private Modbus requests.

6.2 Broadcasting

In broadcast mode, a DTM can send a Modbus request to all devices connected to the bus. This mode is only supported for devices, which are connected via Modbus Serial Line. The connection can either be a direct connection or a connection via a gateway. The broadcast mode must be initiated by a fdtmodbus:ConnectRequest with the slave address of the target device set to 0 (slaveAddress=0). For more detailed information about the address information which has to be provided for broadcast mode, see Table 2.

Table 2 – Address information for broadcast mode

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ModbusSerial	STRUCT			Address information for Modbus Serial Line
	slaveAddress	M	[1..1]	Shall be 0 for broadcast mode
ModbusTCP	STRUCT			Address information for Modbus TCP:
	slaveAddress	M	[1..1]	Shall be 0 for broadcast mode
	tcpAddress	M	[1..1]	IP address of the Modbus TCP/ Modbus Serial Line gateway
	tcpPort	O	[0..1]	Port for Modbus TCP connection (if no port information is provided, the default port 502 shall be used)

Because in broadcast mode no response will be returned by the device, the broadcast mode shall be used only with the transaction requests shown in Table 3:

Table 3 – Broadcast transaction requests

Broadcast transaction requests	Restrictions
WriteSingleCoilReq	None
WriteSingleRegisterReq	None
DiagnosticsReq	This transaction request shall be used in broadcast mode only with the following sub-functions: - 0x01: Restart Communication Option - 0x03: Change ASCII Input Delimiter - 0x04: Force Listen Only Mode - 0x0A: Clear Counters and Diagnostic Register - 0x14: Clear Overrun Counter and Flag
WriteMultipleCoilsReq	None
WriteMultipleRegistersReq	None
WriteFileRecordReq	None
MaskWriteRegisterReq	None
PrivateModbusReq	Shall be used only with private services where no response is required from the device

Although in broadcast mode no response will be returned by the target devices a transaction response, which corresponds to the transaction request shall be generated. This transaction response shall be generated in order to inform the DTM that the broadcast request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it has to provide this generated transaction response (see Figure 2 and Figure 3).

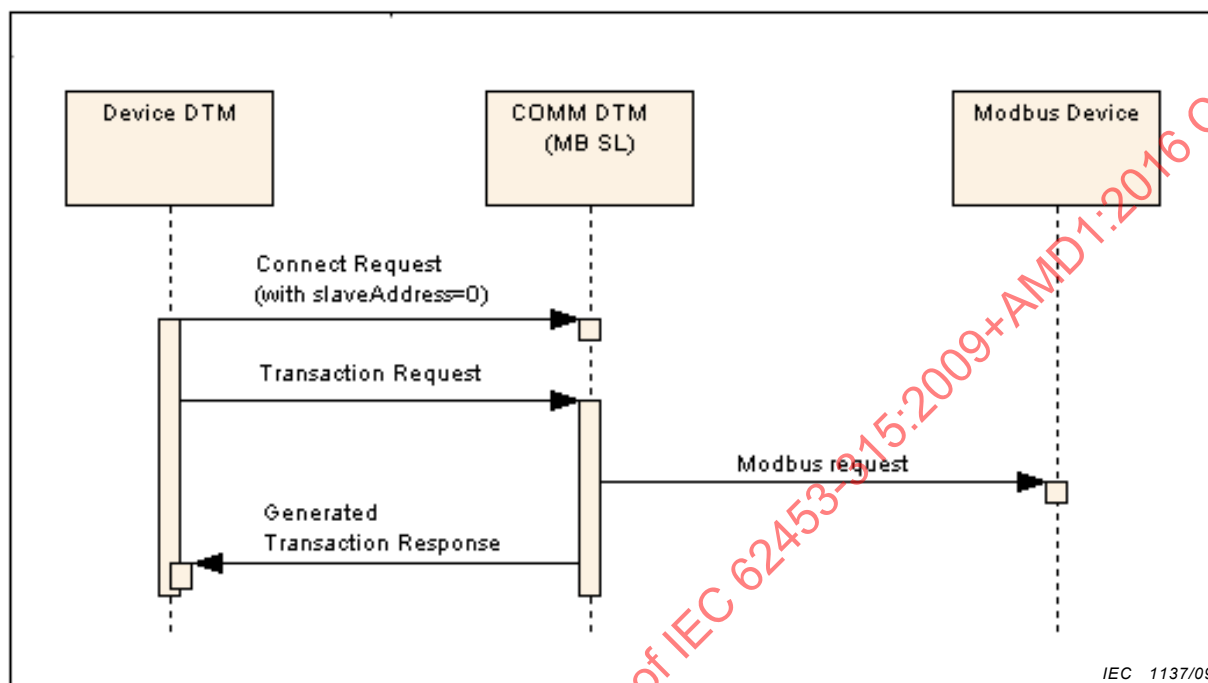


Figure 2 – Broadcast sequence with Modbus Serial Line Communication DTM

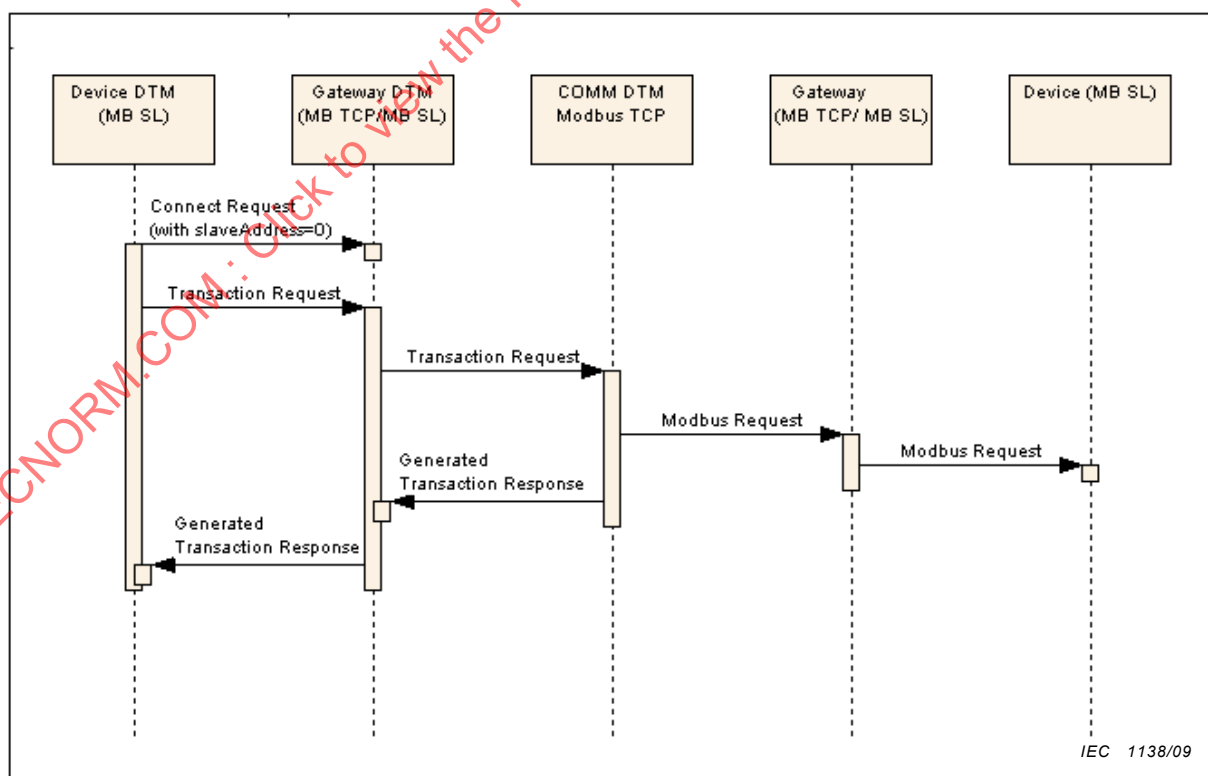


Figure 3 – Broadcast sequence with Modbus TCP Communication DTM

If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, has to provide this transaction response (see Figure 4).

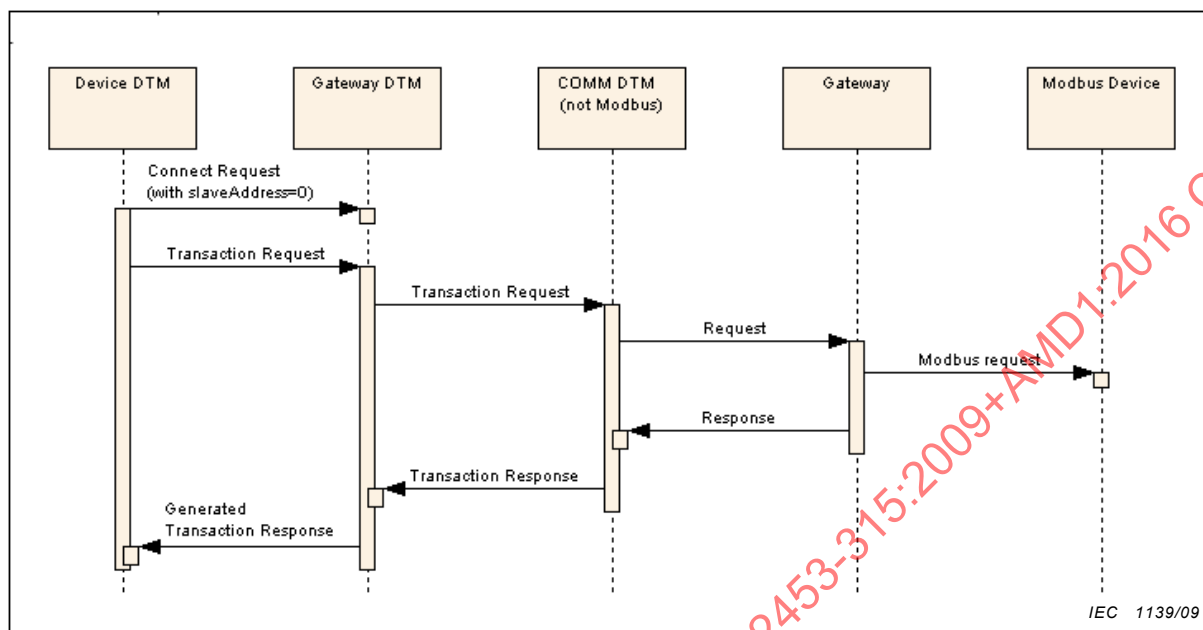


Figure 4 – Broadcast sequence without Modbus Communication DTM

6.3 Unconfirmed private Modbus request

This transaction request can be used to send any unconfirmed private Modbus request. Although no response will be returned by the target device, a transaction response, which corresponds to the transaction request, shall be generated. This transaction response shall be generated in order to inform the DTM that the unconfirmed request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it has to provide this generated transaction response (see Figure 5 and Figure 6).

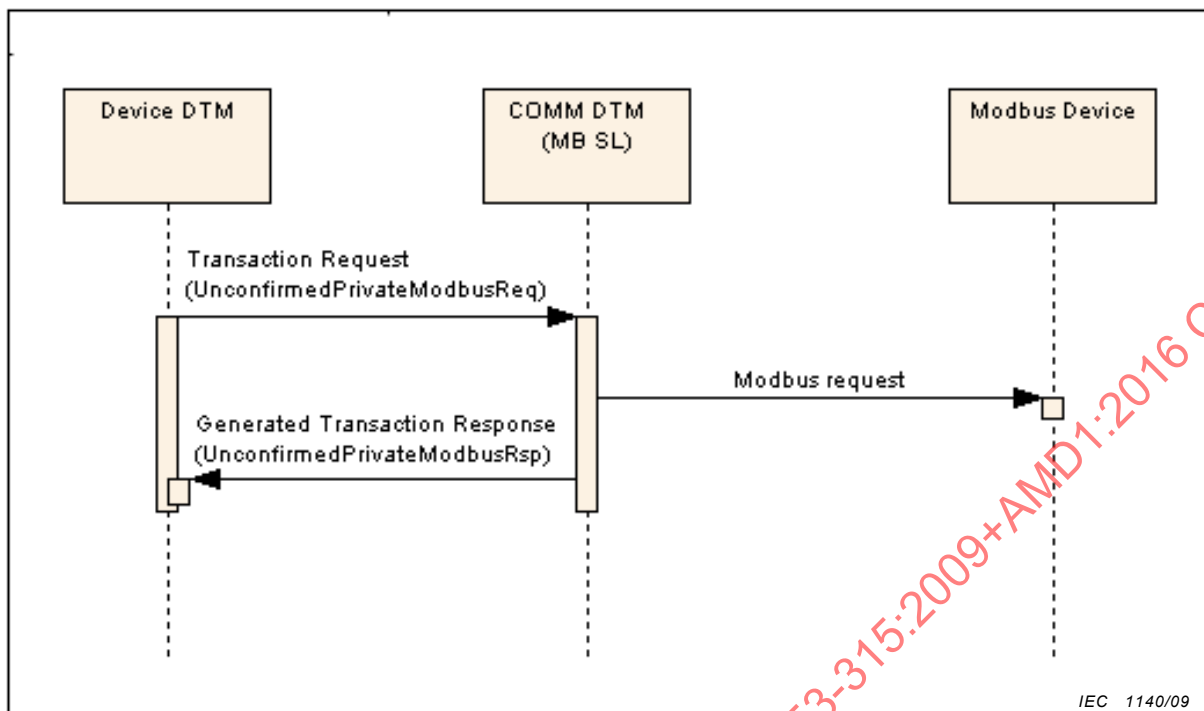


Figure 5 – Unconfirmed request with Modbus Serial Line Communication DTM

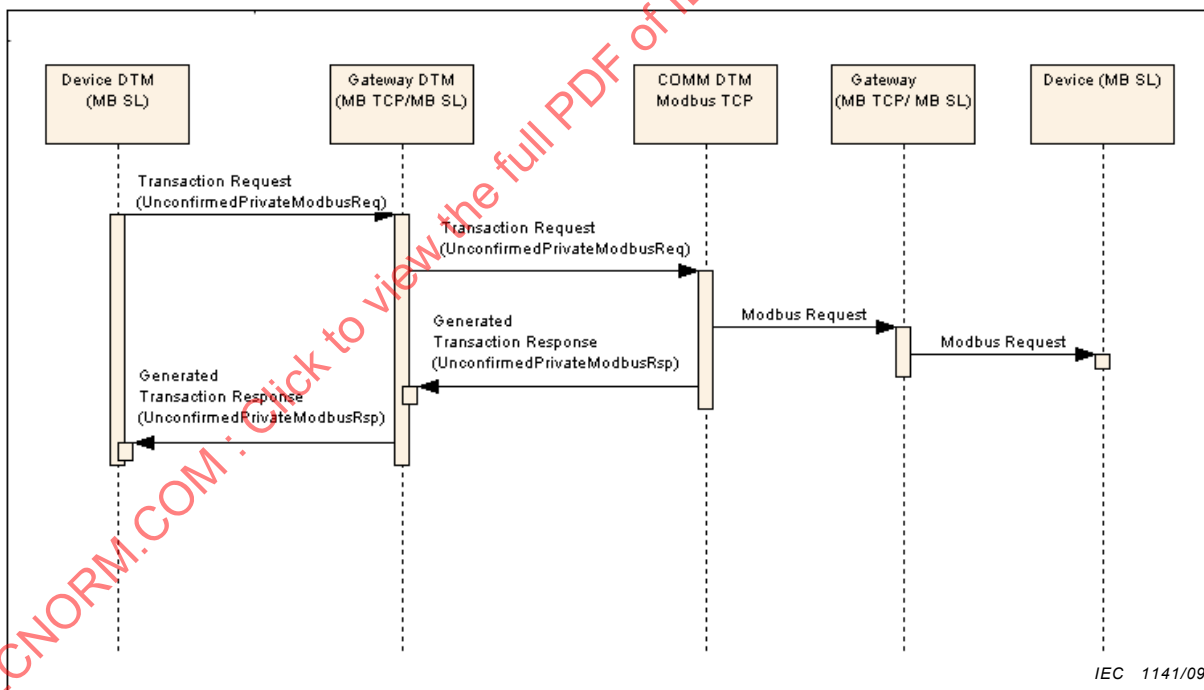


Figure 6 – Unconfirmed request with Modbus TCP Communication DTM

If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, has to provide this transaction response (see Figure 7).

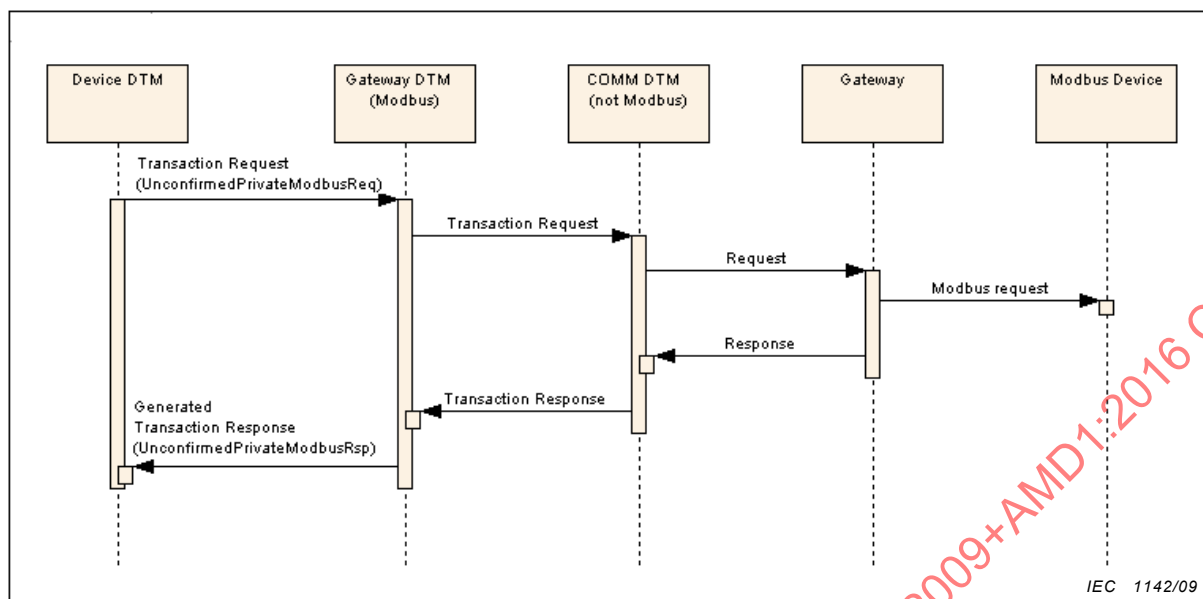


Figure 7 – Unconfirmed request without Modbus Communication DTM

7 Protocol specific usage of general data types

The following table (Table 4) shows how general data types, defined in IEC 62453-2 within the namespace 'fdt', are used with IEC 61784 CPF 15 devices.

Table 4 – Protocol specific usage of general FDT data types

Data type	Description for use
fdt:dataType	Identifier for the FDT data type of the channel object
fdt:busCategory	See Clause 4
deviceTypeInfoInformation	Modbus device description in form of human readable string, which can be provided with a DTM
deviceTypeInfoInformationPath	Path to file which contains the Modbus device description if available
fdt:readAccess	Specifies whether the value can be read from the device: “0”, if reading access is not allowed “1”, if reading access is allowed
fdt:systemTag	Unique identifier of a device instance within a project of a Frame Application
fdt:writeAccess	Specifies whether the value can be written to the device: “0”, if writing access is not allowed “1”, if writing access is allowed

8 Protocol specific common data types

8.1 General

This clause specifies protocol specific common data types which are used in the definition of other data types.

8.2 Address information

This subclause specifies the structure and the semantics of the protocol specific address information in FDT, which is used on the different underlying communication layers for Modbus TCP and Modbus Serial Line.

For a detailed description of how the address information is mapped to data types, see Table 5 and Table 6.

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

Table 5 – Simple address information data types

Data type	Definition	Description
slaveAddress	USINT	Slave address of a Modbus device
tcpAddress	STRING	String representation of the IP address as described in RFC 791 Internet Protocol (IP)
tcpPort	UINT	Port for Modbus TCP connection (if no port information is provided the default port 502 shall be used)

Table 6 – Structured address information data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ModbusSerial	STRUCT			Address information for Modbus Serial Line
	slaveAddress	M	[1..1]	
ModbusTCP	STRUCT			Address information for Modbus TCP: IP address of the target Modbus TCP server Slave address of the target device if the Modbus TCP server provides routing services over Modbus Serial Line
	slaveAddress	O	[0..1]	
	tcpAddress	M	[1..1]	
	tcpPort	O	[0..1]	

9 Network management data types

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

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

This clause describes the information, which has to be provided for Modbus devices, in the UserDefinedBus data type (see Table 7).

Table 7 – Structured network management data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UserDefinedBus	STRUCT			The UserDefinedBus data type contains the address information of the device. It is required that the UserDefinedBus data type contains at least one address information data type of the target device
	mb:ModbusSerial	O	[0..*]	
	mb:ModbusTCP	O	[0..*]	

Some devices and their modules require more than one address on the bus. It is possible that the UserDefinedBus data type may contain several address-elements. In this case, the first address element shall contain the information about the principal address of the device.

10 Communication data types

10.1 General

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

- connect service;
- disconnect service;
- transaction service.

This clause describes the structure and the semantics of the protocol specific communication related data, which is exchanged between a DTM and its parent component.

For a detailed description of the common communication data types used in these services, see Table 8.

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

Table 8 – Simple common communication data types

Data type	Definition	Description
communicationReference	UUID	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

10.2 Connection management data types

For a detailed description of the information used in the connect service and in the disconnect service see Table 9.

Table 9 – Structured connection management service data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
Abort	STRUCT			If the Abort command is sent to the next communication component or to the connected device, it terminates all pending requests and returns without waiting for a result. The termination of the connection will not be confirmed
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Describes the communication request: – to establish a communication link to a Modbus TCP server device; – to establish a communication link to a Modbus Serial Line slave device. The request contains the address information of the target device
	fdt:systemTag	M	[1..1]	
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
ConnectResponse	STRUCT			Describes the communication response to the connect request
	communicationReference	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request to release a communication link
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response to a disconnect request
	communicationReference	M	[1..1]	

10.3 Transaction service specific data types

10.3.1 General

This subclause describes the information which is used in the transaction services.

10.3.2 Data item addressing

This document specifies the mapping of Modbus parameters to FDT data types. For this reason, all address information provided as parameters and concerning the Modbus data items

- coils,
- discrete inputs,
- holding registers and
- input registers,

shall be the same information as will be later sent on the bus (see Figure 8).

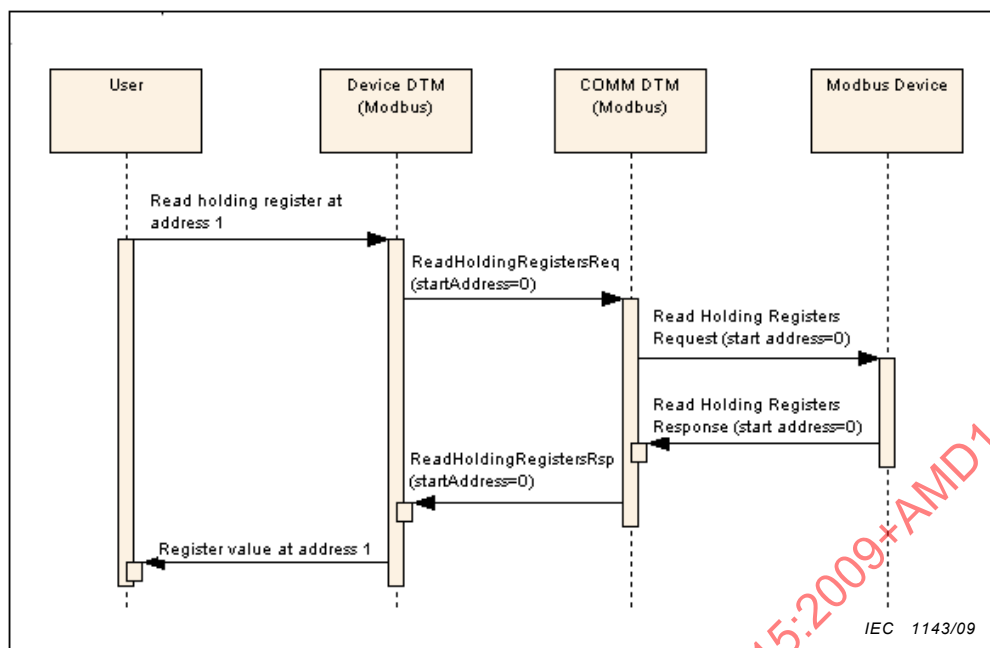


Figure 8 – Data item addressing

10.3.3 Read coils transaction service

The parameters of the transaction service, which is equivalent to the Read Coils service defined in IEC 61158-5-15, are described in Table 10, Table 11, Table 12 and Table 13.

Table 10 – Simple ReadCoilsReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first coil to be read	Address of first coil
quantity	UINT	Number of coils to be read	Quantity of coils

Table 11 – Structured ReadCoilsReq data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ReadCoilsReq	STRUCT			Communication request equivalent to the Read Coils service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Table 12 – Simple ReadCoilsRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
multipleCoilValues	STRING	ASCII string with each coil state coded in one character, where the first character in the string represents the first coil to be read: – “0” = FALSE or “OFF” – “1” = TRUE or “ON”	Data

Table 13 – Structured ReadCoilsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadCoilsRsp	STRUCT			Communication response equivalent to the Read Coils service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	multipleCoilValues	M	[1..1]	

10.3.4 Read discrete inputs transaction service

The parameters of the transaction service, which is equivalent to the Read Discretes service defined in IEC 61158-5-15, are described in Table 14, Table 15, Table 16 and Table 17.

Table 14 – Simple ReadDiscreteInputsReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first discrete input to be read	Address of first discrete
quantity	UINT	Number of discrete inputs to be read	Quantity of discretes

Table 15 – Structured ReadDiscreteInputsReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDiscreteInputsReq	STRUCT			Communication request equivalent to the Read Discretes service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Table 16 – Simple ReadDiscreteInputsRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
discreteInputsStatus	STRING	ASCII string with each discrete input state coded in one character, where the first character in the string represents the first discrete input to be read:: – “0”= FALSE or “OFF” – “1”= TRUE or “ON”	Data

Table 17 – Structured ReadDiscreteInputsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDiscreteInputsRsp	STRUCT			Communication response equivalent to the Read Discretes service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	discreteInputsStatus	M	[1..1]	

10.3.5 Read holding registers transaction service

The parameters of the transaction service, which is equivalent to the Read Holding Registers service defined in IEC 61158-5-15, are described in Table 18, Table 19, Table 20 and Table 21.

Table 18 – Simple ReadHoldingRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first holding register to be read	Address of first holding register to read
quantity	UINT	Number of holding registers to be read	Quantity of holding registers to read

Table 19 – Structured ReadHoldingRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadHoldingRegistersReq	STRUCT			Communication request equivalent to the Read Holding Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	startAddress	M	[1..1]	
	quantity	M	[1..1]	

Table 20 – Simple ReadHoldingRegistersRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
registerValues	ARRAY OF WORD	Read holding register values	Data

Table 21 – Structured ReadHoldingRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadHoldingRegistersRsp	STRUCT			Communication response equivalent to the Read Holding Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	registerValues	M	[1..1]	

10.3.6 Read input registers transaction service

The parameters of the transaction service, which is equivalent to the Read Input Registers service defined in IEC 61158-5-15, are described in Table 22, Table 23, Table 24 and Table 25.

Table 22 – Simple ReadInputRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
startAddress	UINT	Address of the first input register to be read	Address of first input register
quantity	UINT	Number of input registers to be read	Quantity of input registers

Table 23 – Structured ReadInputRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadInputRegistersReq	STRUCT			Communication request equivalent to the Read Input Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	quantity	M	[1..1]	
	startAddress	M	[1..1]	

Table 24 – Simple ReadInputRegistersRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
registerValues	ARRAY OF WORD	Read input register values	Data

Table 25 – Structured ReadInputRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadInputRegistersRsp	STRUCT			Communication response equivalent to the Read Input Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	registerValues	M	[1..1]	

10.3.7 Write single coil transaction service

The parameters of the transaction service, which is equivalent to the Write Single Coil service defined in IEC 61158-5-15, are described in Table 26, Table 27 and Table 28.

Table 26 – Simple WriteSingleCoilReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
outputAddress	UINT	Address of the coil to be forced	Address of first coil
singleCoilValue	BOOL	Coil state to be forced, with: – “0”=FALSE or “OFF” – “1”=TRUE or “ON”	Data single coil

Table 27 – Structured WriteSingleCoilReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleCoilReq	STRUCT			Communication request equivalent to the Write Single Coil service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	singleCoilValue	M	[1..1]	

Table 28 – Structured WriteSingleCoilRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleCoilRsp	STRUCT			Communication response equivalent to the Write Single Coil service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.8 Write single register transaction service

The parameters of the transaction service, which is equivalent to the Write Single Holding Register service defined in IEC 61158-5-15, are described in Table 29, Table 30 and Table 31.

Table 29 – Simple WriteSingleRegisterReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
outputAddress	UINT	Address of the holding register to be written	Address of first holding register to write
singleRegister	WORD	Value to be written to the register	Data

Table 30 – Structured WriteSingleRegisterReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleRegisterReq	STRUCT			Communication request equivalent to the Write Single Holding Register service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	singleRegister	M	[1..1]	

Table 31 – Structured WriteSingleRegisterRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteSingleRegisterRsp	STRUCT			Communication response equivalent to the Write Single Holding Register service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.9 Read exception status transaction service

The parameters of the transaction service, which is equivalent to the Read Exception Status service defined in the MODBUS Application Protocol specification, are described in Table 32, Table 33 and Table 34.

Table 32 – Structured ReadExceptionStatusReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadExceptionStatusReq	STRUCT			Communication request equivalent to the Read Exception Status service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line				

Table 33 – Simple ReadExceptionStatusRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
exceptionStatus	BYTE	Exception status of Modbus Serial Line device	Output Data

Table 34 – Structured ReadExceptionStatusRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadExceptionStatusRsp	STRUCT			Communication response equivalent to the Read Exception Status service response defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	exceptionStatus	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.10 Diagnostics transaction service

The parameters of the transaction service, which is equivalent to the Diagnostics service defined in the MODBUS Application Protocol specification, are described in Table 35, Table 36 and Table 37.

Table 35 – Simple diagnostics data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
diagnosticsData	ARRAY OF WORD	Required data, depending on the sub-function type	Data
diagnosticsSubFct	UINT	Sub-function code	Sub-function

Table 36 – Structured DiagnosticsReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
DiagnosticsReq	STRUCT			Communication request equivalent to the Diagnostics service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	diagnosticsData	M	[1..1]	
	diagnosticsSubFct	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 37 – Structured DiagnosticsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
DiagnosticsRsp	STRUCT			Communication response equivalent to the Diagnostics service response defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	diagnosticsData	M	[1..1]	
	diagnosticsSubFct	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.11 Get Comm event counter transaction service

The parameters of the transaction service, which is equivalent to the Get Comm Event Counter service defined in the MODBUS Application Protocol specification, are described in Table 38, Table 39 and Table 40.

Table 38 – Structured GetCommEventCounterReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventCounterReq	STRUCT			Communication request equivalent to the Get Comm Event Counter service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 39 – Simple GetCommEventCounterRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
commStatus	WORD	Two-byte status information. The status information will be 0xFFFF if a previously-issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000.	Status
eventCount	UINT	Number of events counted by the device	Event Count

Table 40 – Structured GetCommEventCounterRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventCounterRsp	STRUCT			Communication response equivalent to the Get Comm Event Counter service response defined in the MODBUS Application Protocol specification
	commStatus	M	[1..1]	
	communicationReference	M	[1..1]	
	eventCount	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.12 Get Comm event log transaction service

The parameters of the transaction service, which is equivalent to the Get Comm Event Log service defined in the MODBUS Application Protocol specification, are described in Table 41, Table 42 and Table 43.

Table 41 – Structured GetCommEventLogReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventLogReq	STRUCT			Communication request equivalent to the Get Comm Event Log service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 42 – Simple GetCommEventLogRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
commStatus	WORD	Two-byte status information. The status information will be 0xFFFF if a previously-issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000	Status
eventCount	UINT	Number of events counted by the device	Event Count
events	ARRAY OF BYTE	Each byte corresponds to the status of one MODBUS send or receive operation for the remote device	Events
messageCount	UINT	Quantity of messages processed by the remote device since its last restart, clear counters operation, or power-up	Message Count

Table 43 – Structured GetCommEventLogRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
GetCommEventCounter Rsp	STRUCT			Communication response equivalent to the Get Comm Event Log service response defined in the MODBUS Application Protocol specification
	commStatus	M	[1..1]	
	communicationReference	M	[1..1]	
	eventCount	M	[1..1]	
	events	O	[0..1]	
	messageCount	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.13 Write multiple coils transaction service

The parameters of the transaction service, which is equivalent to the Write Multiple Coils service defined in IEC 61158-5-15, are described in Table 44, Table 45 and Table 46.

Table 44 – Simple WriteMultipleCoilsReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
multipleCoilValues	STRING	ASCII string with each coil state coded in one character, where the first character in the string represents the first coil to be written: – “0”=FALSE or “OFF” – “1”=TRUE or “ON”	Data
outputAddress	UINT	Address of the first coil to be forced	Address of first coil

Table 45 – Structured WriteMultipleCoilsReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleCoilsReq	STRUCT			Communication request equivalent to the Write Multiple Coils service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	multipleCoilValues	M	[1..1]	
	outputAddress	M	[1..1]	

Table 46 – Structured WriteMultipleCoilsRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleCoilsRsp	STRUCT			Communication response equivalent to the Write Multiple Coils service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.14 Write multiple registers transaction service

The parameters of the transaction service, which is equivalent to the Write Multiple Holding Registers service defined in IEC 61158-5-15, are described in Table 47, Table 48 and Table 49.

Table 47 – Simple WriteMultipleRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
outputAddress	UINT	Address of the first holding register to be written	Address of first holding register to write
registerValues	ARRAY OF WORD	Register values to be written	Data

Table 48 – Structured WriteMultipleRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleRegisters Req	STRUCT			Communication request equivalent to the Write Multiple Holding Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	outputAddress	M	[1..1]	
	registerValues	M	[1..1]	

Table 49 – Structured WriteMultipleRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteMultipleRegisters Rsp	STRUCT			Communication response equivalent to the Write Multiple Holding Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.15 Report slave ID transaction service

The parameters of the transaction service, which is equivalent to the Report Slave ID service defined in the MODBUS Application Protocol specification, are described in Table 50, Table 51 and Table 52.

Table 50 – Structured ReportSlaveIDReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReportSlaveIDReq	STRUCT			Communication request equivalent to the Report Slave ID service request defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

Table 51 – Simple ReportSlaveIDRsp data types

Data type	Definition	Description	Equivalent MODBUS Application protocol service parameter
data	ARRAY OF BYTE	This attribute contains the: - Slave ID, - the Run Indicator Status (0x00 or 0xFF) - and the additional device specific data in the same format and order as defined in the MODBUS Application Protocol Specification	Slave ID, Run Indicator Status Additional data

Table 52 – Structured ReportSlaveIDRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReportSlaveIDRsp	STRUCT			Communication response equivalent to the Report Slave ID service response defined in the MODBUS Application Protocol specification
	communicationReference	M	[1..1]	
	data	M	[1..1]	
NOTE According to the MODBUS Application Protocol Specification V1.1a this Modbus service can only be used on Modbus over Serial Line.				

10.3.16 Read file record transaction service

The parameters of the transaction service, which is equivalent to the Read File Record service defined in IEC 61158-5-15, are described in Table 53 to Table 58.

Table 53 – Simple ReadFileSubRequest data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fileNumber	UINT	File number	File number
quantity	UINT	Length of the record to be read as quantity of registers	Record length
recordNumber	UINT	Starting record number within the file	Record number
referenceType	USINT	Reference type	Reference type
NOTE According to IEC 61158-5-15 the reference type shall be specified as 0x06.			

Table 54 – Structured ReadFileSubRequest data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileSubRequest	STRUCT			Sub-request equivalent to the Read File Record sub-request element defined in IEC 61158-5-15
	fileNumber	M	[1..1]	
	quantity	M	[1..1]	
	recordNumber	M	[1..1]	
	referenceType	M	[1..1]	

Table 55 – Structured ReadFileRecordReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileRecordReq	STRUCT			Communication request equivalent to the Read File Record service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	ReadFileSubRequest	M	[1..*]	

Table 56 – Simple ReadFileSubResponse data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
recordData	ARRAY OF WORD	Record data	Record data array

Table 57 – Structured ReadFileSubResponse data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileSubResponse	STRUCT			Sub-response equivalent to the Read File Record sub-response element defined in IEC 61158-5-15
	recordData	M	[1..1]	

Table 58 – Structured ReadFileRecordRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFileRecordRsp	STRUCT			Communication response equivalent to the Read File Record service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	ReadFileSubResponse	M	[1..*]	

10.3.17 Write file record transaction service

The parameters of the transaction service, which is equivalent to the Write File Record service defined in IEC 61158-5-15, are described in Table 59 to Table 62.

Table 59 – Simple WriteFileSubRequest data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fileNumber	UINT	File number	File number
recordData	ARRAY OF WORD	Length of the record to be read as quantity of registers	Record data array
recordNumber	UINT	Starting record number within the file	Record number
referenceType	USINT	Reference type	Reference type
NOTE According to IEC 61158-5-15 the reference type shall be specified as 0x06.			

Table 60 – Structured WriteFileSubRequest data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteFileSubRequest	STRUCT			Sub-request equivalent to the Write File Record sub-request element defined in IEC 61158-5-15
	fileNumber	M	[1..1]	
	recordData	M	[1..1]	
	RecordNumber	M	[1..1]	
	ReferenceType	M	[1..1]	

Table 61 – Structured WriteFileRecordReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteFileRecordReq	STRUCT			Communication request equivalent to the Write File Record service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	WriteFileSubRequest	M	[1..*]	

Table 62 – Structured WriteFileRecordRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
WriteFileRecordRsp	STRUCT			Communication response equivalent to the Write File Record service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.18 Mask write register transaction service

The parameters of the transaction service, which is equivalent to the Mask Write Holding Register service defined in IEC 61158-5-15, are described in Table 63, Table 64 and Table 65.

Table 63 – Simple MaskWriteRegisterReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
andMask	WORD	AND mask	AND Mask
orMask	WORD	OR mask	OR Mask
referenceAddress	UINT	Address of the holding register to which the mask shall be applied	Address of first holding register to write

Table 64 – Structured MaskWriteRegisterReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
MaskWriteRegisterReq	STRUCT			Communication request equivalent to the Mask Write Holding Register service request defined in IEC 61158-5-15
	andMask	M	[1..1]	
	communicationReference	M	[1..1]	
	orMask	M	[1..1]	
	referenceAddress	M	[1..1]	

Table 65 – Structured MaskWriteRegisterRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
MaskWriteRegisterRsp	STRUCT			Communication response equivalent to the Mask Write Holding Register service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	

10.3.19 Read/write holding registers transaction service

The parameters of the transaction service, which is equivalent to the Read/Write Holding Registers service defined in IEC 61158-5-15, are described in Table 66, Table 67, Table 68 and Table 69.

Table 66 – Simple ReadWriteRegistersReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
readQuantity	UINT	Number of holding registers to be read	Quantity of holding registers to read
readStartAddress	UINT	Address of the first holding register to be read	Address of first holding register to read
writeRegisterValues	ARRAY OF WORD	Register values to be written	Data
writeStartAddress	UINT	Address of the first holding register to be written	Address of first holding register to write

Table 67 – Structured ReadWriteRegistersReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadWriteRegistersReq	STRUCT			Communication request equivalent to the Read/Write Holding Registers service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	readQuantity	M	[1..1]	
	readStartAddress	M	[1..1]	
	writeRegisterValues	M	[1..1]	
	writeStartAddress	M	[1..1]	

Table 68 – Simple ReadWriteRegistersRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
readRegisterValues	ARRAY OF WORD	Read holding register values	Data

Table 69 – Structured ReadWriteRegistersRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadWriteRegistersRsp	STRUCT			Communication response equivalent to the Read/Write Holding Registers service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	readRegisterValues	M	[1..1]	

10.3.20 Read FIFO queue transaction service

The parameters of the transaction service, which is equivalent to the Read FIFO service defined in IEC 61158-5-15, are described in Table 70, Table 71, Table 72 and Table 73.

Table 70 – Simple ReadFifoQueueReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fifoPointerAddress	UINT	Address of the first register to be read in a First-In-First-Out (FIFO) queue of registers	Address of FIFO queue

Table 71 – Structured ReadFifoQueueReq data types

Data type	Definition	Description
-----------	------------	-------------

	Elementary data types	U s a g e	Multiplicity	
ReadFifoQueueReq	STRUCT			Communication request equivalent to the Read FIFO service request defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	fifoPointerAddress	M	[1..1]	

Table 72 – Simple ReadFifoQueueRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
fifoRegisterValues	ARRAY OF WORD	Register values read from the FIFO queue	Data

Table 73 – Structured ReadFifoQueueRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadFifoQueueRsp	STRUCT			Communication response equivalent to the Read FIFO service response defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	fifoRegisterValues	M	[1..1]	

10.3.21 Encapsulated interface transport transaction service

The parameters of the transaction service, which is equivalent to the service provided by the Encapsulated Interface defined in IEC 61158-5-15, are described in Table 74, Table 75, Table 76 and Table 77.

Table 74 – Simple EncapsulatedInterfaceTransportReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
meiData	ARRAY OF BYTE	Modbus encapsulated interface type specific data	MEI type specific data
meiType	USINT	Modbus encapsulated interface type (MEI type)	MEI type

Table 75 – Structured EncapsulatedInterfaceTransportReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
EncapsulatedInterfaceTransportReq	STRUCT			Communication request equivalent to the service request provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	meiData	M	[1..1]	
	meiType	M	[1..1]	

Table 76 – Simple EncapsulatedInterfaceTransportRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
meiData	ARRAY OF BYTE	Modbus encapsulated interface type specific data	MEI type specific data
meiType	USINT	Modbus encapsulated interface type (MEI type)	MEI type

Table 77 – Structured EncapsulatedInterfaceTransportRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
EncapsulatedInterfaceTransportRsp	STRUCT			Communication response equivalent to the service response provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	meiData	M	[1..1]	
	meiType	M	[1..1]	

10.3.22 Read device identification transaction service

The parameters of the transaction service, which is equivalent to the Read Device Identification service provided by the Encapsulated Interface defined in IEC 61158-5-15, are described in Table 78 to Table 83.

Table 78 – Simple ReadDeviceIdentificationReq data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
readDeviceIdCode	USINT	The parameter " Read Device ID code " allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code
objectId	USINT	Identification of the first object to obtain	Requested object ID

Table 79 – Structured ReadDeviceIdentificationReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDeviceIdentificationReq	STRUCT			Communication request equivalent to the Read Device Identification service request provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	readDeviceIdCode	M	[1..1]	
	objectId	M	[1..1]	

Table 80 – Simple IdentificationObject data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
objectId	USINT	Identification of the returned object	Returned object ID
objectValue	ARRAY OF BYTES	ASCII string representing the object value	Object value

Table 81 – Structured IdentificationObject data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdentificationObject	STRUCT			Identification objects of the Read Device Identification service response provided by the Encapsulated Interface defined in IEC 61158-5-15
	objectId	M	[1..1]	
	objectValue	M	[1..1]	

Table 82 – Simple ReadDeviceIdentificationRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
conformityLevel	BYTE	Identification conformity level of the device and type of supported access 01 : basic identification (stream access only) 02 : regular identification (stream access only) 03 : extended identification (stream access only) 81 : basic identification (stream access and individual access) 82 : regular identification (stream access and individual access) 83 : extended identification (stream access and individual access)	Conformity level
moreFollows	BOOL	In case of readDeviceIdCode 01, 02 or 03 (stream access), if the identification data does not fit into a single response and several request/response transactions may be required: – “0” : no more objectst are available – “1” : other identification objects are available and further Modbus transactions are required In case of ReadDevId code 04 (individual access), this field shall be set to “0”	More-available flag
nextObjectId	USINT	– “0”, if no more identification objects are available (moreFollows=“0”) – Identification of the next object to be obtained, if more identification objects are available (moreFollows=“1”)	Next object ID
numberOfObjects	USINT	Number of identification objects returned in this response (for an individual access, numberOfObjects=“1”)	Number of objects
readDeviceIdCode	USINT	The parameter " Read Device ID code " allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code

Table 83 – Structured ReadDeviceIdentificationRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ReadDeviceIdentificationRsp	STRUCT			Communication response equivalent to the Read Device Identification service response provided by the Encapsulated Interface defined in IEC 61158-5-15
	communicationReference	M	[1..1]	
	conformityLevel	M	[1..1]	
	IdentificationObject	M	[1..*]	
	moreFollows	M	[1..1]	
	nextObjectId	M	[1..1]	
	numberOfObjects	M	[1..1]	
	readDeviceIdCode	M	[1..1]	

10.3.23 Private Modbus transaction service

This transaction service can be used in order to send any private Modbus request. The parameters of this transaction service are described in Table 84, Table 85, Table 86 and Table 87.

Table 84 – Simple PrivateModbusReq data types

Data type	Definition	Description
privateRequest	ARRAY OF BYTE	Sequence of hexadecimal digits representing the private Modbus request

Table 85 – Structured PrivateModbusReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
PrivateModbusReq	STRUCT			Communication request of the private Modbus transaction service
	communicationReference	M	[1..1]	
	privateRequest	M	[1..1]	

Table 86 – Simple PrivateModbusRsp data types

Data type	Definition	Description
privateResponse	ARRAY OF BYTE	Sequence of hexadecimal digits representing the private Modbus response

Table 87 – Structured PrivateModbusRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
PrivateModbusRsp	STRUCT			Communication response of the private Modbus transaction service
	communicationReference	M	[1..1]	
	privateResponse	M	[1..1]	

10.3.24 Unconfirmed private Modbus transaction service

This transaction service can be used in order to send any private unconfirmed Modbus request. The parameters of this transaction service are described in Table 88, Table 89 and Table 90. Although this service specifies an unconfirmed Modbus service, at the FDT level a transaction response shall be returned in order to inform the DTM that the request was sent. For more information about this behaviour see 6.3.

Table 88 – Simple UnconfirmedPrivateModbusReq data types

Data type	Definition	Description
privateRequest	ARRAY OF BYTE	Sequence of hexadecimal digits representing the private unconfirmed Modbus request

Table 89 – Structured UnconfirmedPrivateModbusReq data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UnconfirmedPrivateModbusReq	STRUCT			Communication request of the private unconfirmed Modbus transaction service
	communicationReference	M	[1..1]	
	privateRequest	M	[1..1]	

Table 90 – Structured UnconfirmedPrivateModbusRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UnconfirmedPrivateModbusRsp	STRUCT			Communication response of the private unconfirmed Modbus transaction service
	communicationReference	M	[1..1]	

10.3.25 Modbus exception response

The parameters of the transaction service response, which is equivalent to the exception response defined in IEC 61158-5-15, are described in Table 91 and Table 92.

Table 91 – Simple ModbusExceptionRsp data types

Data type	Definition	Description	Equivalent IEC 61158-5-15 service parameter
modbusExceptionCode	BYTE	Modbus Exception Code	Exception code
modbusService	enumeration (ReadCoils ReadDiscreteInputs ReadHoldingRegisters ReadInputRegisters WriteSingleCoil WriteSingleRegister ReadExceptionStatus Diagnostics GetCommEventCounter GetCommEventLog WriteMultipleCoils WriteMultipleRegisters ReportSlaveID ReadFileRecord WriteFileRecord MaskWriteRegister ReadWriteRegisters ReadFifoQueue EncapsulatedInterface Transport ReadDeviceIdentification PrivateModbus)	Name of the Modbus service which failed: – ReadCoils – ReadDiscreteInputs – ReadHoldingRegisters – ReadInputRegisters – WriteSingleCoil – WriteSingleRegister – ReadExceptionStatus – Diagnostics: Diagnostics – GetCommEventCounter – GetCommEventLog – WriteMultipleCoils – WriteMultipleRegisters – ReportSlaveID – ReadFileRecord – WriteFileRecord – MaskWriteRegister – ReadWriteRegisters – ReadFifoQueue – EncapsulatedInterfaceTransport – ReadDeviceIdentification – PrivateModbus	Not defined

Table 92 – Structured ModbusExceptionRsp data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
ModbusExceptionRsp	STRUCT			Communication response of the private Modbus transaction service
	communicationReference	M	[1..1]	
	modbusExceptionCode	M	[1..1]	
	modbusService	M	[1..1]	

11 Channel parameter data types

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

- ReadChannelInformation;
- WriteChannelInformation.

A DTM can provide Process Channels, in order to expose detailed information to external systems such as controllers, HMI applications or to the Frame Application itself, about the process values available on the device. External systems can use this information to interpret and to directly access the runtime values in the device with standard protocol services.

This clause describes the structure and the semantic of the information which can be exposed by means of Process Channels provided by DTMs. It is up to a DTM whether it provides any channels. For a detailed description of the Process Channel parameters see Table 93 and Table 94.

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

Table 93 – Simple channel parameter data types

Data type	Definition	Description
address	UINT	Address of the discrete input, coil or register which shall be accessed. In case of a range of these Modbus data items shall be accessed, the address data type contains the address of the first Modbus data item within this range
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
iecDataType	enumeration (BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TimeOfDay DateAndTime STRING BYTE WORD DWORD LWORD WSTRING)	Identifier for IEC 61131-3 data type of the channel object
modbusDataTypes	enumeration (coil discreteInput holdingRegister inputRegister)	Identifier for the Modbus data item of the channel object
protectedByChannelAssignment	BOOL	– “0”, no restriction regarding the access on the parameter values – “1”, if the channel is set to read only by the Frame Application and neither the Frame Application nor the DTM itself are allowed to modify the parameter values. Usually set to “1” if a channel assignment exists
quantity	UINT	Number of discrete inputs, coils or registers which represent the channel object in the device

Data type	Definition	Description
scaleValue	STRING	ASCII coded REAL value with a dot as decimal separator. This value represents the scaling factor by which an actual value has to be multiplied to form a scaled value
statusChannel	BOOL	TRUE if the channel is for status information only

Table 94 – Structured channel parameter data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
FDTChannel	STRUCT			Description of the channel
	applId:applicationId	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:dataType	M	[1..1]	
	fdt:Deadband	O	[0..1]	
	fdt:descriptor	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:id	M	[1..1]	
	fdt:Ranges	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:signalType	M	[1..1]	
	fdt:StructuredElements	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:tag	M	[1..1]	
	fdt:Unit	O	[0..1]	
	frameApplicationTag	O	[0..1]	
	iecDataType	O	[0..1]	
	ModbusAccessData	O	[0..1]	
	protectedByChannel Assignment	M	[1..1]	
	UnitScaling	O	[0..1]	
FDTChannelType	STRUCT			Description of the channel component in case of channels with gateway functionality
	fdt:VersionInformation	M	[1..1]	
ModbusAccessData	STRUCT			Collection of required information to access the process value with standard Modbus services for reading or writing access
	address	M	[1..1]	
	modbusDataTypes	M	[1..1]	
	quantity	M	[1..1]	
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
UnitScaling	STRUCT			Scaling information for the actual value of the parameter provided by the channel object
	scaleValue	M	[1..1]	

12 Device Identification

The following clauses describe what kind of device identification information has to be provided by a device and how this information is interpreted in FDT in order to explicitly identify Modbus devices in FDT.

12.1 Common device type identification data types

This subclause describes the information, which is used for the protocol specific device identification. For a detailed description of the common device identification parameters see Table 95.

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

Table 95 – Simple common identification data types

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport profileSupport blockspecificProfileSupport specificSupport identSupport)	Defines the support level of a DTMDeviceType for a physical device. This data type can be used by a Frame Application to display the support level of a DTM to the user. Users may use this information for an assignment decision. - genericSupport: DTMDeviceType applies to all kind of physical devices of the corresponding fieldbus protocol; - profileSupport: DTMDeviceType applies to physical devices of a certain profile of the fieldbus protocol; - blockspecificProfileSupport DTMDeviceType applies to blocks included in physical device types (e.g.: Pressure TransducerBlock in Profibus PA); - specificSupport (DTMDeviceType is developed for this physical device type; - identSupport: The DTMDeviceType is capable of identifying the physical device in a vendor specific manner and to propose a better DTMDeviceType.
majorMinorRevision	STRING	Information of the protocol specific identification object MajorMinorRevision
match	STRING	Regular expression string, which shall match with a data type provided by scan result
name	enumeration (ProductName ModelName VendorURL UserApplicationName)	Name of the protocol specific identification object (Regular category)

Data type	Definition	Description
nomatch	STRING	Regular expression string, which shall not match with a data type provided by scan result
productCode	STRING	Information of the protocol specific identification object ProductCode
protocolName	STRING	Name of the supported communication protocol. This data type shall contain the information "protocol_Modbus"
protocolSpecificName	enumeration (ProductName ModelName VendorURL UserApplicationName)	Fieldbus protocol specific name of one single identification information
value	STRING	Single identification information
vendorName	STRING	Information of the protocol specific identification object VendorName

12.2 Topology scan data types

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

- Scan.

This subclause describes the device information, for one device in the list of scanned devices. For a detailed description of the device information parameters see Table 96 and Table 97.

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

Table 96 – Simple device information data types

Data type	Definition	Description
majorMinorRevision	STRING	Information of the protocol specific identification object MajorMinorRevision
name	enumeration (ProductName ModelName VendorURL UserApplicationName)	Name of the protocol specific identification object (Regular category)
productCode	STRING	Information of the protocol specific identification object ProductCode
protocolName	STRING	Name of the supported communication protocol. This data type shall contain the information "protocol_Modbus"
protocolSpecificName	enumeration (ProductName ModelName VendorURL UserApplicationName)	Fieldbus protocol specific name of one single identification information
value	STRING	Single identification information
vendorName	STRING	Information of the protocol specific identification object VendorName

Table 97 – Structured device information data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdManufacturer	STRUCT			Manufacturer identification
	vendorName	M	[1..1]	
IdSoftwareRevision	STRUCT			Tool relevant version of the physical device – Firmware Version
	majorMinorRevision	M	[1..1]	
IdTypeID	STRUCT			Device type identification information of a single device
	productCode	M	[1..1]	
IdValue	STRUCT			FDT name, protocol specific name and value for one identification parameter without semantic information for the Frame Application
	name	M	[1..1]	
	protocolSpecificName	M	[1..1]	
	value	M	[1..1]	
IdValues	STRUCT			Collection of multiple identification objects
	IdValue	O	[0..*]	
ModbusDevice	STRUCT			Device identification information
	schemaVersion	O	[0..1]	
	protocolName	M	[1..1]	
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	

12.3 Scan identification data types

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

- Scan.

This subclause describes the information, which is used for the protocol specific scan identification. For a detailed description of the scan identification parameters, see Table 98 and Table 99.

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

Table 98 – Simple scan identification data types

Data type	Definition	Description
configuredState	enumeration (configuredAndPhysically Available configuredAndNotPhysicallyAvailable availableButNotConfigured notApplicable)	Shall contain the value “notApplicable”
resultState	enumeration (provisional final error)	Current state of the transmitted scan information

Table 99 – Structured scan identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdAddress	STRUCT			Address information of the target device
	choice of	M	[1..1]	
	mb:ModbusSerial	S	[1..1]	
	mb:ModbusTCP	S	[1..1]	
IdBusProtocol	STRUCT			Information about the used protocol
	modbusident: protocolName	M	[1..1]	
IdManufacturer	STRUCT			Manufacturer identification
	modbusident:vendorName	M	[1..1]	
ScanIdentification	STRUCT			Scan identification information of a single device
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	
ScanIdentifications	STRUCT			Collection of multiple device scan information
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	
IdSoftwareRevision	STRUCT			Tool relevant version of the physical device – Firmware Version
	modbusident: majorMinorRevision	M	[1..1]	
IdTypeID	STRUCT			Device type identification information of a single device
	modbusident:productCode	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
IdValue	STRUCT			FDT name, protocol specific name and value for one identification parameter without semantic information for the Frame Application
	modbusident:name	M	[1..1]	
	modbusident:value	M	[1..1]	
	modbusident:protocolSpecificName	M	[1..1]	
IdValues	STRUCT			Collection of multiple identification objects
	IdValue	O	[0..*]	

12.4 Device type identification data types – provided by DTM

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

- GetIdentificationInformation.

This subclause describes the information, which is used for the protocol specific device type identification. For a detailed description of the device type identification parameters, see Table 100.

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

Table 100 – Structured device type identification data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
DeviceIdentification	STRUCT			Single device identification information
	IdBusProtocol	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdValues	O	[0..1]	
	modbusident:idDTMSupportLevel	M	[1..1]	
DeviceIdentifications	STRUCT			Collection of multiple device identification information
	DeviceIdentification	M	[1..*]	
IdBusProtocol	STRUCT			Information about the used protocol
	modbusident:protocolName	M	[1..1]	
IdManufacturer	STRUCT			Manufacturer identification
	modbusident:RegExpr	O	[0..*]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multiplicity	
	modbusident:vendorName	M	[1..1]	
IdSoftwareRevision	STRUCT			Tool relevant version of the physical device – Firmware Version
	modbusident: majorMinorRevision	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
IdTypeID	STRUCT			Device type identification information of a single device
	modbusident:productCode	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
IdValue	STRUCT			FDT name, protocol specific name and value for one identification parameter without semantic information for the Frame Application
	modbusident: protocolSpecificName	M	[1..1]	
	modbusident:name	M	[1..1]	
	modbusident:RegExpr	O	[0..*]	
	modbusident:value	O	[0..1]	
IdValues	STRUCT			Collection of multiple identification objects
	IdValue	O	[0..*]	

12.5 Mapping of protocol specific device identification objects to FDT data types

The protocol specific device identification in FDT is based on the information, which can be retrieved from a device by the protocol specific Read Device Identification service (0x2B/0xE). For this reason, it is required for Modbus devices to support the protocol specific Read Device Identification service for all objects of the Basic Device Identification category. In order to clearly identify a device in FDT, it is recommended for Modbus devices also to support the protocol specific Read Device Identification service for the objects of the Regular Device Identification category.

The following description how the protocol specific device identification information is mapped to FDT data types applies to

- the FDT Device Type Identification (explained in 12.4),
- and to the FDT Scan (explained in 12.2).

All information provided by the protocol specific identification objects of the Basic Device Identification category shall be mapped to the specific data types as described in Table 101. All Information provided by the protocol specific identification objects of the Regular Device Identification category shall be mapped to the value data type of the structured data type IdValue, together with the additional information described in Table 102.

Table 101 – Mapping of protocol specific identification objects in FDT

Protocol specific device information				Protocol specific FDT information		
Object ID	Object name	Requirement for Modbus	Category	Semantics data type	Data type	Requirement for FDT
0x00	VendorName	Mandatory	Basic	IdManufacturer	vendorName	Mandatory
0x01	ProductCode	Mandatory		IdTypeID	productCode	Mandatory
0x02	MajorMinorRevision	Mandatory		IdSoftwareRevision	majorMinorRevision	Mandatory
0x03	VendorUrl	Optional	Regular	IdValue	value	Recommended
0x04	ProductName	Optional		IdValue	value	Recommended
0x05	ModelName	Optional		IdValue	value	Recommended
0x06	UserApplicationName	Optional		IdValue	value	Recommended

Table 102 – Additional information for optional identification objects

Protocol specific device information		Protocol specific FDT information		
Object ID	Object name	Semantics element	Name	protocolSpecificName
0x03	VendorUrl	IdValue	VendorURL	VendorURL
0x04	ProductName	IdValue	ProductName	ProductName
0x05	ModelName	IdValue	ModelName	ModelName
0x06	UserApplicationName	IdValue	UserApplicationName	UserApplicationName

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

Bibliography

- [1] *MODBUS Application Protocol Specification V1.1a*, Modbus IDA, June 4, 2004
 - [2] *MODBUS over Serial Line Specification & Implementation guide V1.0*, Modbus IDA, 12 December 2002
 - [3] *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0a*, Modbus IDA, June 4, 2004
 - [4] IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*
 - [5] ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*
-

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

SOMMAIRE

AVANT-PROPOS	61
INTRODUCTION.....	63
1 Domaine d'application	65
2 Références normatives	65
3 Termes, définitions, symboles, abréviations et conventions	66
3.1 Termes et définitions	66
3.2 Abréviations	66
3.3 Conventions	66
3.3.1 Dénominations des types de données et références aux types de données.....	66
3.3.2 Vocabulaire relatif aux exigences	66
4 Catégorie de bus	66
5 Accès à l'instance et aux données du dispositif	68
5.1 Objets de la voie de processus fournis par le DTM	68
5.2 Services du DTM pour accéder à l'instance et aux données du dispositif	68
6 Comportement spécifique à un protocole.....	69
6.1 Généralités.....	69
6.2 Diffusion.....	69
6.3 Demande privée non confirmée à Modbus	73
7 Utilisation spécifique à un protocole des types de données généraux	76
8 Types de données communs spécifiques à un protocole.....	77
8.1 Généralités.....	77
8.2 Information d'adresse	77
9 Types de données pour la gestion du réseau.....	78
10 Types de données de communication.....	79
10.1 Généralités.....	79
10.2 Types de données pour la gestion de la connexion.....	79
10.3 Types de données spécifiques au service transaction.....	80
10.3.1 Généralités	80
10.3.2 Adressage des éléments de données	80
10.3.3 Service transaction Read coils	81
10.3.4 Service transaction Read discrete inputs	82
10.3.5 Service transaction Read holding registers	84
10.3.6 Service transaction Read input registers.....	85
10.3.7 Service transaction Write single coil	86
10.3.8 Service transaction Write single register.....	87
10.3.9 Service transaction Read exception status.....	88
10.3.10 Service transaction Diagnostics.....	89
10.3.11 Service transaction Get Comm event counter	90
10.3.12 Service transaction Get Comm event log	91
10.3.13 Service transaction Write multiple coils.....	92
10.3.14 Service transaction Write multiple registers	93
10.3.15 Service transaction Report slave ID.....	94
10.3.16 Service transaction Read file record	96
10.3.17 Service transaction Write file record	97

10.3.18	Service transaction Mask write register.....	99
10.3.19	Service transaction Read/write holding registers.....	100
10.3.20	Service transaction Read FIFO queue.....	101
10.3.21	Service transaction Encapsulated interface transport.....	102
10.3.22	Service transaction Read device identification.....	103
10.3.23	Service transaction Private Modbus.....	107
10.3.24	Service transaction Unconfirmed private Modbus.....	108
10.3.25	Réponse d'exception de Modbus.....	109
11	Types de données pour les paramètres des voies.....	110
12	Identification du dispositif.....	113
12.1	Types de données pour l'identification commune d'un type de dispositif.....	113
12.2	Types de données pour le balayage topologique.....	114
12.3	Types de données pour l'identification lors du balayage.....	115
12.4	Types de données pour l'identification du type de dispositif – fourni par le DTM.....	117
12.5	Mise en correspondance des objets d'identification du dispositif spécifique à un protocole avec les types de données du FDT.....	119
	Bibliographie.....	120
	Figure 1 – Partie 315 de la série IEC 62453.....	64
	Figure 2 – Séquence de diffusion avec un DTM de Communication Modbus Serial Line.....	71
	Figure 3 – Séquence de diffusion avec un DTM de Communication Modbus TCP.....	72
	Figure 4 – Séquence de diffusion sans DTM de Communication Modbus.....	73
	Figure 5 – Demande non confirmée avec un DTM de Communication Modbus Serial Line.....	74
	Figure 6 – Demande non confirmée avec un DTM de Communication Modbus TCP.....	75
	Figure 7 – Demande non confirmée sans DTM de Communication Modbus.....	76
	Figure 8 – Adressage des éléments de données.....	81
	Tableau 1 – Identificateurs du protocole.....	66
	Tableau 2 – Information d'adresse pour le mode de diffusion.....	69
	Tableau 3 – Demandes de transaction en mode diffusion.....	70
	Tableau 4 – Utilisation spécifique à un protocole des types de données généraux de FDT.....	77
	Tableau 5 – Types de données simples pour l'information d'adresse.....	77
	Tableau 6 – Types de données structurés pour l'information d'adresse.....	78
	Tableau 7 – Types de données structurés pour la gestion du réseau.....	78
	Tableau 8 – Types de données communs de communication.....	79
	Tableau 9 – Types de données structurés pour la gestion du service connexion.....	80
	Tableau 10 – Types de données simples ReadCoilsReq.....	81
	Tableau 11 – Types de données structurés ReadCoilsReq.....	82
	Tableau 12 – Types de données simples ReadCoilsRsp.....	82
	Tableau 13 – Types de données structurés ReadCoilsRsp.....	82
	Tableau 14 – Types de données simples ReadDiscreteInputsReq.....	83
	Tableau 15 – Types de données structurés ReadDiscreteInputsReq.....	83
	Tableau 16 – Types de données simples ReadDiscreteInputsRsp.....	83

Tableau 17 – Types de données structurés ReadDiscreteInputsRsp.....	83
Tableau 18 – Types de données simples ReadHoldingRegistersReq	84
Tableau 19 – Types de données structurés ReadHoldingRegistersReq	84
Tableau 20 – Types de données simples ReadHoldingRegistersRsp	84
Tableau 21 – Types de données structurés ReadHoldingRegistersRsp	85
Tableau 22 – Types de données simples ReadInputRegistersReq.....	85
Tableau 23 – Types de données structurés ReadInputRegistersReq	85
Tableau 24 – Types de données simples ReadInputRegistersRsp	85
Tableau 25 – Types de données structurés ReadInputRegistersRsp	86
Tableau 26 – Types de données simples WriteSingleCoilReq	86
Tableau 27 – Types de données structurés WriteSingleCoilReq	86
Tableau 28 – Types de données structurés WriteSingleCoilRsp	87
Tableau 29 – Types de données simples WriteSingleRegisterReq.....	87
Tableau 30 – Types de données structurés WriteSingleRegisterReq	87
Tableau 31 – Types de données structurés WriteSingleRegisterRsp	88
Tableau 32 – Types de données structurés ReadExceptionStatusReq.....	88
Tableau 33 – Types de données simples ReadExceptionStatusRsp	88
Tableau 34 – Types de données structurés ReadExceptionStatusRsp.....	89
Tableau 35 – Types de données simples pour les diagnostics.....	89
Tableau 36 – Types de données structurés DiagnosticsReq.....	89
Tableau 37 – Types de données structurés DiagnosticsRsp.....	90
Tableau 38 – Types de données structurés GetCommEventCounterReq	90
Tableau 39 – Types de données simples GetCommEventCounterRsp	90
Tableau 40 – Types de données structurés GetCommEventCounterRsp	91
Tableau 41 – Types de données structurés GetCommEventLogReq.....	91
Tableau 42 – Types de données simples GetCommEventLogRsp	92
Tableau 43 – Types de données structurés GetCommEventLogRsp.....	92
Tableau 44 – Types de données simples WriteMultipleCoilsReq	93
Tableau 45 – Types de données structurés WriteMultipleCoilsReq.....	93
Tableau 46 – Types de données structurés WriteMultipleCoilsRsp	93
Tableau 47 – Types de données simples WriteMultipleRegistersReq.....	94
Tableau 48 – Types de données structurés WriteMultipleRegistersReq	94
Tableau 49 – Types de données structurés WriteMultipleRegistersRsp	94
Tableau 50 – Types de données structurés ReportSlaveIDReq	95
Tableau 51 – Types de données simples ReportSlaveIDRsp	95
Tableau 52 – Types de données structurés ReportSlaveIDRsp	95
Tableau 53 – Types de données simples ReadFileSubRequest.....	96
Tableau 54 – Types de données structurés ReadFileSubRequest	96
Tableau 55 – Types de données structurés ReadFileRecordReq.....	96
Tableau 56 – Types de données simples ReadFileSubResponse	97
Tableau 57 – Types de données structurés ReadFileSubResponse.....	97
Tableau 58 – Types de données structurés ReadFileRecordRsp	97
Tableau 59 – Types de données simples WriteFileSubRequest.....	98

Tableau 60 – Types de données structurés WriteFileSubRequest	98
Tableau 61 – Types de données structurés WriteFileRecordReq	98
Tableau 62 – Types de données structurés WriteFileRecordRsp	99
Tableau 63 – Types de données simples MaskWriteRegisterReq	99
Tableau 64 – Types de données structurés MaskWriteRegisterReq	99
Tableau 65 – Types de données structurés MaskWriteRegisterRsp	100
Tableau 66 – Types de données simples ReadWriteRegistersReq	100
Tableau 67 – Types de données structurés ReadWriteRegistersReq	100
Tableau 68 – Types de données simples ReadWriteRegistersRsp	101
Tableau 69 – Types de données structurés ReadWriteRegistersRsp	101
Tableau 70 – Types de données simples ReadFifoQueueReq	101
Tableau 71 – Types de données structurés ReadFifoQueueReq	101
Tableau 72 – Types de données simples ReadFifoQueueRsp	102
Tableau 73 – Types de données structurés ReadFifoQueueRsp	102
Tableau 74 – Types de données simples EncapsulatedInterfaceTransportReq	102
Tableau 75 – Types de données structurés EncapsulatedInterfaceTransportReq	103
Tableau 76 – Types de données simples EncapsulatedInterfaceTransportRsp	103
Tableau 77 – Types de données structurés EncapsulatedInterfaceTransportRsp	103
Tableau 78 – Types de données simples ReadDeviceIdentificationReq	104
Tableau 79 – Types de données structurés ReadDeviceIdentificationReq	104
Tableau 80 – Types de données simples IdentificationObject	104
Tableau 81 – Types de données structurés IdentificationObject	105
Tableau 82 – Types de données simples ReadDeviceIdentificationRsp	106
Tableau 83 – Types de données structurés ReadDeviceIdentificationRsp	107
Tableau 84 – Types de données simples PrivateModbusReq	107
Tableau 85 – Types de données structurés PrivateModbusReq	107
Tableau 86 – Types de données simples PrivateModbusRsp	108
Tableau 87 – Types de données structurés PrivateModbusRsp	108
Tableau 88 – Types de données simples UnconfirmedPrivateModbusReq	108
Tableau 89 – Types de données structurés UnconfirmedPrivateModbusReq	108
Tableau 90 – Types de données structurés UnconfirmedPrivateModbusRsp	109
Tableau 91 – Types de données Simple ModbusExceptionRsp	109
Tableau 92 – Types de données structurés ModbusExceptionRsp	110
Tableau 93 – Types de données simples pour les paramètres des voies	111
Tableau 94 – Types de données structurés pour les paramètres des voies	112
Tableau 95 – Types de données simples pour l'identification commune	113
Tableau 96 – Types de données simples pour l'information relative au dispositif	114
Tableau 97 – Types de données structurés pour l'information relative au dispositif	115
Tableau 98 – Types de données simples pour l'identification lors du balayage	116
Tableau 99 – Types de données structurés pour l'identification lors du balayage	116
Tableau 100 – Types de données pour l'identification du type de dispositif	118
Tableau 101 – Mise en correspondance des objets d'identification spécifique à un protocole dans le FDT	119

Tableau 102 – Informations supplémentaires pour les objets d'identification facultatifs.....	119
Tableau 103 – Identificateurs de couche physique pour le protocole Modbus TCP.....	67
Tableau 104 – Identificateurs de couche physique pour le protocole Modbus Serial	68

IECNORM.COM : Click to view the full PDF of IEC 62453-315:2009+AMD1:2016 CSV

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

Partie 315: Intégration des profils de communication – IEC 61784 CPF 15

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 62453-315 porte le numéro d'édition 1.1. Elle comprend la première édition (2009-07) [documents 65E/131/FDIS et 65E/144/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 les amendement 1. Une version Redline montrant toutes les modifications est disponible dans cette publication.

La Norme internationale IEC 62453-315 a été établie par le sous-comité 65 E: 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.

Chacune des parties de la série IEC 62453-3xy est destinée à être lue conjointement avec l'IEC 62453-2.

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

Cette publication a été rédigée conformément aux 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)*, est disponible 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 "colour inside" 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.

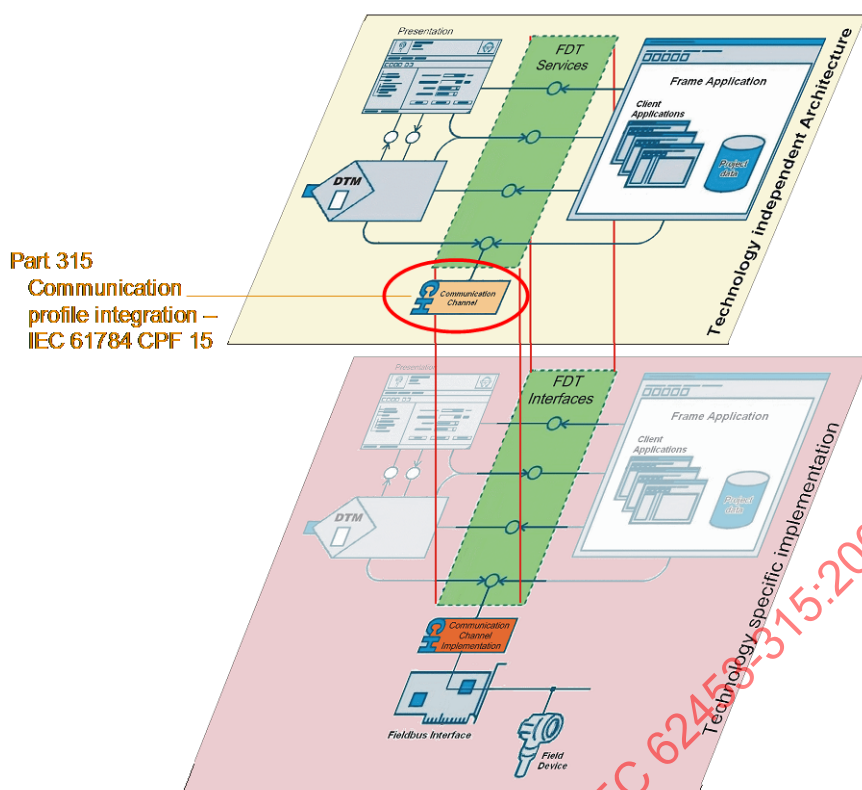
INTRODUCTION

Cette partie de l'IEC 62453 désigne une spécification de l'interface pour les développeurs des composants de FDT (Field Device Tool) (Outil pour Dispositifs de Terrain) pour la commande des fonctions et l'accès aux données au sein d'une architecture client/serveur. La spécification résulte d'une analyse et d'un processus de conception destiné à développer des interfaces normalisées afin de faciliter le développement de serveurs et de clients par de multiples vendeurs ayant besoin d'interagir sans couture.

Des bus de terrain étant intégrés aux systèmes de commande, quelques tâches supplémentaires doivent être effectuées. En plus des outils relatifs aux dispositifs ainsi qu'aux bus de terrain, il est nécessaire d'intégrer ces outils à des outils de planification à l'échelle du système à un niveau plus élevé ou à des outils d'études. En particulier, pour des utilisations dans des systèmes de commande vastes et hétérogènes, généralement dans le secteur de l'industrie de transformation, il est très important de définir clairement les interfaces d'ingénierie faciles d'utilisation pour toutes celles concernées.

Un composant logiciel spécifique à un dispositif appelé DTM (Device Type Manager) (Gestionnaire de type de dispositifs) est fourni par le fabricant du dispositif de terrain avec le dispositif auquel il est associé. Le DTM est intégré aux outils d'ingénierie via les interfaces du FDT définies dans la présente spécification. L'approche de l'intégration est généralement ouverte à tous les types de bus de terrain et par conséquent satisfait aux exigences relatives à l'intégration de différents types de dispositifs dans des systèmes de commande hétérogènes.

La Figure 1 présente la manière dont l'IEC 62453-315 est alignée dans la structure de la série IEC 62453.



IEC 1136/09

Légende

Anglais	Français
Part 315 Communication profile integration – IEC 61784 CPF 15	Partie 315 Intégration des profils de communication – IEC 61784 CPF 15
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	Service de FDT
FDT Interfaces	Interfaces de FDT
Fieldbus interface	Interface de bus de terrain

Figure 1 – Partie 315 de la série IEC 62453

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

Partie 315: Intégration des profils de communication – IEC 61784 CPF 15

1 Domaine d'application

La Famille de Profils de Communication 15 (généralement connue sous le nom de Modbus¹) définit les profils de communication basés sur l'IEC 61158-5-15 et l'IEC 61158-6-15. Le profil de base CP 15/1 (Modbus TCP) est défini dans l'IEC 61784-1. Un autre profil de communication (Modbus Serial Line) est défini en [2].

Cette partie de l'IEC 62453 fournit des informations pour intégrer le protocole Modbus TCP® et Modbus Serial Line® dans des systèmes basés sur FDT.

NOTE Cette partie de l'IEC 62453 ne spécifie que la mise en correspondance des paramètres Modbus avec les types de données de FDT. Pour connaître les restrictions des paramètres spécifiques à un protocole concernant les valeurs autorisées ainsi que les limites des matrices utilisées dans la définition des types de données de FDT, se référer à l'IEC 61158-5-15 ainsi qu'à la Spécification du Protocole pour l'Application de MODBUS.

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, *Automates programmables – Partie 3: Langages de programmation*

IEC 61158-5-15, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-15: Définition des services des couches d'application – Éléments de type 15*

IEC 61158-6-15, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-20: Spécification des services des couches d'application – Éléments de type 15*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles* (disponible en anglais uniquement)

IEC 61784-2, *Industrial communication networks – Profiles – Part -2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3* (disponible en anglais uniquement)

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

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

1) Modbus est la marque commerciale de Schneider Automation Inc. Elle est déposée aux Etats-Unis d'Amérique. 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 avec ce profil ne nécessite pas l'utilisation de la marque commerciale déposée. L'utilisation de la marque commerciale Modbus requiert l'autorisation de Schneider Automation Inc.

RFC 791, *Protocole Internet* (disponible sous <<http://www.ietf.org/rfc/rfc0791.txt>>)

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, l'IEC 62453-2 ainsi que les suivantes s'appliquent.

3.2 Abréviations

Pour les besoins du présent document, les abréviations données dans l'IEC 62453-1, l'IEC 62453-2 ainsi que les suivantes s'appliquent.

ASCII	American Standard Code for Information Interchange (Code Américain Normalisé pour l'Echange d'Informations)
DTM	Device Type Manager (Gestionnaire du type de dispositif)
FA	Frame Application (Application cadre)
IP	Internet Protocol (Protocole Internet)
RFC	Request For comment (Demande de commentaire)
TCP	Transmission Control Protocol (Protocole de commande de transport (protocoleTCP))

3.3 Conventions

3.3.1 Dénominations des types de données et références aux types de données

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

3.3.2 Vocabulaire relatif aux exigences

Les expressions suivantes sont utilisées lors de la spécification des exigences.

- | | |
|--|--|
| – Utilisation de "doit" ou de "obligatoire" | – Aucune exception autorisée. |
| – Utilisation de "il convient de" ou de "recommandé" | – Recommandation forte. Il peut être légitime, dans des cas particuliers exceptionnels, de s'écarter du comportement décrit. |
| – Utilisation de "peut" ou de "facultatif" | – La fonction ou le comportement peut être indiqué(e) selon les conditions définies. |

4 Catégorie de bus

Le protocole Modbus est identifié dans l'élément protocolld du type de données structuré 'fdt:BusCategory' par les identificateurs uniques suivants (voir Tableau 1).

Tableau 1 – Identificateurs du protocole

Valeur de l'identificateur	Nom du Protocolld	Description
59629a40-285f-11db-a98b-0800200c9a66	'Modbus over Serial Line'	Modbus sur Ligne Sérielle
59629a41-285f-11db-a98b-0800200c9a66	'Modbus over TCP'	Modbus sur TCP

Le protocole Modbus TCP utilise les identificateurs uniques suivants dans les membres physicalLayer au sein du type de données PhysicalLayer (voir Tableau 103):

Tableau 103 – Identificateurs de couche physique pour le protocole Modbus TCP

Valeur d'identificateur	Description
307DD810-C010-11DB-90E7-0002B3ECDCBE	100BaseTXFD (valeur par défaut pour Media Type Copper)
307DD812-C010-11DB-90E7-0002B3ECDCBE	100BaseFXFD (valeur par défaut pour Media Type Fiber Optic)
307DD813-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD816-C010-11DB-90E7-0002B3ECDCBE	1000BaseXFD
307DD818-C010-11DB-90E7-0002B3ECDCBE	1000BaseLXFD
307DD81A-C010-11DB-90E7-0002B3ECDCBE	1000BaseSXFD
307DD81C-C010-11DB-90E7-0002B3ECDCBE	1000BaseTFD
307DD81D-C010-11DB-90E7-0002B3ECDCBE	10GbaseFX
307DD81E-C010-11DB-90E7-0002B3ECDCBE	10GbaseLX4
307DD81F-C010-11DB-90E7-0002B3ECDCBE	10GBaseR
307DD820-C010-11DB-90E7-0002B3ECDCBE	10GbaseER
307DD821-C010-11DB-90E7-0002B3ECDCBE	10GbaseLR
307DD822-C010-11DB-90E7-0002B3ECDCBE	10GbaseSR
307DD823-C010-11DB-90E7-0002B3ECDCBE	10GbaseW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseEW
307DD824-C010-11DB-90E7-0002B3ECDCBE	10GbaseLW
307DD825-C010-11DB-90E7-0002B3ECDCBE	10GbaseSW
307DD826-C010-11DB-90E7-0002B3ECDCBE	10GbaseCX4
307DD827-C010-11DB-90E7-0002B3ECDCBE	2BaseTL
307DD828-C010-11DB-90E7-0002B3ECDCBE	10PassTS
307DD829-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10D
307DD82A-C010-11DB-90E7-0002B3ECDCBE	100BaseBX10U
307DD82B-C010-11DB-90E7-0002B3ECDCBE	100BaseLX10
307DD82C-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10D
307DD82D-C010-11DB-90E7-0002B3ECDCBE	1000BaseBX10U
307DD82F-C010-11DB-90E7-0002B3ECDCBE	1000BaseLX10
307DD830-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10D
307DD831-C010-11DB-90E7-0002B3ECDCBE	1000BasePX10U
307DD832-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20D
307DD833-C010-11DB-90E7-0002B3ECDCBE	1000BasePX20U
307DD834-C010-11DB-90E7-0002B3ECDCBE	10GBaseT ou 100BasePXFD
307DD835-C010-11DB-90E7-0002B3ECDCBE	10GBaseLRM
307DD836-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX
307DD837-C010-11DB-90E7-0002B3ECDCBE	1000BaseKX4
307DD838-C010-11DB-90E7-0002B3ECDCBE	1000BaseKR
307DD839-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXD1
307DD83A-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR2
307DD83B-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR3
307DD83C-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePRXU1
307DD83D-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR2U
307DD83E-C010-11DB-90E7-0002B3ECDCBE	10G1GBasePXR3U
307DD83F-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD1

Valeur d'identificateur	Description
307DD840-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD2
307DD841-C010-11DB-90E7-0002B3ECDCBE	10GBasePRD3
307DD842-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU1
307DD843-C010-11DB-90E7-0002B3ECDCBE	10GBasePRU3
307DD844-C010-11DB-90E7-0002B3ECDCBE	40GbaseKR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseCR4
307DD845-C010-11DB-90E7-0002B3ECDCBE	40GbaseSR4
307DD846-C010-11DB-90E7-0002B3ECDCBE	40GbaseFR
307DD847-C010-11DB-90E7-0002B3ECDCBE	40GbaseLR4
307DD848-C010-11DB-90E7-0002B3ECDCBE	100GbaseCR10
307DD849-C010-11DB-90E7-0002B3ECDCBE	100GbaseSR10
307DD84A-C010-11DB-90E7-0002B3ECDCBE	100GbaseLR4
307DD84B-C010-11DB-90E7-0002B3ECDCBE	100GbaseER4
307DD84C-C010-11DB-90E7-0002B3ECDCBE	100BasePXFD
307DD84D-C010-11DB-90E7-0002B3ECDCBE	Radiocommunication
307DD84E-C010-11DB-90E7-0002B3ECDCBE	Vitesse à 100 Mbit/s (et plus) et duplex intégral

Le protocole Modbus Serial utilise les identificateurs uniques suivants dans les membres physicalLayer au sein du type de données PhysicalLayer (voir Tableau 104):

Tableau 104 – Identificateurs de couche physique pour le protocole Modbus Serial

Valeur d'identificateur	Description
C0458028-F240-45A5-8664-70DC84FDC6FE	RS-232
3BF008DC-5A44-4220-8C3E-3C46A589A0B4	RS-422
036D1591-387B-11D4-86E1-00E0987270B9	RS-485

La propriété DataLinkLayer n'est pas applicable pour Modbus et doit être mise à "null".

5 Accès à l'instance et aux données du dispositif

5.1 Objets de la voie de processus fournis par le DTM

Pour Modbus, il n'existe pas d'ensemble minimal de paramètres défini devant être fourni par un DTM. Si un DTM fournit des valeurs de processus, il convient que les variables du processus soient modélisées comme les Voies de Processus.

5.2 Services du DTM pour accéder à l'instance et aux données du dispositif

Non applicable.

6 Comportement spécifique à un protocole

6.1 Généralités

Cet article spécifie comment traiter dans le FDT les demandes Modbus non confirmées. Il existe deux types de demandes Modbus non confirmées prises en charge dans le FDT:

- les diffusions;
- les demandes Modbus privées non confirmées.

6.2 Diffusion

En mode de diffusion, un DTM peut envoyer une demande Modbus à l'ensemble des dispositifs connectés au bus. Ce mode n'est pris en charge que par les dispositifs connectés par l'intermédiaire de Modbus Serial Line. Il peut s'agir d'une connexion directe ou d'une connexion par l'intermédiaire d'une passerelle. Le mode de diffusion doit être initié par une `fdtmodbus:ConnectRequest` avec l'adresse d'esclave du dispositif cible mise à 0 (`slaveAddress=0`). Pour plus de détails concernant l'information d'adresse à fournir pour le mode de diffusion, voir Tableau 2.

Tableau 2 – Information d'adresse pour le mode de diffusion

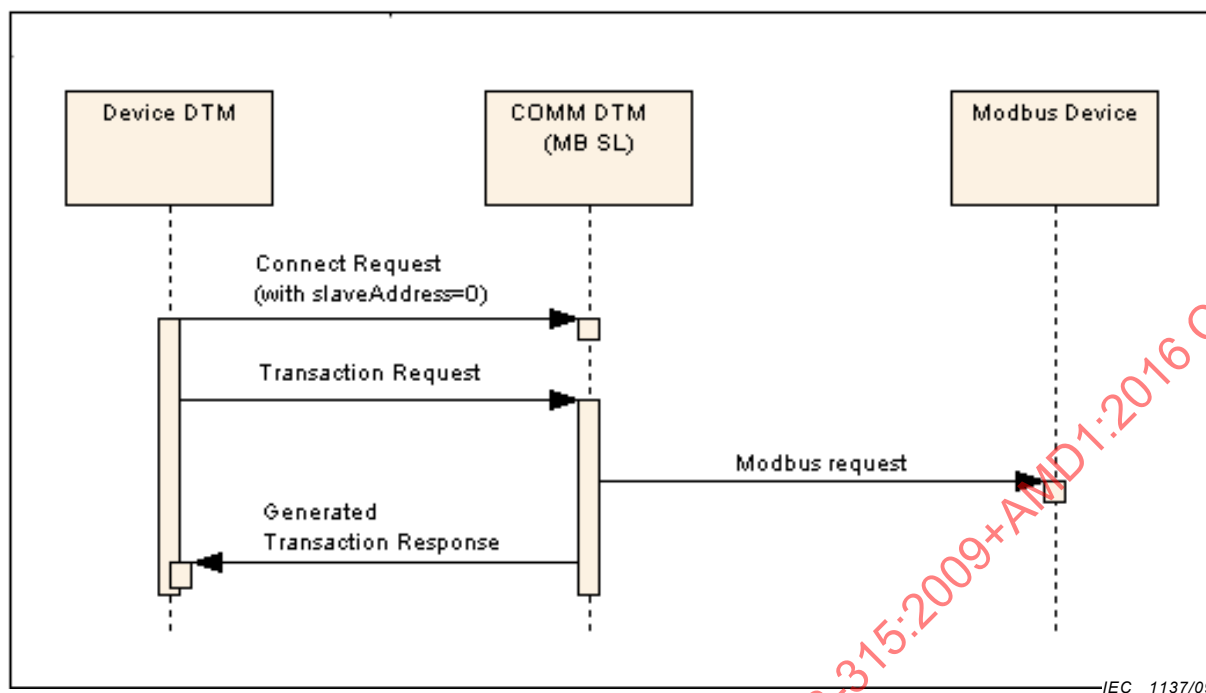
Type de données	Définition			Description
	Types de données élémentaires	Utilisation	Multiplicité	
ModbusSerial	STRUCT			Information d'adresse pour Modbus Serial Line
	slaveAddress	M	[1..1]	Doit être 0 pour le mode de diffusion générale
ModbusTCP	STRUCT			Information d'adresse pour Modbus TCP:
	slaveAddress	M	[1..1]	Doit être 0 pour le mode de diffusion générale
	tcpAddress	M	[1..1]	Adresse IP de la passerelle Modbus TCP/ Modbus Serial Line
	tcpPort	O	[0..1]	Port pour la connexion Modbus TCP (si aucune information n'est fournie, le port par défaut 502 doit être utilisé)

Étant donné qu'aucune réponse n'est retournée par le dispositif en mode de diffusion, ce dernier ne doit être utilisé qu'avec les demandes de transaction présentées dans le Tableau 3:

Tableau 3 – Demandes de transaction en mode diffusion

Demandes de transaction en mode diffusion	Restrictions
WriteSingleCoilReq	Aucune
WriteSingleRegisterReq	Aucune
DiagnosticsReq	Cette demande de transaction doit être utilisée uniquement en mode diffusion avec les sous-fonctions suivantes: - 0x01: Restart Communication Option (Redémarrer l'option de communication) - 0x03: Change ASCII Input Delimiter (Modifier le délimiteur d'entrée ASCII) - 0x04: Force Listen Only Mode (Forcer le mode en écoute uniquement) - 0x0A: Clear Counters and Diagnostic Register (Effacer les compteurs et le registre des diagnostics) - 0x14: Clear Overrun Counter and Flag (Effacer le compteur de surcharge et le fanion)
WriteMultipleCoilsReq	Aucune
WriteMultipleRegistersReq	Aucune
WriteFileRecordReq	Aucune
MaskWriteRegisterReq	Aucune
PrivateModbusReq	Doit être utilisé uniquement avec des services privés où aucune réponse du dispositif n'est requise

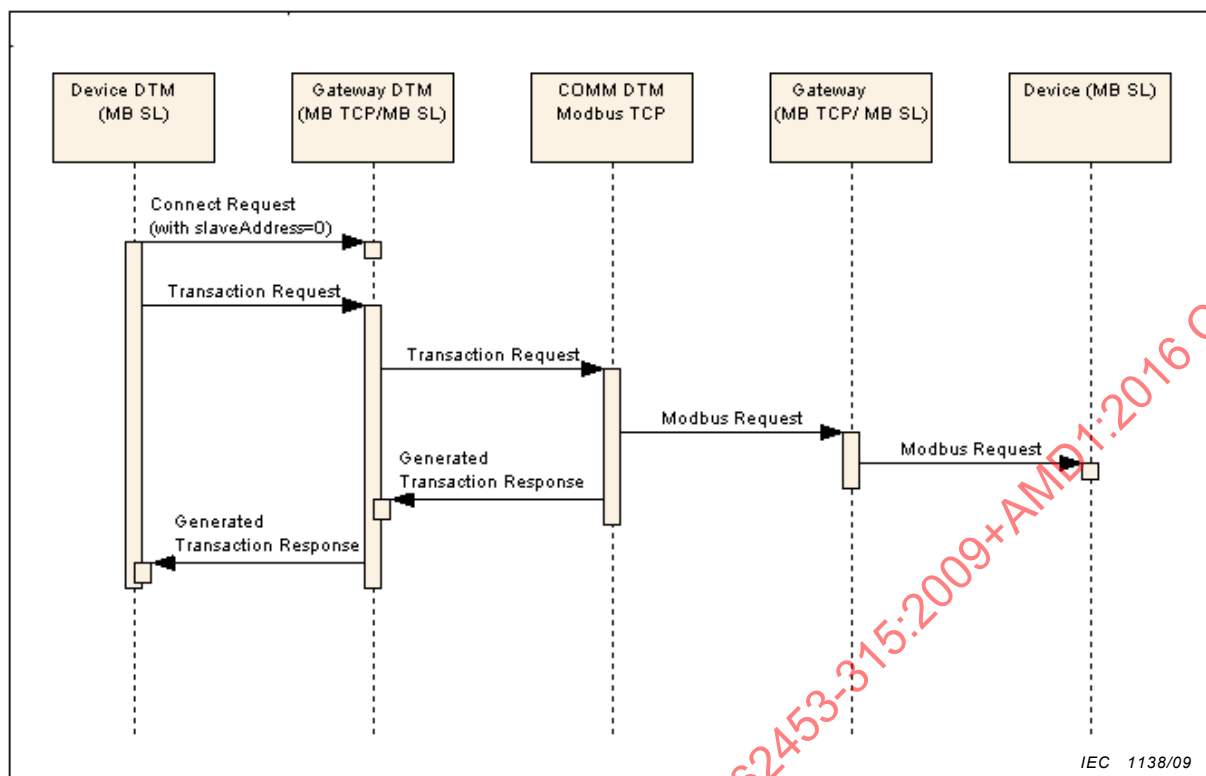
Bien qu'aucune réponse ne soit retournée par les dispositifs cibles en mode diffusion, une réponse de transaction correspondant à la demande de transaction doit être générée. Cette réponse de transaction doit être générée afin d'informer le DTM que la demande de diffusion a été envoyée sur le bus. Si un DTM de Communication Modbus est utilisé afin d'établir la communication, c'est lui qui fournit cette réponse de transaction générée (voir Figure 2 et Figure 3).



Légende

Anglais	Français
Device DTM	DTM du dispositif
COMM DTM	DTM de communication
Modbus Device	Dispositif Modbus
Connect request (with slave Address =0)	Demande de connexion (avec Adresse d'esclave = 0)
Transaction Request	Demande de transaction
Generated Transaction Response	Réponse de transaction générée
Modbus request	Demande Modbus

Figure 2 – Séquence de diffusion avec un DTM de Communication Modbus Serial Line



Légende

Anglais	Français
Device DTM	DTM de dispositif
Gateway DTM	DTM de passerelle
COMM DTM Modbus TCP	DTM de communication Modbus TCP
Gateway	Passerelle
Device MB SL	Dispositif MB SL
Connect request (with slave Address =0)	Demande de connexion (avec Adresse d'esclave = 0)
Transaction Request	Demande de transaction
Generated Transaction Response	Réponse de transaction générée
Modbus request	Demande Modbus

Figure 3 – Séquence de diffusion avec un DTM de Communication Modbus TCP

Si aucun DTM de Communication Modbus n'est utilisé pour établir la communication, le DTM de la passerelle Modbus, qui assure la liaison au niveau de communication le plus élevé différent de Modbus, doit fournir cette réponse de transaction (voir Figure 4).