IEC 61784-3-13

Edition 3.0 2021-05

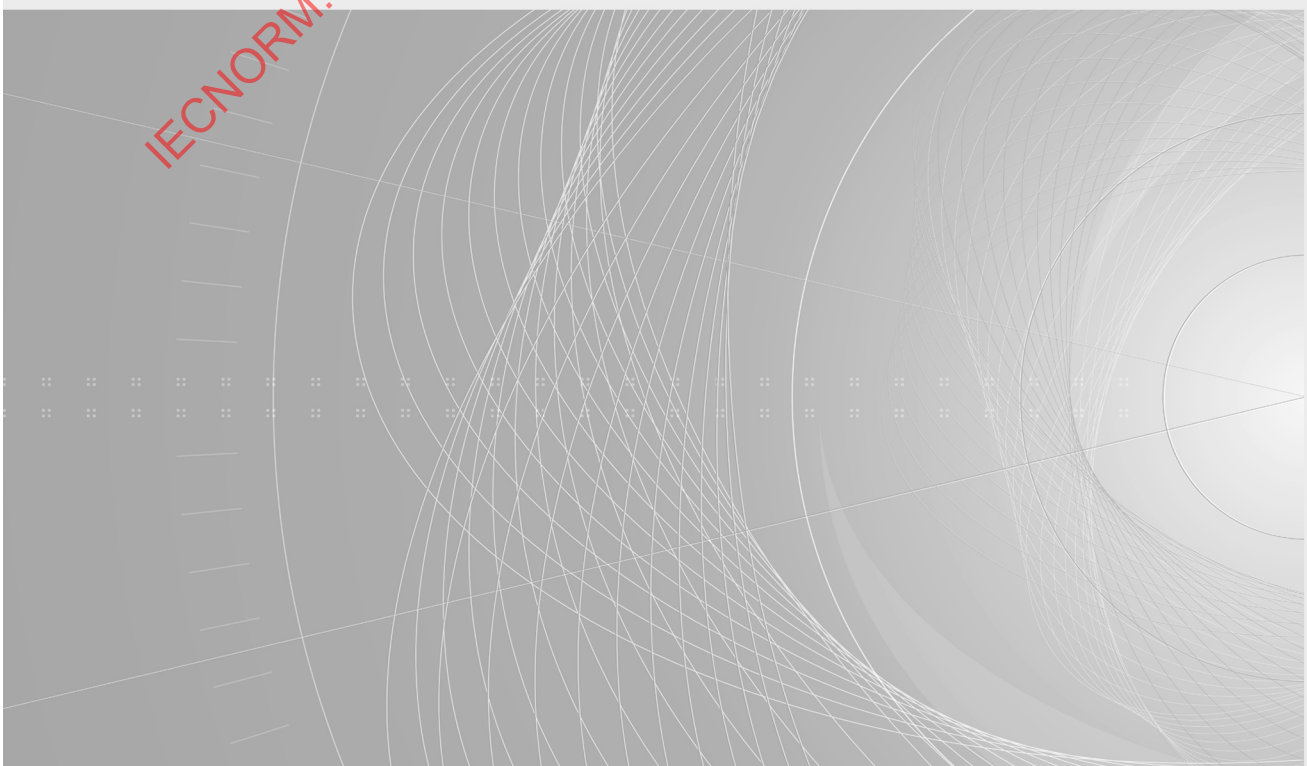
INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Industrial communication networks – Profiles – Part 3-13: Functional safety fieldbuses – Additional specifications for CPF 13

Réseaux de communication industriels – Profils – Partie 3-13: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 13



IEC 61784-3-13:2021-05(en-fr)

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

Tel.: +41 22 919 02 11
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 corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
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 It also gives information on projects, replaced and withdrawn publications.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 once a month by email.

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: sales@iec.ch.

IEC online collection - oc.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

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.

Recherche de publications IEC - webstore.iec.ch/advsearchform
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 une fois par mois par email.

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

IEC online collection - oc.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IEC 61784-3-13

Edition 3.0 2021-05

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Industrial communication networks – Profiles – Part 3-13: Functional safety fieldbuses – Additional specifications for CPF 13

Réseaux de communication industriels – Profils – Partie 3-13: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 13

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-9759-9

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

INDUSTRIAL COMMUNICATION NETWORKS – PROFILES –

Part 3-13: Functional safety fieldbuses – Additional specifications for CPF 13

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.

IEC 61784-3-13 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

This third edition cancels and replaces the second edition published in 2016. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- removal of block up-/download;
- addition of SSDO with preload;
- addition of SPDO feature sets;

- addition of SPDO with 40 bit counter;
- addition of SPDO dynamic SCT;
- addition of additional parameter sets;
- corrections and editorial improvements.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
65C/1083/FDIS	65C/1087/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts of the IEC 61784-3 series, published under the general title *Industrial communication networks – Profiles – Functional safety fieldbuses*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific document. At this date, the document 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.

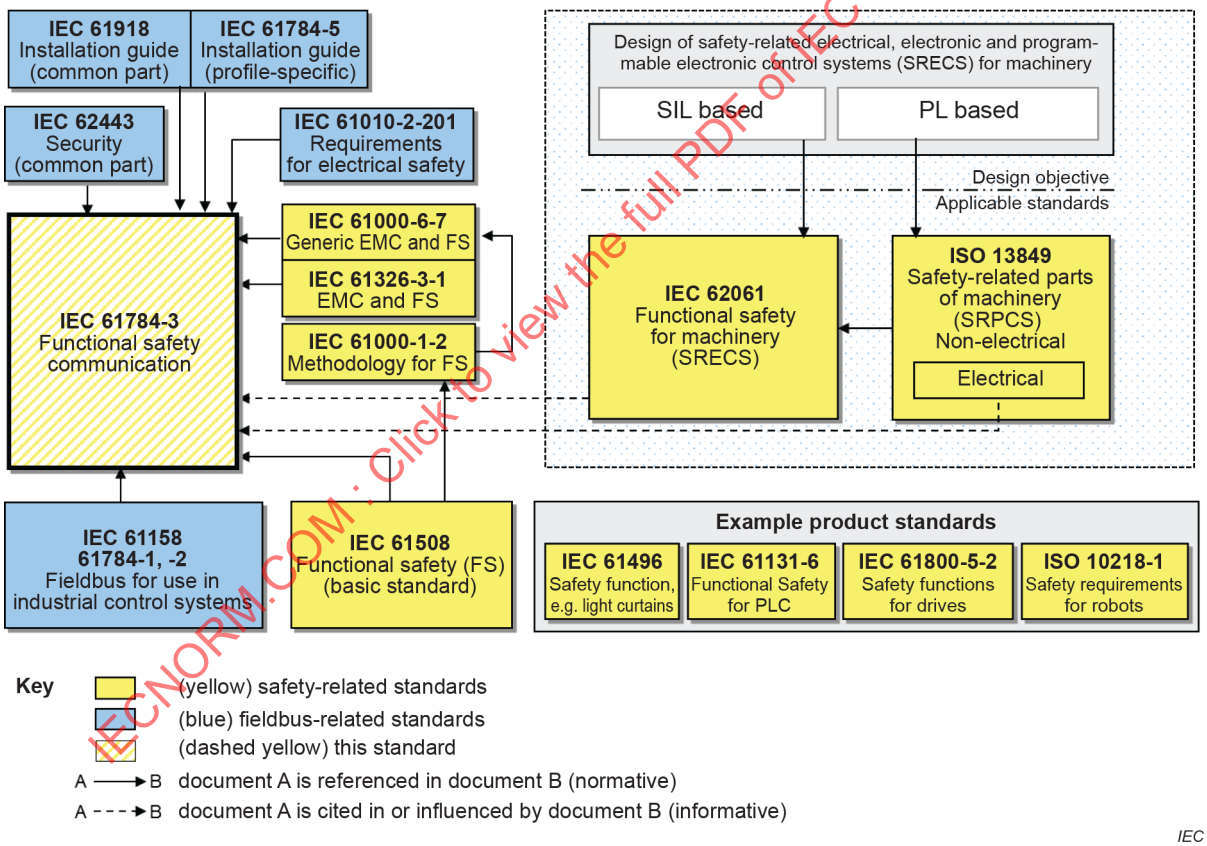
0 Introduction

0.1 General

The IEC 61158 (all parts) fieldbus standard together with its companion standards IEC 61784-1 and IEC 61784-2 defines a set of communication protocols that enable distributed control of automation applications. Fieldbus technology is now considered well accepted and well proven. Thus fieldbus enhancements continue to emerge, addressing applications for areas such as real time and safety-related applications.

IEC 61784-3 (all parts) explains the relevant principles for functional safety communications with reference to IEC 61508 (all parts) and specifies several safety communication layers (profiles and corresponding protocols) based on the communication profiles and protocol layers of IEC 61784-1, IEC 61784-2 and IEC 61158 (all parts). It does not cover electrical safety and intrinsic safety aspects. It also does not cover security aspects nor does it provide any requirements for security.

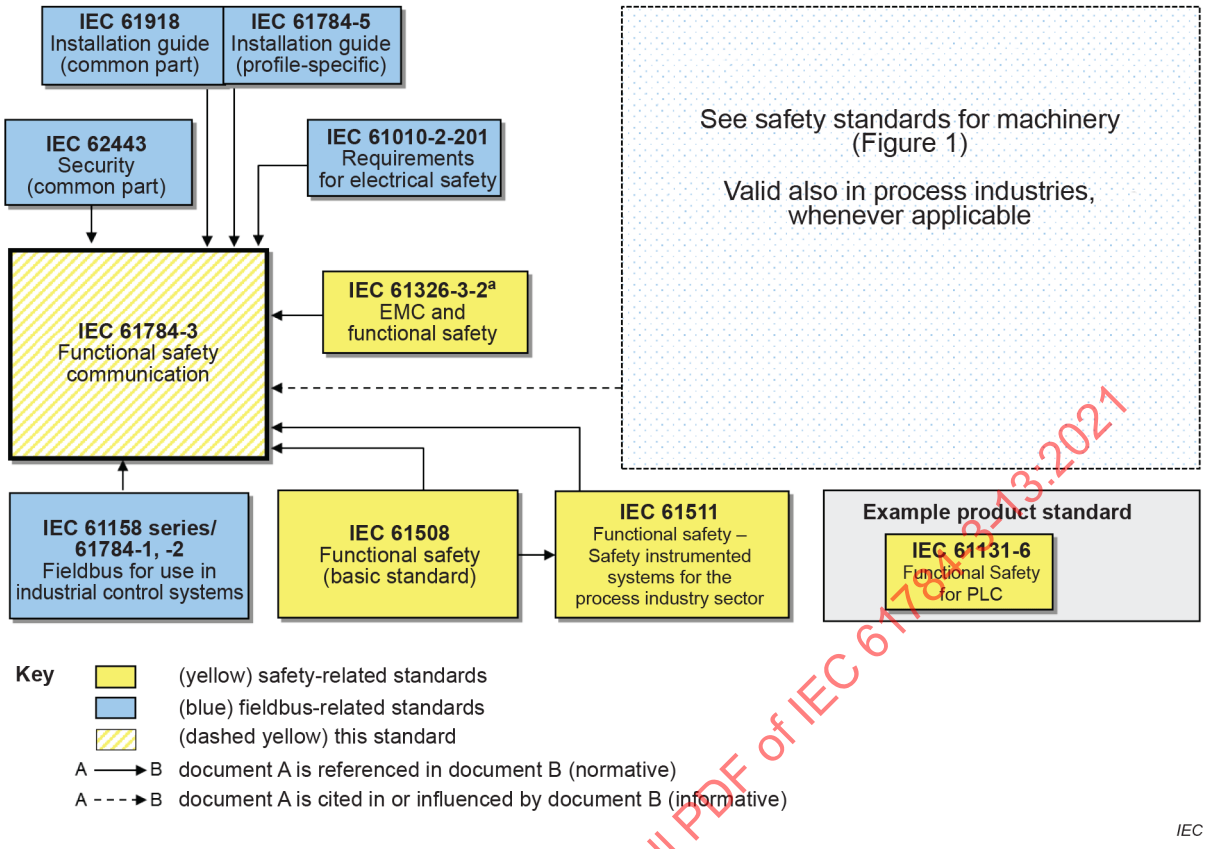
Figure 1 shows the relationships between IEC 61784-3 (all parts) and relevant safety and fieldbus standards in a machinery environment.



NOTE IEC 62061 specifies the relationship between PL (Category) and SIL.

Figure 1 – Relationships of IEC 61784-3 with other standards (machinery)

Figure 2 shows the relationships between IEC 61784-3 (all parts) and relevant safety and fieldbus standards in a process environment.



[a] For specified electromagnetic environments; otherwise IEC 61326-3-1 or IEC 61000-6-7.

Figure 2 – Relationships of IEC 61784-3 with other standards (process)

Safety communication layers which are implemented as parts of safety-related systems according to IEC 61508 (all parts) provide the necessary confidence in the transportation of messages (information) between two or more participants on a fieldbus in a safety-related system, or sufficient confidence of safe behaviour in the event of fieldbus errors or failures.

Safety communication layers specified in IEC 61784-3 (all parts) do this in such a way that a fieldbus can be used for applications requiring functional safety up to the Safety Integrity Level (SIL) specified by its corresponding functional safety communication profile.

The resulting SIL claim of a system depends on the implementation of the selected functional safety communication profile (FSCP) within this system – implementation of a functional safety communication profile in a standard device is not sufficient to qualify it as a safety device.

IEC 61784-3 (all parts) describes:

- basic principles for implementing the requirements of IEC 61508 (all parts) for safety-related data communications, including possible transmission faults, remedial measures and considerations affecting data integrity;
- functional safety communication profiles for several communication profile families in IEC 61784-1 and IEC 61784-2, including safety layer extensions to the communication service and protocols sections of IEC 61158 (all parts).

0.2 Patent declaration

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning the functional safety communication profiles for family 13. IEC takes no position concerning the evidence, validity, and scope of these patent rights.

The holder of these patent rights has assured IEC that s/he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of these patent rights is registered with IEC. Information may be obtained from the patent database available at http://patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. IEC shall not be held responsible for identifying any or all such patent rights.

INDUSTRIAL COMMUNICATION NETWORKS – PROFILES –

Part 3-13: Functional safety fieldbuses – Additional specifications for CPF 13

1 Scope

This part of IEC 61784-3 (all parts) specifies a safety communication layer (services and protocol) based on CPF 13 of IEC 61784-2 and IEC 61158 Type 13. It identifies the principles for functional safety communications defined in IEC 61784-3 that are relevant for this safety communication layer. This safety communication layer is intended for implementation in safety devices only.

NOTE 1 It does not cover electrical safety and intrinsic safety aspects. Electrical safety relates to hazards such as electrical shock. Intrinsic safety relates to hazards associated with potentially explosive atmospheres.

This document defines mechanisms for the transmission of safety-relevant messages among participants within a distributed network using fieldbus technology in accordance with the requirements of IEC 61508 (all parts)[1] for functional safety. These mechanisms may be used in various industrial applications such as process control, manufacturing automation and machinery.

This document provides guidelines for both developers and assessors of compliant devices and systems.

NOTE 2 The resulting SIL claim of a system depends on the implementation of the selected functional safety communication profile within this system – implementation of a functional safety communication profile according to this document in a standard device is not sufficient to qualify it as a safety device.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 61158-3-13, *Industrial communication networks – Fieldbus specifications – Part 3-13: Data-link layer service definition – Type 13 elements*

IEC 61158-4-13, *Industrial communication networks – Fieldbus specifications – Part 4-13: Data-link layer protocol specification – Type 13 elements*

IEC 61158-5-13, *Industrial communication networks – Fieldbus specifications – Part 5-13: Application layer service definition – Type 13 elements*

IEC 61158-6-13, *Industrial communication networks – Fieldbus specifications – Part 6-13: Application layer protocol specification – Type 13 elements*

[1] In the following pages of this document, "IEC 61508" will be used for "IEC 61508 (all parts)".

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

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

IEC 61784-3:2021, *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions*

IEC 61784-5-13, *Industrial communication networks – Profiles – Part 5-13: Installation of fielbuses – Installation profiles for CPF 13*

IEC 61918, *Industrial communication networks – Installation of communication networks in industrial premises*

ISO/IEC 19501, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*

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 61784-3 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

NOTE Italics are used in the definitions to highlight terms which are themselves defined in 3.1.

3.1.1 Common terms and definitions

NOTE These common terms and definitions are inherited from IEC 61784-3:2021.

3.1.1.1
bit error probability
Pe
probability for a given bit to be received with the incorrect value

3.1.1.2
black channel
defined communication system containing one or more elements without evidence of design or validation according to IEC 61508

Note 1 to entry: This definition expands the usual meaning of channel to include the system that contains the channel.

3.1.1.3
closed communication system
fixed number or fixed maximum number of participants linked by a *communication system* with well-known and fixed properties, and where the *risk* of unauthorized access is considered negligible

[SOURCE: IEC 62280:2014, 3.1.6, modified – transmission replaced by communication]

3.1.1.4
communication channel
logical *connection* between two end-points within a *communication system*

3.1.1.5
communication system
arrangement of hardware, software and propagation media to allow the transfer of *messages* (ISO/IEC 7498-1 application layer) from one application to another

3.1.1.6
connection
logical binding between two application objects within the same or different devices

3.1.1.7
Cyclic Redundancy Check
CRC
<value> redundant data derived from, and stored or transmitted together with, a block of data in order to detect data corruption

<method> procedure used to calculate the redundant data

Note 1 to entry: Terms "CRC code" and "CRC signature", and labels such as CRC1, CRC2, may also be used in this document to refer to the redundant data.

Note 2 to entry: See also [31], [32][2].

3.1.1.8
defined communication system
defined channel
fixed number or fixed maximum number of participants linked by a fieldbus based communication system with well-known and fixed properties, such as installation conditions, electromagnetic immunity, industrial (active) network elements, and where the risk of unauthorized access is reduced to a tolerated level according to the lifecycle model of IEC 62443 (all parts), using for example zones and conduits

3.1.1.9
DLPDU
DEPRECATED: frame
Data Link Protocol Data Unit

3.1.1.10
error
discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition

Note 1 to entry: Errors may be due to design mistakes within hardware/software and/or corrupted information due to electromagnetic interference and/or other effects.

Note 2 to entry: Errors do not necessarily result in a *failure* or a *fault*.

[SOURCE: IEC 61508-4:2010, 3.6.11 modified – notes added]

2 Figures in square brackets refer to the bibliography.

3.1.1.11
failure
termination of the ability of a functional unit to perform a required function or operation of a functional unit in any way other than as required

Note 1 to entry: Failure may be due to an *error* (for example, problem with hardware/software design or message disruption).

[SOURCE: IEC 61508-4:2010, 3.6.4, modified – notes and figures replaced]

3.1.1.12
fault
abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function

Note 1 to entry: IEC 60050-191:1990, 191-05-01 defines "fault" as a state characterized by the inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources.

[SOURCE: IEC 61508-4:2010, 3.6.1, modified – figure reference deleted]

3.1.1.13
fieldbus
communication system based on serial data transfer and used in industrial automation or process control applications

3.1.1.14
fieldbus system
system using a *fieldbus* with connected devices

3.1.1.15
hash function
(mathematical) function that maps values from a (possibly very) large set of values into a (usually) smaller range of values

Note 1 to entry: Hash functions can be used to detect data corruption.

Note 2 to entry: Common hash functions include parity, checksum or CRC.

3.1.1.16
hazard
potential source of harm

Note 1 to entry: The term includes danger to persons arising within a short time scale (for example, fire and explosion) and also those that have a long-term effect on a person's health (for example, release of a toxic substance).

[SOURCE: IEC 61508-4:2010, 3.1.2 and ISO/IEC Guide 51:2014, definition 3.2]

3.1.1.17
master
communication entity able to initiate and schedule communication activities by other stations which may be masters or slaves

3.1.1.18
message
<information theory and communication theory> ordered sequence of characters (usually octets) intended to convey information

[SOURCE: ISO/IEC 2382:2015, 2123205, modified – insertion of "(usually octets)", deletion of notes and source]

3.1.1.19
performance level
PL
discrete level used to specify the ability of safety-related parts of control systems to perform a *safety function* under foreseeable conditions

[SOURCE: ISO 13849-1:2015, 3.1.23]

3.1.1.20
redundancy
existence of more than one means for performing a required function or for representing information

[SOURCE: IEC 61508-4:2010, 3.4.6, modified – example and notes deleted]

3.1.1.21
residual error rate
statistical rate at which the SCL *safety measures* fail to detect *errors*

3.1.1.22
risk
combination of the probability of occurrence of harm and the severity of that harm

Note 1 to entry: For more discussion on this concept see Annex A of IEC 61508-5:2010.

[SOURCE: IEC 61508-4:2010, 3.1.6, and ISO/IEC Guide 51:2014, definition 3.9, modified – different note]

3.1.1.23
safety communication channel
communication channel starting at the top of the SCL of the source and ending at the top of the SCL of the sink

Note 1 to entry: It can be modelled as two SCLs connected by a *black channel* or a *defined communication system*, or a *defined channel*.

3.1.1.24
safety communication layer
SCL
communication layer above the FAL that includes all necessary additional measures to ensure safe transmission of data in accordance with the requirements of IEC 61508

3.1.1.25
safety data
data transmitted across a safety network using a safety protocol

Note 1 to entry: The *Safety Communication Layer* does not ensure safety of the data itself, only that the data is transmitted safely.

3.1.1.26
safety device
device designed in accordance with IEC 61508 and which implements the functional safety communication profile

3.1.1.27
safety function
function to be implemented by an E/E/PE *safety-related system* or other *risk* reduction measures, that is intended to achieve or maintain a safe state for the EUC, in respect of a specific hazardous event

[SOURCE: IEC 61508-4:2010, 3.5.1, modified – references and example deleted]

3.1.1.28
safety function response time
worst case elapsed time following an actuation of a safety sensor connected to a *fieldbus*, until the corresponding safe state of its safety actuator(s) is achieved in the presence of *errors* or *failures* in the *safety function*

Note 1 to entry: This concept is introduced in IEC 61784-3:2021, 5.2.4 and addressed by the functional safety communication profiles defined in this document.

3.1.1.29
safety integrity level
SIL
discrete level (one out of a possible four), corresponding to a range of safety integrity values, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest

Note 1 to entry: The target *failure* measures (see IEC 61508-4:2010, 3.5.17) for the four safety integrity levels are specified in Tables 2 and 3 of IEC 61508-1:2010.

Note 2 to entry: Safety integrity levels are used for specifying the safety integrity requirements of the *safety functions* to be allocated to the E/E/PE *safety-related systems*.

Note 3 to entry: A safety integrity level (SIL) is not a property of a system, subsystem, element or component. The correct interpretation of the phrase "SILn *safety-related system*" (where n is 1, 2, 3 or 4) is that the system is potentially capable of supporting *safety functions* with a safety integrity level up to n.

[SOURCE: IEC 61508-4:2010, 3.5.8]

3.1.1.30
safety measure
measure to control possible communication *errors* that is designed and implemented in compliance with the requirements of IEC 61508

Note 1 to entry: In practice, several safety measures are combined to achieve the required *safety integrity level*.

Note 2 to entry: Communication *errors* and related safety measures are detailed in IEC 61784-3:2021, 5.3 and 5.4.

3.1.1.31
safety PDU
SPDU
PDU transferred through the *safety communication channel*

Note 1 to entry: The SPDU may include more than one copy of the *safety data* using differing coding structures and *hash functions* together with explicit parts of additional protections such as a key, a sequence count, or a *time stamp* mechanism.

Note 2 to entry: Redundant SCLs may provide two different versions of the SPDU for insertion into separate fields of the *fieldbus frame*.

3.1.1.32
safety-related application
programs designed in accordance with IEC 61508 to meet the SIL requirements of the application

3.1.1.33
safety-related system
system performing *safety functions* according to IEC 61508

3.1.1.34
slave
communication entity able to receive *messages* and send them in response to another communication entity which may be a *master* or a slave, but not to initiate communication activities

3.1.1.35
time stamp
time information included in a *message*

3.1.2 CPF 13: Additional terms and definitions

3.1.2.1
bit error rate
likelihood of a bit misinterpretation due to electrical noise

3.1.2.2
Device Vendor Information
DVI
information for identifying a *safety node*

3.1.2.3
Safety Address
SADR
address of a *safety node* within a *safety domain*

3.1.2.4
Safety Configuration Manager
SCM
service which is responsible to manage safety related services

Note 1 to entry: Examples of managed safety related services are configuration, parameterization.

3.1.2.5
Safety Node
SN
node with *safety function*

3.1.2.6
Safety Domain
SD
address space for up to 1 023 *safety nodes*

3.1.2.7
Safety Domain Gateway
SDG
gateway to share data between different *safety domains*

3.1.2.8
Safety Domain Number
SDN
identification of a *safety domain*

3.1.2.9
Safety Network Management
SNMT
services for safety layer management

3.1.2.10
Safety Object Dictionary
SOD
repository of all data objects accessible using services of the functional safety *fieldbus*

3.1.2.11
Safety Process Data Object
SPDO
object for process data exchange between different *safety nodes*

3.1.2.12
Safety Service Data Object
SSDO
object for service data exchange between *safety nodes*

3.1.2.13
initialization
initial *safety node* communication state

3.1.2.14
operational
safety node communication state where all functional safety functions are working properly

3.1.2.15
pre-operational
safety node communication state where all application outputs are in safe state

3.1.2.16
Unique Device Identification
UDID
worldwide unique identification of a device

3.2 Symbols and abbreviated terms

3.2.1 Common symbols and abbreviated terms

CP	Communication Profile	[IEC 61784-1]
CPF	Communication Profile Family	[IEC 61784-1]
CRC	Cyclic Redundancy Check	
DLL	Data Link Layer	[ISO/IEC 7498-1]
DLPDU	Data Link Protocol Data Unit	
EMC	Electromagnetic Compatibility	
EUC	Equipment Under Control	[IEC 61508-4:2010]
E/E/PE	Electrical/Electronic/Programmable Electronic	[IEC 61508-4:2010]
FAL	Fieldbus Application Layer	[IEC 61158-5 (all parts)]
FS	Functional Safety	
FSCP	Functional Safety Communication Profile	
MTBF	Mean Time Between Failures	
MTTF	Mean Time To Failure	
PDU	Protocol Data Unit	[ISO/IEC 7498-1]
PFH	Average frequency of dangerous failure [h-1] per hour	[IEC 61508-4:2010]
PhL	Physical Layer	[ISO/IEC 7498-1]
PL	Performance Level	[ISO 13849-1]
PLC	Programmable Logic Controller	
SCL	Safety Communication Layer	
SIL	Safety Integrity Level	[IEC 61508-4:2010]
SPDU	Safety PDU	

3.2.2 CPF 13: Additional symbols and abbreviated terms

ACM	Automatic Configuration Mode	
ADR	Address information (source or destination)	
APDU	Application Protocol Data Unit	[ISO/IEC 7498-1]
CN	Controlled Node	[IEC 61158-3-13]
CT	Consecutive Time field	
DBx	Data Octet	
DSADR	Destination SADR	
DVI	Device Vendor Information	
ID	PDU Identification	
LE	Length field	
LSB	Least Significant Octet	
MAC	Medium Access Layer	
MCM	Manual Configuration Mode	
MN	Managing Node	[IEC 61158-3-13]
MSB	Most Significant Octet	
PDO	Process Data Object	[IEC 61158-3-13]
RxSPDO	Receive Process data objects	
SADR	Safety Address	
SCM	Safety Configuration Manager	
SCT	Safety Control Time	

SD	Safety Domain	
SDG	Safety Domain Gateway	
SDN	Safety Domain Number	
SDO	Service Data Object	[IEC 61158-3-13]
SN	Safety Node	
SNMT	Safety Network Management	
SOD	Safety Object Dictionary	
SPDO	Safety Process Data Object	
SPST	Safety Parameterization Software Tool	
SSDO	Safety Service Data Object	
SSADR	Source SADR	
TADR	Additional SADR for timing information	
TR	Time Request Distinctive Number	
TReq	Time Synchronization Request	
TRes	Time Synchronization Response	
TÜV	Technischer Überwachungsverein	
TxSPDO	Safety Transmit Process data objects	
UDID	Unique Device Identification	

3.3 Conventions

3.3.1 Hexadecimal values

Some values within this document are specified in hexadecimal notation. This shall be depicted by appending a "h" to the value itself. The characters 1, 2, 3, 4, 5, 6, 8, 9, 0, a, b, c, d, e, f, A, B, C, D, E and F shall be used. All values in hexadecimal notation shall depict whole octets; therefore a hexadecimal value representation always consists of an even number of characters.

EXAMPLE 1 12ABh, 05h, 1234ABCDh

To depict data where a sequence of octets shall explicitly be defined – to be independent from the exact memory representation (Big-Endian or Little-Endian) – a hexadecimal sequence notation shall be used. Every octet is shown by a two digit hexadecimal value (without the h suffix), the single octet representations shall be separated by a dash (-).

EXAMPLE 2 00-00-00-00-00-00, AC-3E

3.3.2 Binary values

Some values within this document are specified in binary notation. This shall be depicted by appending a "b" to the value itself. The characters 1 and 0 shall be used.

EXAMPLE 11001010b

3.3.3 Wildcard digits

If parts of constant values are "don't care", or have additional meaning outside the scope of the descriptive text, this shall be depicted by an x at the respective digit position.

EXAMPLE 12xBh, 1x0xx010b, 123x45

3.3.4 Diagrams

Graphical notations according to ISO/IEC 19501 are used throughout this document where appropriate.

4 Overview of FSCP 13/1 (openSAFETY™)

4.1 Functional Safety Communication Profile 13/1

Communication Profile Family 13 (commonly known as Ethernet POWERLINK™[3]) defines communication profiles based on IEC 61158-3-13, IEC 61158-4-13, IEC 61158-5-13, and IEC 61158-6-13.

The basic profile CP 13/1 is defined in IEC 61784-2. The CPF 13 functional safety communication profile FSCP 13/1 (openSAFETY™) is based on the CPF 13 basic profiles in IEC 61784-2 and the safety communication layer specifications defined in this document.

4.2 Technical overview

FSCP 13/1 defines an open safety protocol for Ethernet-based communication down to the microsecond range.

The FSCP 13/1 services and protocol specify safety related data communication between Safety Nodes (SNs). The FSCP 13/1 protocol technology is designed to provide SIL 3 safety function according to IEC 61508. The network configuration is done by using the Safety Configuration Manager (SCM). Safety Network Management (SNMT) services are used for booting and runtime diagnosis of FSCP 13/1 networks. The Safety Object Dictionary (SOD) is database provided by every SN containing configuration and user data. Safety Service Data Objects (SSDO) are used for the exchange of spontaneous data. Safety Process Data Objects (SPDO) provide services for exchange of synchronous data.

Synchronous data communication between SNs is modelled according to the publisher subscriber model (see Figure 3), whereas spontaneous data communication uses the client server model (see Figure 4).

[3] Ethernet POWERLINK™ and openSAFETY™ are trade names of the non-profit organization Ethernet POWERLINK™ Standardization Group (EPSG). This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this document does not require use of the trade names Ethernet POWERLINK™ or openSAFETY™. Use of the trade names Ethernet POWERLINK™ or openSAFETY™ requires permission of Ethernet POWERLINK Standardization Group (EPSG) and compliance with conditions for their use (such as testing and validation).

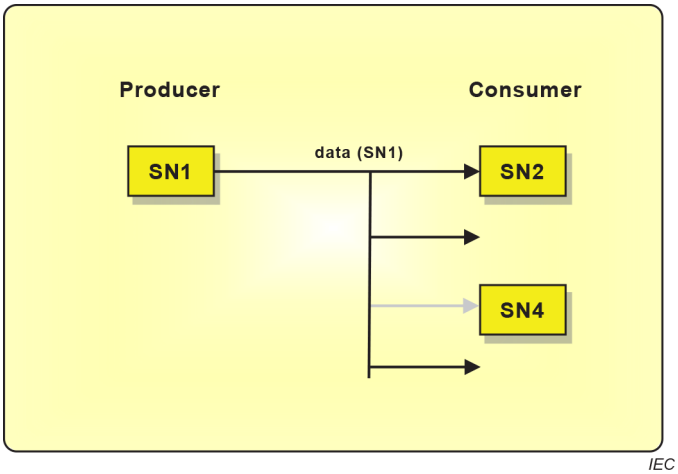


Figure 3 – Producer consumer example

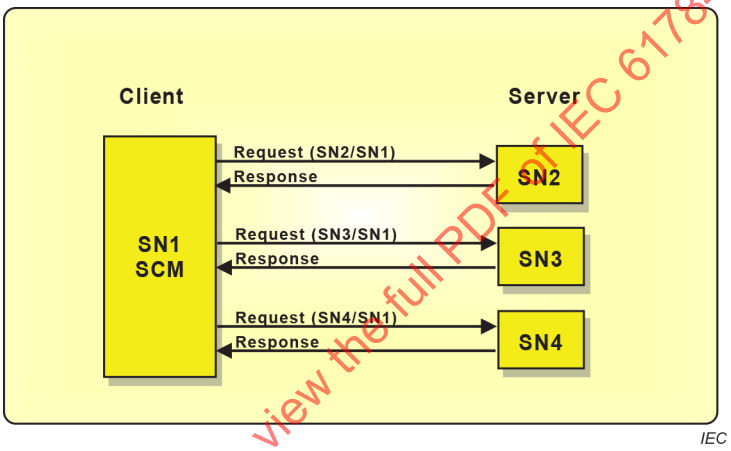


Figure 4 – Client server example

5 General

5.1 External documents providing specifications for the profile

The documents [38] and [39] may help understanding FSCP 13/1 concepts.

5.2 Safety functional requirements

The following requirements were used in the development of FSCP 13/1 and shall apply to the development of devices that implement the FSCP 13/1 protocol:

- FSCP 13/1 shall support Safety Integrity Level 3 (SIL 3) according to IEC 61508,
- The basic requirements for the development of FSCP 13/1 are specified in IEC 61784-3,
- FSCP 13/1 utilizes services defined in IEC 61158-5-13,
- FSCP 13/1 protocol is based on a black channel approach; there is no safety related dependency on the CPF 13 communication profile,
- Implementations of FSCP 13/1 shall comply with IEC 61508,
- CPF 13 requirements shall be unchanged for safety unless specified otherwise in this document,

- Safety communication and standard communication shall be independent. However, standard devices and FSCP 13/1 devices shall be able to use the same communication channel,
- Non safety related network infrastructure devices shall be usable to build a FSCP 13/1 network,
- Up to 1 023 safety related devices shall be supported within one safety domain,
- Up to 1 023 safety domains shall be supported,
- Bandwidth used by safety messages shall be kept as low as possible, safety PDUs shall be optimized for devices with less data (e.g. digital IO), and
- Unless specified otherwise in this document, the safe state of any value shall be zero (0).

5.3 Safety measures

FSCP 13/1 shall use the safety measures listed in Table 1 to detect communication errors for cyclic data transferred in SPDO (see 7.2). The safety measures shall be processed and monitored within each FSCP 13/1 device.

Table 1 – Communication errors and detection measures (cyclic)

Communication errors	Safety measures							
	Sequence number	Time stamp	Time expectation	Connection authentication	Feedback message	Data integrity assurance	Redundancy with cross checking	Different data integrity assurance systems
Corruption						X	X	
Unintended repetition		X						
Incorrect sequence		X						
Loss			X[a]					
Unacceptable delay		X	X					
Insertion			X[b]					
Masquerade			X			X	X	X
Addressing				X				
NOTE The errors "loss" and "insertion" will not be detected by the use of a time stamp. A loss of data is detected within the maximum reaction time by the use of a watchdog. If a message was sent more than once (insertion), the timestamp is unchanged and the receiving device ignores this message.								
[a] Missing PDUs shall be detected within maximum reaction time. [b] Only one message shall be accepted during a defined time frame.								

FSCP 13/1 shall use the safety measures listed in Table 2 to detect communication errors for acyclic data transferred in SSDO (see 7.3) or SNMT (see 7.4). The safety measures shall be processed and monitored within each FSCP 13/1 device.

Table 2 – Communication errors and detection measures (acyclic)

Communication errors	Safety measures							
	Sequence number	Time stamp	Time expectation	Connection authentication	Feedback message[e]	Data integrity assurance	Redundancy with cross checking	Different data integrity assurance systems
Corruption					X	X	X	
Unintended repetition	X							
Incorrect sequence	X							
Loss	X				X			
Unacceptable delay			X					
Insertion	X				X			
Masquerade						X	X	X
Addressing				X				
NOTE The errors "loss" and "insertion" will not be detected by the use of a time stamp. A loss of data is detected within the maximum reaction time by the use of a watchdog. If a message was sent more than once (insertion), the timestamp is unchanged and the receiving device ignores this message.								

In order to avoid systematic and stochastic errors, the following guidelines shall be considered during development and implementation of FSCP 13/1 (see 7.1):

– The producer of safety data shall:
  - generate a time stamp,

    NOTE Any new SPDU contains a new time stamp.
  - generate CRCs, and
  - duplicate the safety message depending on the used Safety PDU;
– The consumer of a safety message data shall:
  - check PDU part one with CRC,
  - check PDU part two with CRC,
  - compare PDU part one and two data area bit-for-bit,
  - check the received address,
  - reset the internal watchdog (counter) if a new valid PDU has been received,
  - if a PDU has the same time stamp as the last one or an older one, the PDU shall be ignored,
  - if no valid PDU is received within a defined time (Safety Control Time, see SCT_U32, Table 158), the device shall enter the safe state, and
  - if the watchdog (counter elapsed) has an overrun (not been reset within the maximum time), the device shall enter a safe state.

5.4 Safety communication layer structure

The FSCP 13/1 protocol is layered on top of a standard communication channel, which shall be considered a black channel. Figure 5 shows how the safety communication layer relates to the standard communication channel.

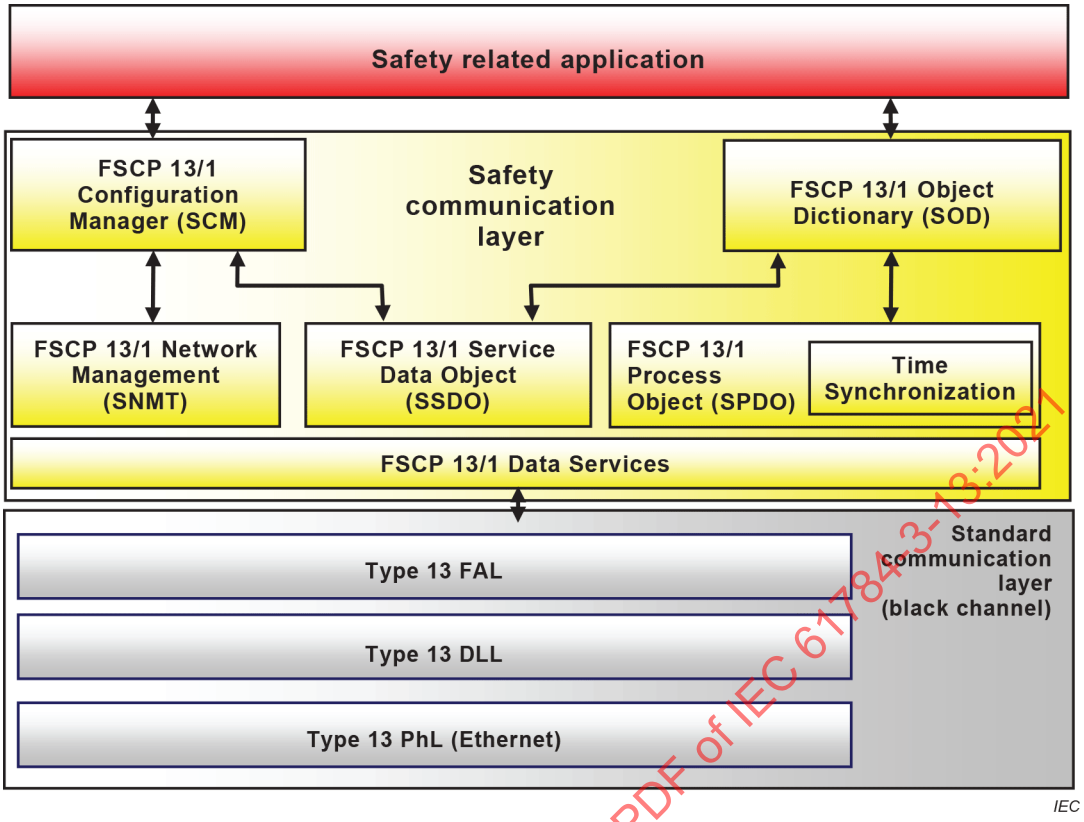


Figure 5 – Communication layer structure

Safety data is produced by the safety-related application entity. This data is encapsulated in a safety message by the safety communication layers and sent via the standard communication channel. The safety communication layer in the receiving device decodes the safety message and verifies integrity and timeliness of the data. The data is presented to the safety-related application entity. If no data is received within a defined time frame, the application entity is informed of this event, which may be used to set all outputs to the safe state.

The safety communication layer provides for:

– safe data transport (integrity and time monitoring),
– network bootup,
– run time diagnosis,
– commissioning,
– program and firmware download,
– parameterization, and
– device replacement.

All these functionalities can be accessed and controlled via the safety communication channel (see Figure 6).

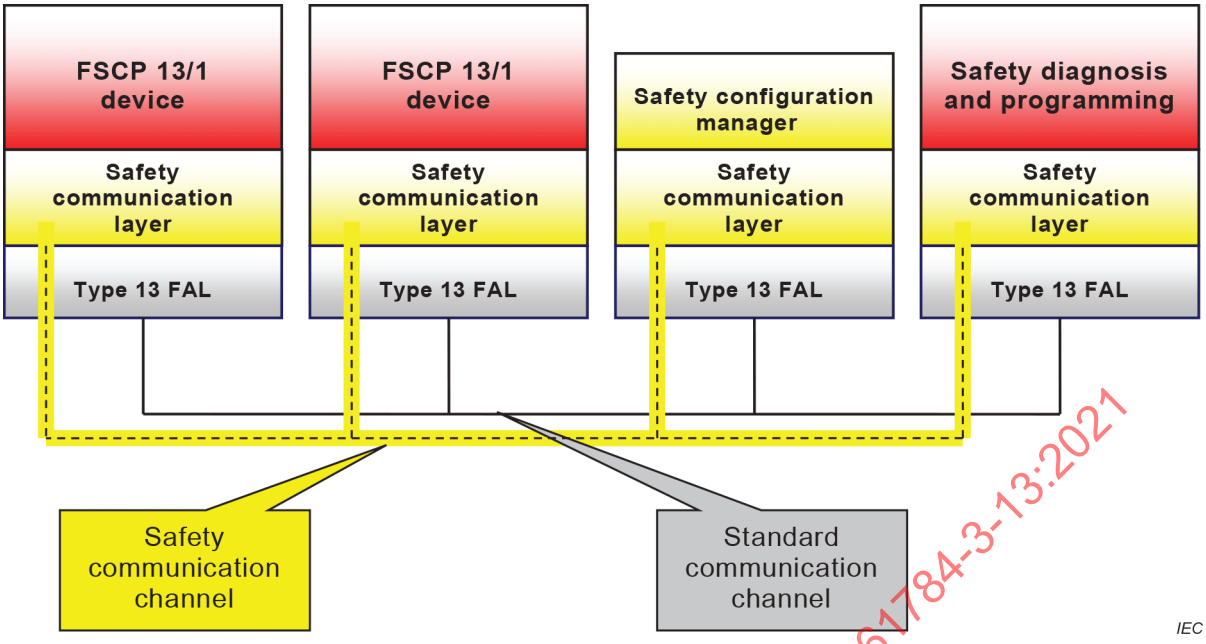


Figure 6 – Safety communication channel

5.5 Relationships with FAL (and DLL, PhL)

5.5.1 General

FSCP 13/1 shall use the services defined in IEC 61158-5-13.

5.5.2 Data types

Profiles defined in this document support all the CPF 13 data types as defined in IEC 61158-5-13.

6 Safety communication layer services

6.1 Modelling

6.1.1 Reference model

6.1.1.1 General

A FSCP 13/1 compliant implementation shall provide the following application objects:

- Safety Network Management (SNMT),
- Safety Configuration Manager (SCM),
- Safety Service Data Objects (SSDO),
- Safety Object Dictionary (SOD), and
- Safety Process Data Objects (SPDO).

How these application objects relate to each other is shown in Figure 5.

The usage of an underlying communication layer shall be mandatory.

6.1.1.2 Safety network management (SNMT)

The SNMT shall provide the required services for the Safety Configuration Manager (SCM) to configure and verify the safety related nodes within the network.

6.1.1.3 Safety service data objects (SSDO)

The SSDO shall provide the required services to access the Safety Object Dictionary (SOD) and to support the SCM.

6.1.1.4 Safety process data objects (SPDO)

The SPDO shall provide the required services for safety related process data exchange between certain entries within the SOD of communicating nodes. The time synchronization services, which are part of SPDO, shall verify the network performance between the communicating nodes.

6.1.1.5 Safety object dictionary (SOD)

The SOD shall describe the format and access services of the device specific data.

6.1.1.6 Safety configuration manager (SCM)

The SCM shall provide the required services for start-up, parameterization and all further measures to ensure safe operation over the entire life cycle of the application.

6.1.2 Communication model

FSCP 13/1 shall use FAL services defined in IEC 61158-5-13 for data exchange between Safety Nodes (SNs), FSCP 13/1 defines a producer / consumer communication model, built using these services. FSCP 13/1 nodes shall be logically separated into producer and consumer nodes.

Producer nodes shall send data identified by a specific Safety Address (SADR). Any SN within the Safety Domain may receive this data. This is done using the mechanism as described in 7.5. As a result of this communication model, the SADR uniquely identifies an SN as well as a safety PDU sent from a specific node.

Figure 7 shows characteristic producer / consumer communication. The producer node SN1 sends data. The data is uniquely identified by the producers SADR. Each node within the Safety Domain is able to receive this data.

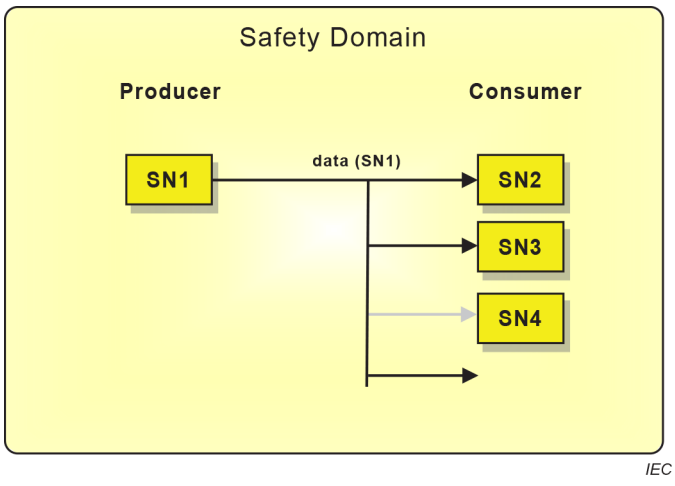


Figure 7 – Characteristic producer / consumer communication

Figure 8 demonstrates an extended version of the producer / consumer communication model which allows the producer to produce separated data for individual consumers. The producer shall use SADRs for this. The Safety Configuration Manager (SCM) shall ensure that the SADRs used are still unique within the Safety Domain.

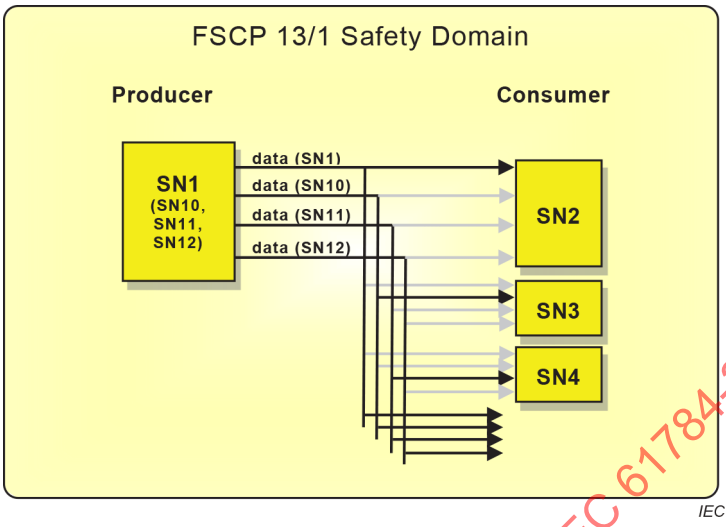


Figure 8 – Extended producer / consumer communication

Figure 9 shows the client server communication which shall be used for configuration services and network management. The SCM (or a dedicated configuration tool, see Clause 8) shall act as the client. SNs shall be limited to the server role.

Using the client server communication model, the client (SCM or Tool) sends a request to the server (SN) which is addressed within the ADR Field (see 7.1.2) using the server address and also contains the client address within the TADR Field (see 7.1.8). The server replies with a response which is addressed within the ADR Field (see 7.1.2) using the server address and also contains the client address within the TADR Field (see 7.1.8).

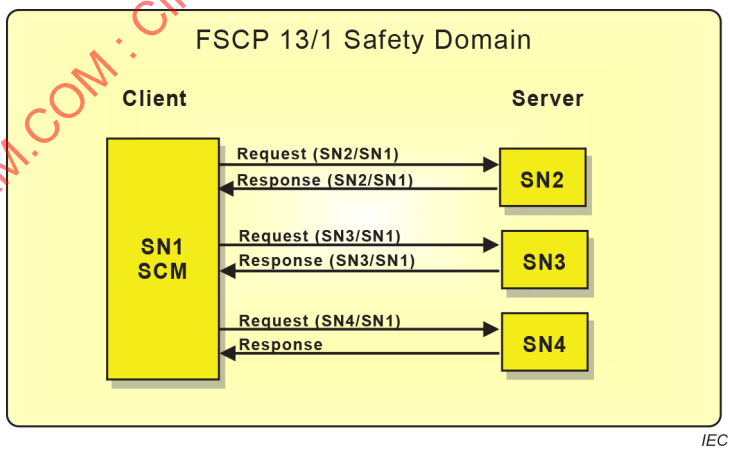


Figure 9 – Client Server communication

6.1.3 Device roles and topology

6.1.3.1 General

FSCP 13/1 device roles are defined in Table 3.

Table 3 – Device roles

Device role	Description
Safety Node (SN)	A FSCP 13/1 compliant device
Safety Configuration Manager(SCM)	Central service for managing a Safety Domain (SD) [a]
Safety Domain Gateway (SDG)	FSCP 13/1 device to exchange data between different SDs
[a] A Safety Domain defines an isolated address space of up to 1 023 SNs.	

An FSCP 13/1 device shall implement the device role of an SN and may implement SCM and/or an SDG.

Figure 10 gives an overview of possible interrelationships of device roles and Safety Domains. SD1 gives an example of a simple Safety Domain. Safety related communication of the FSCP 13/1 devices within SD1 is limited to the SNs within SD1.

SD2 in combination with SD3 shows the possibility to set up a FSCP 13/1 communication between different Safety Domains. This is done using a special SDG which is able to communicate with different Safety Domains. Inter-domain communication is handled by the device internally. SDG capability is independent from the SCM capability of a device. The SDG shall have one SADR in SD2 and one SADR in SD3.

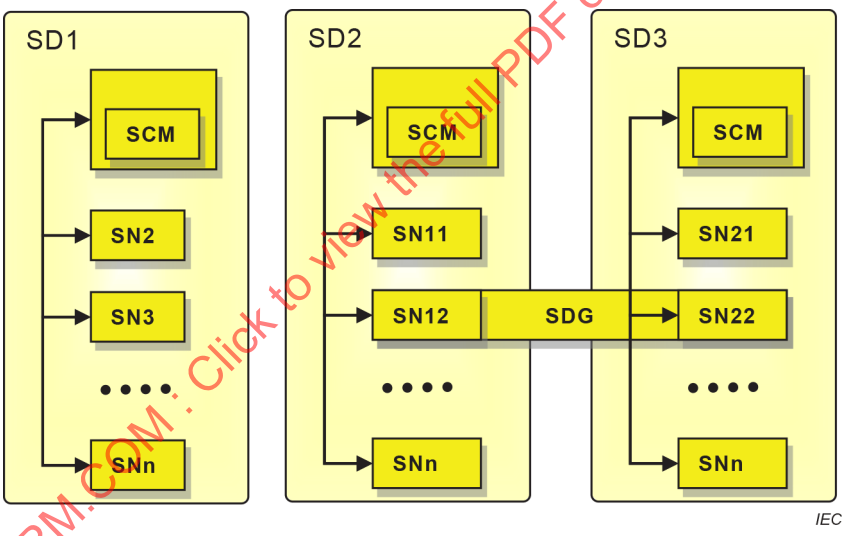


Figure 10 – Topology overview

6.1.3.2 Safety Node (SN)

An SN is a node within any network which conforms to this document. An SN shall be identified by:

- A logical address which is unique within an SD, called Safety Address (SADR). The number range for the SADR shall be 1 – 1 023 (0 is reserved and shall not be used). The SADR shall be defined at the application planning stage. Duplicate SADRs within an SD shall be detected by the SCM during network verification at start-up of the SCM or at start-up of any node within the Safety Domain.
- A physical address which is globally unique within all FSCP 13/1 compliant devices, called Unique Device Identification (UDID). Duplicate UDIDs within an SD shall be detected by the SCM.

For operation, an SN may need further data like safety related parameters or application data. It depends on the capability of the device, where this data is stored:

- At least the device specific firmware, the UDID and the Device Vendor Information (DVI) (see 7.5.4.9) shall be stored permanently on the device. There shall be no support within FSCP 13/1 to change this data via the network.
- Simple SNs without permanent memory shall receive SDN, SADR and all further parameters from the SCM after start-up.

  NOTE 1 The SCM will store all this SN specific data.

- Complex SNs with permanent memory or physical switches to adjust the SDN and SADR shall hold their data permanently on the device.

  NOTE 2 The SCM will store only the data which is not permanently stored on the SN; for all other data, the SCM will only verify it.

6.1.3.3 Safety Domain (SD)

6.1.3.3.1 General

A Safety Domain shall constitute an address space of up to 1 023 SADRs (1 – 1 023). Only SNs sharing the same SD shall be able to communicate directly with each other. SNs on different SDs shall be restricted to communication via a Safety Domain Gateway.

An SD is identified by the Safety Domain Number (SDN) having a number range of 1 – 1 023. 0 is reserved and shall not be used. The SDN is defined manually. When planning the application layout, unique SDNs shall be ensured.

6.1.3.3.2 Safety Domain protection

The danger of an external attack of FSCP 13/1 within the safety PDU process data (SPDO) is negligible. But there is a potential risk for the configuration services. When using the Internet in connection with FSCP 13/1, the security of the system shall be observed. Choosing an adequate encryption system is the responsibility of deployment planning and is outside the scope of this document. Proper state of the art security measures shall be applied.

Figure 11 depicts an example of FSCP 13/1 Safety Domain protection.

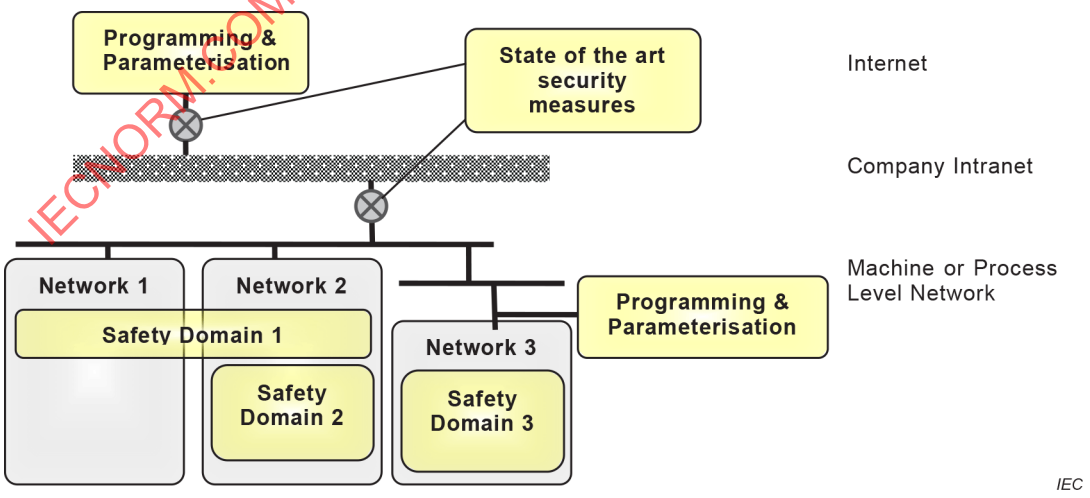


Figure 11 – Safety Domain protection (example)

6.1.3.3.3 Safety Domain separation

Interaction between Safety Domains as well as the danger of confusion during configuration and parameterization is excluded by using unique Safety Domain numbers.

If the size of the company intranet and the organization of the network management do not allow the administration of unique Safety Domain numbers, the network should be split into several manageable network scopes and separated by established methods like firewalls.

To avoid safety critical situations, the SPDO and SSDO PDO data shall additionally be coded to be unique within a certain Safety Domain (see 7.1.10). Therefore Safety PDOs which erroneously are transmitted in the wrong domain can be identified and rejected.

Figure 12 depicts an example of Safety Domain separation.

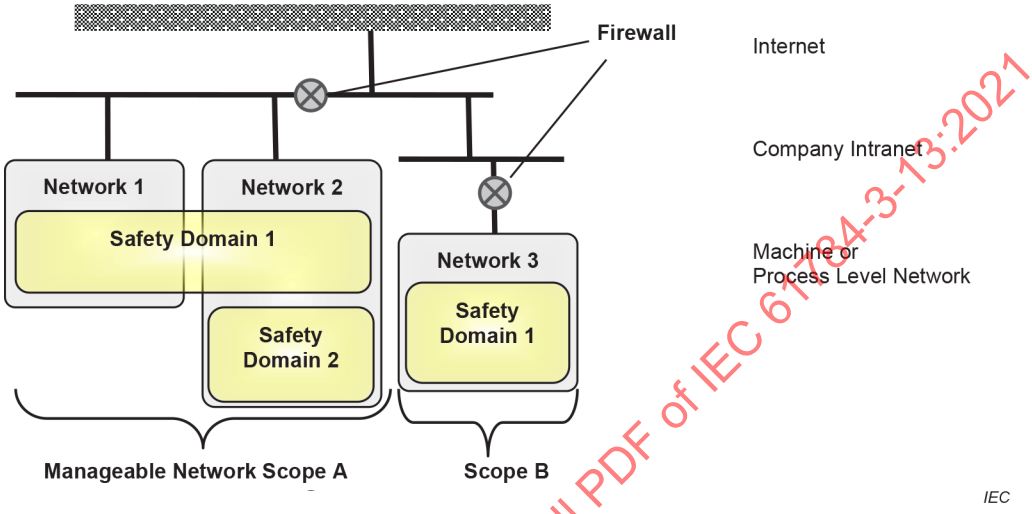


Figure 12 – Safety Domain separation (example)

6.1.3.4 Safety Domain Gateway (SDG)

This shall be a special FSCP 13/1 device role which is able to communicate with different Safety Domains (SD). Inter-domain communication is handled by the SDG internally. Within an SD, the SDG acts like a standard SN.

An SDG is responsible for:

- forward SSDO PDUs between Safety Domains, if an SN wants to send an SSDO to an SN in another Safety Domain, this SSDO is sent to the SDG which forwards the SSDO to the destination node, several SDG may have to be passed to reach the destination SN,
- forward SPDO data between Safety Domains, the SDG may provide means to read data from SPDOs in one Safety Domain and send this data into another Safety Domain inside the SPDO of the SDG.

The configuration of the SDG is vendor specific.

The scope of this document is to specify the safety related communication mechanisms which are needed for data exchange within a single Safety Domain. The communication between different Safety Domains (and the Safety Domain Gateway is a mean for this) is outside the scope of this document.

6.1.3.5 Configuration Manager (SCM)

The Safety Configuration Manager (SCM) shall be centrally responsible for managing FSCP 13/1. Within a Safety Domain, only one SN running the SCM service shall be permitted.

The SCM shall be responsible for:

- permanent storage of SN specific parameters which are not stored on the SN (see 6.1.3.2),
- unique assignment and verification of the SADR of the SNs within the SD,
- verification of the uniqueness of the UDID within the SD,
- verification of the expected parameters and DVI of the SNs, and
- sending a periodic life guard signal to all SNs within the SD.

NOTE With this periodic life guard signal from the SCM the SN detects certain failure situations where no SCM is responsible for the SN.

6.2 Life cycle model

6.2.1 General

FSCP 13/1 supports the safety-related application during several phases of its life cycle, as defined in IEC 61508. The safety related communication is only one part of the complete safety concept of the machinery. Of course, the other safety related parts of the machinery shall be considered as well. Subclause 6.2 focuses only on safety-related communication with FSCP 13/1.

6.2.2 Concept, planning and implementation

6.2.2.1 Application layout

When designing the application layout, the characteristics of FSCP 13/1 shall be taken into account. The following possibilities and restrictions shall be observed:

- Safety Domain, assigning SDNs,
- separation and protection of Safety Domains,
- Safety Addresses,
- Safety Configuration Manager, and
- Inter-domain communication.

6.2.2.2 Programming and parameterization

6.2.2.2.1 General

Programming and parameterization shall be carried out using safety related tools. Application files and parameters shall be stored in the SCM device only or directly in the corresponding device depending on the device capabilities.

Programming and parameterization of the safety related application shall be comprised of the following tasks:

- accessing SNs within an SD is referenced by the SADR only,

NOTE This allows installing several identical applications with different SDNs without changing the application.

- communication to other SDs is only possible via an SDG,
- locate SCM service,
- list DVI of the devices,
- implement safety related application, and
- specify safety related parameters (see 7.5).

The SCM is the central node to configure and verify the safety related devices. FSCP 13/1 uses a single datastream to configure a device. This is necessary to satisfy all requirements coming from the wide range of applications FSCP 13/1 is ready for. Configuration modes are:

– Automatic Configuration Mode (ACM), and
– Manual Configuration Mode (MCM).

6.2.2.2.2 Automatic Configuration Mode (ACM)

Using the Automatic Configuration Mode (ACM), the system automatically gathers the UDID from the connected SNs. The ACM shall be used only if all of the following preconditions are fulfilled:

– all safety related devices can be connected at configuration time,
– one device (single UDID) for each SN with a specific SADR,
– if a safety related device is replaced, the SCM verifies that the new device is of the same type as the replaced one; If this is fulfilled, the new UDID will be gathered and the operator shall confirm the replacement.

6.2.2.2.3 Manual Configuration Mode (MCM)

Manual Configuration Mode (MCM) is applicable for all kinds of applications; however this mode requires manual configuration of the safety related devices. So, this mode is mainly intended for applications with the following features:

– all connectable safety related devices can not be connected at one time,

  EXAMPLE 1 Robotic arm with several safety related tools.

– multiple devices (multiple UDID) for an SN with a specific SADR,

  EXAMPLE 2 Consider a modular machine application with several machine lines within a plant. Machine module A (equals to the SADR) may be used for all machine lines. For bottleneck reasons, there is more than one piece of equipment (i.e. more than one UDID) for module A (=SADR), and these pieces of equipment (UDID) are all applicable on each machine line.

– module replacement shall be handled automatically without manual intervention.

6.2.3 Commissioning

6.2.3.1 General

Commissioning can be separated into three main steps:

a) installing the application (see 6.2.3.2),
b) setting up the configuration depending on the configuration mode (see 6.2.3.3),
c) verification of the safety functionality (see 6.2.3.4).

6.2.3.2 Installation

Installation of applications and parameters on the FSCP 13/1 devices shall be done using one of the following methods:

– an FSCP 13/1 compliant programming tool which uses the specified FSCP 13/1 services (see 7.3), or
– proprietary safety related transfer protocol to exchange data from the programming system to a safety related device via network or direct communication connection (e.g. serial line).

During installation phase of the life cycle, the capability of the device shall determine what data and where the data shall be downloaded:

– if the device itself holds the data, the data shall be downloaded to the device,
– if the device is not able to hold the data, the data shall be stored on the SCM. The SCM shall ensure the proper update of all devices during power up using the FSCP 13/1 configuration services.

Following the installation of applications and parameters, the SDN shall be updated if needed (see 6.1.3.3).

6.2.3.3 Configuration setup

6.2.3.3.1 Configuration setup using ACM

A configuration setup using ACM shall be comprised of the following four steps:

a) power up all safety related devices,
b) ensure that communication layer works for all SNs,
c) SCM starts gathering UDID (without operator interference),
d) the commissioner shall acknowledge each device.

6.2.3.3.2 Configuration setup using MCM

A configuration setup using MCM shall be comprised of the following three steps:

a) the commissioner shall enter the list of possible SADRs with the corresponding UDIDs manually, for this a safety related configuration tool shall be used,
b) the SCM shall accept all configured SADR/UDID combinations,
c) the SCM shall not accept mismatches and shall not repair mismatches as done in the ACM. This means, all mismatches shall be corrected by the commissioner by entering the corrected value using the safety related configuration tool.

6.2.3.4 Verification

Verification shall be comprised of the following five steps:

a) The commissioner shall verify the safety functionality against the safety specification for the application. In the case of device replacements or changes in the routing within the communication layer, the commissioner shall continue with step 6.2.3.3;
b) If verification is carried out during commissioning, all combinations of exchangeable modules shall be verified;

NOTE This is especially important in the case of complex modular machines (in combination with MCM).

c) If verification is carried out due to maintenance, only the safety related functionality affected by the maintenance task shall be verified;
d) When verification is finished, an installation specific backup of the current configuration should be made;
e) In case of an SCM replacement, a download of application files, parameters, and the configuration shall be done; otherwise the commissioning steps (installation, configuration and verification) shall be repeated.

6.2.4 Operation terms

6.2.4.1 Transfer of safety related data

During operation, the safety related data transfer is handled by FSCP 13/1. The underlying network is responsible for the data transport between the nodes with a defined timing quality. Loss of timing or transport quality within the underlying communication layer is detected by the FSCP 13/1 layer (see 7.7.2) and will result in loss of availability, but not in loss of safety (black channel approach).

EXAMPLE See Figure 13.

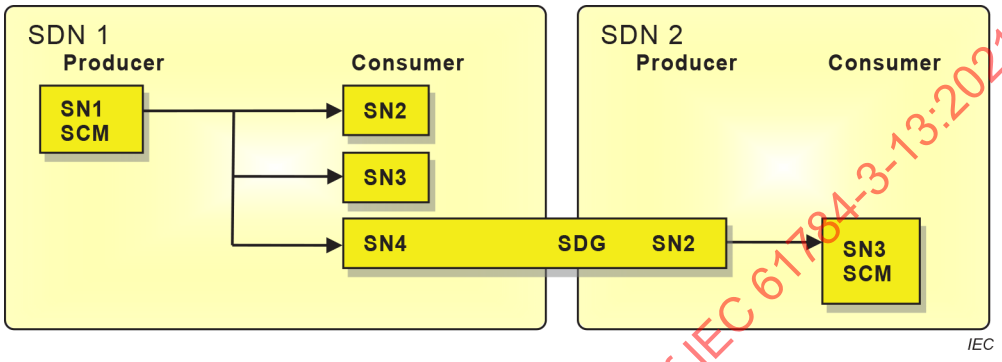


Figure 13 – Data flow example

6.2.4.2 Time synchronization and validation

In order to avoid any missed or sudden delays of data, network performance verification shall be done. From time to time each consumer shall query all connected producers for their relative time. By comparison of the relative time, the elapsed time and the time information from any transfer, the consumer shall check the timeliness of the received data.

The network performance verification is part of the data transfer. At first, a consumer sends a "Time Request" PDU to the connected producer. After receiving this PDU, the producer sends its data with an additional code, which enables the consumer to identify this message as a "Time Response". By measuring the time delay between the "Time Request" and the "Time Response", the consumer is able to check whether the network is fast enough to meet the requirements of the application depending on the worst case reaction time.

For a detailed description, refer to 7.7.2.

6.2.4.3 Life guarding

With the service Life Guarding (see 7.5.4.6), the FSCP 13/1 shall ensure that only configured SNs are Operational in an FSCP 13/1 network. An SN expects frequent Life Guarding signals to stay in the Operational state. Otherwise it shall change to the state Pre-Operational and therefore no longer being a risk to the safety related network.

The interval of the Life Guarding signal shall be application specific, and is typically determined by the amount of time needed to verify the application safety functions.

Using organizational measurements like switching off all SNs in case of reconfiguring the SCM, may eliminate the risk in this situation. As a result, the interval for the life guarding signal may be set to a very large value (several days) for these cases.

6.2.4.4 Startup after power up or reset

After power up, the SCM (see 7.7.9) shall do the network verification. If there was no change within the network, no special tasks are required.

6.2.4.5 Recover from network failure

Following a network breakdown, all affected SNs will enter the Pre-Operational state. Since the corresponding consumer SNs will get no further data, these nodes shall enter the safe state.

After recovery of the network, all affected SNs shall send an SNMT_SN_in_PRE_OP information to the SCM. The SCM, then, shall start with the initialization of the SNs.

6.2.5 Maintenance terms

See 9.6.

6.3 Non safety communication layer

6.3.1 General

Subclause 6.3 lists the requirements on the non safety communication layer for transportation of safety PDUs. Communication protocols according to CPF 13 (see IEC 61784-2, IEC 61158-3-13, IEC 61158-4-13, IEC 61158-5-13 and IEC 61158-6-13) fulfil all these requirements.

6.3.2 Requirements for data transport

6.3.2.1 General

The non safety communication layer is responsible for transporting the data between the SNs. The coexistence of not safety related nodes and safety related nodes is allowed.

6.3.2.2 Masquerading

To avoid any possibilities of masquerading data as safety PDUs, non safety related nodes shall not contain code which is able to generate safety PDUs according to Clause 7.

The non safety communication layer is permitted read only access of data within a safety PDU. This shall be done by accessing the data within the safety PDUs without using the safety PDUs specific data integrity measures, such as CRC and data repetition of PDU part one in PDU part two.

6.3.2.3 Communication model

Seen from the view of CPF 13, Figure 14 depicts the common safety communication model of CNs with 0 to n SNs.

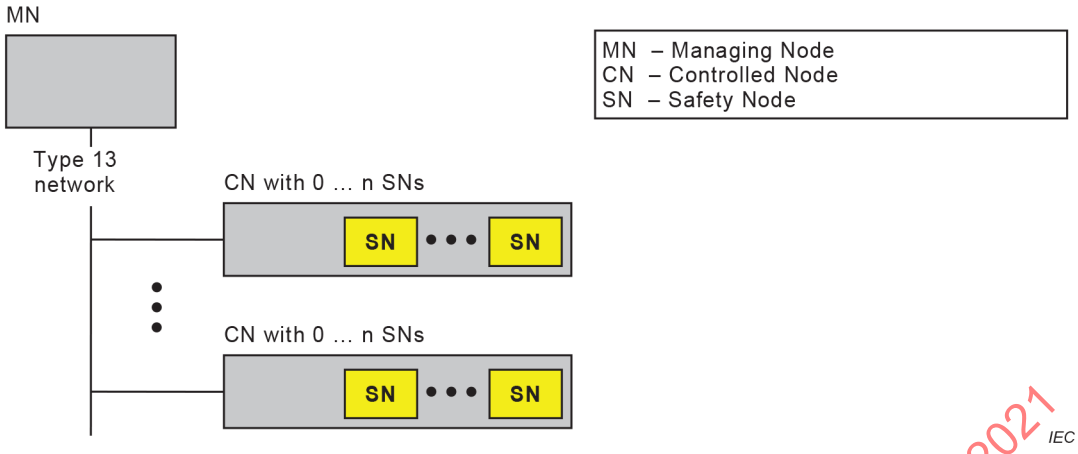


Figure 14 – Communication model

The MN and CN functionality is as defined in IEC 61158-3-13; the MN and CN are not involved in any safety function.

The safety function is implemented by an entity inside the CN which is called the Safety Node (SN). Different cases are possible:

– CN without an SN,
– CN with only one SN,
– CN with several safety related devices represented as a single SN. The safety of the module bus and the add-on modules itself shall be ensured by the vendor,
– CN with multiple SNs, or
– CN as a modular system and multiple SNs. The safety of the bus connecting the modules is provided by FSCP 13/1.

6.3.2.4 Transport of SPDO

The non safety communication layer shall transport the SPDO data between the SNs. The services as defined in IEC 61158-5-13 shall be used.

NOTE 1 SPDO payload is used for cyclic process data, like the state of an input or output. SPDOs are cyclically retransmitted within a certain time. Sending an SPDO is solely time triggered and not bound to a change in the data itself.

SPDO data shall be transmitted as safety PDUs inside Type 13 application layer PDUs (APDUs). Isoc2 APDUs according to IEC 61158-6-13 shall be used. An SPDO is generated cyclically by an SN, the producer, this SPDO can have one or many target SNs, the consumers. To transport an SPDO from its producer to its consumers, Type 13 mechanisms shall be used.

The basic principle of SPDO transport is shown in Figure 15.

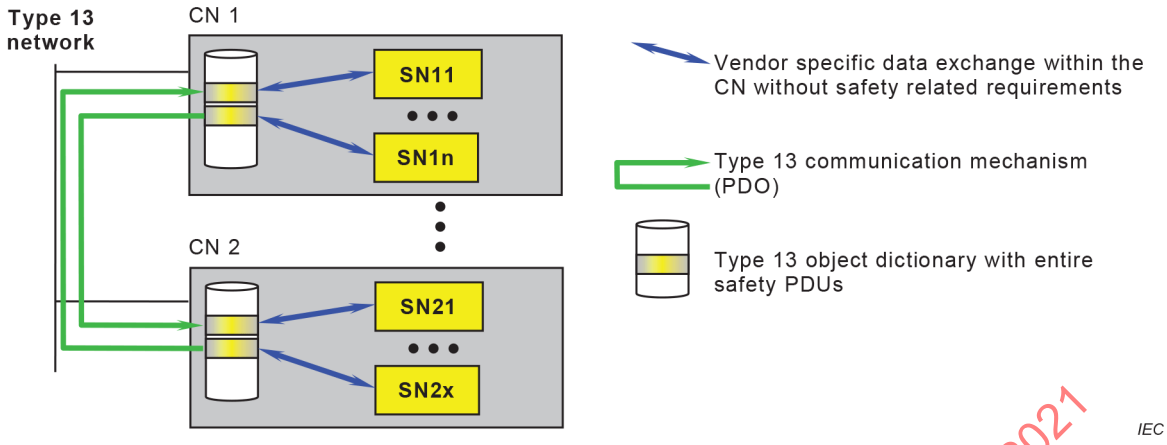


Figure 15 – SPDO transport

The transmission path between the producer SN and the consumer SN is defined as:

- the producer SN builds an SPDU,
- in the CN to which the producer SN is connected (producer CN), an object dictionary entry of data type Domain shall be defined to embody the SPDU,
- it is the responsibility of the producer CN implementation to move the SPDU data to the object dictionary entry,
- this object dictionary entry, then, shall be transmitted in a Type 13 application layer PDU (APDU), which is sent cyclically using the process data object ASE (PDO-ASE) as specified in IEC 61158-5-13,
- the object dictionary of the receiving CN (consumer CN) shall define an object of data type Domain, matching the definition of the producer's object dictionary entry; The consumer CN shall receive the APDU and copy the contained safety PDU to the Domain object mentioned before, and
- in the last step, the consumer CN implementation is responsible to move the SPDU data from the object dictionary entry to the consumer SN.

The mechanisms involved in transmitting the SPDO between the CN's object dictionary and the SNs (depicted by blue arrows in Figure 15) are outside the scope of this document and may be vendor specific.

NOTE 2 From the view of the Type 13 communication services and protocol it is not necessary to distinguish communication of SPDO from other data. There is no special treatment of safety data by Type 13 mechanisms. This fits with the black channel approach applied.

NOTE 3 The SNs involved in the communication do not need any knowledge about the Type 13 communication path.

6.3.2.5 Transport of SSDO

The non safety communication layer shall transport the SSDO data between the SNs. The services as defined in IEC 61158-5-13 shall be used.

The basic principle of SSDO transport is shown in Figure 16.

NOTE 1 SSDO payload is used for spontaneous data communication, like configuration or parameterization.

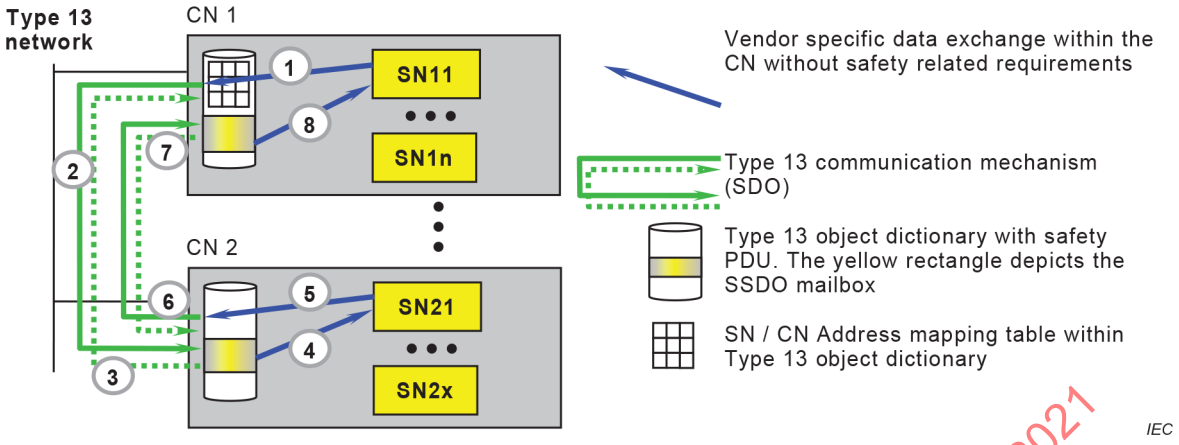


Figure 16 – SSDO transport

SSDO data shall be transmitted as safety PDUs inside Type 13 application layer PDUs (APDUs). Asyn2 APDUs with pduBody type SDO according to IEC 61158-6-13 shall be used.

An SSDO is generated spontaneously by an SN (the source SN) which is transmitted to exactly one target SN. To transport an SSDO from its source to its destination, Type 13 mechanisms shall be used.

Exchanging SSDOs between SNs is done using Type 13 services and protocols in combination with a mailbox mechanism (figures in parentheses reference Figure 16):

(1) The source SN generates an SSDO request to a specific destination SN and builds a corresponding SPDU,

(2) the CN serving the source SN creates an SDO write access to the mailbox of the CN serving the destination SN, the SPDU of step (1) is written into the object dictionary of the target CN; This requires an SN to CN address mapping table within the CN serving the source SN, to get the Type 13 node ID of the CN serving the destination SN (target CN); It is the responsibility of the Type 13 system configuration to provide this address mapping table, the mailbox shall be a Type 13 object dictionary entry of data type Domain which should be defined to embody an SPDU,

(3) the SDO write access is confirmed by the SDO protocol, the connection is kept open,

(4) the destination CN redirects the data (SPDU) within the mailbox to the destination SN, for this the destination CN shall read the address information inside the SPDU,

(5) the destination SN generates an SSDO response and builds a corresponding SPDU,

(6) the destination CN creates an SDO write access to the mailbox of the source CN and writes the response SPDU into the object dictionary of the source CN,

(7) the SDO write access is confirmed, the connection is closed, and

(8) the source CN redirects the data (SPDU) within the mailbox to the destination SN, for this the source CN shall read the address information inside the SPDU.

The mechanisms involved in transmitting the SPDU between the CN's object dictionary and the SNs (depicted by blue arrows in Figure 16) are outside the scope of this document and may be vendor specific.

NOTE 2 From the view of the Type 13 communication services and protocol it is not necessary to distinguish communication of SSDO from other data. There is no special treatment of safety data by Type 13 mechanisms. This fits with the black channel approach applied.

NOTE 3 The SNs involved in the communication do not need any knowledge about the Type 13 communication path.

6.3.2.6 Representation of diagnostic data

The non safety communication layer shall support services to provide FSCP 13/1 specific diagnostic data according to Clause 7.

The basic principle of diagnostic data representation is shown in Figure 17.

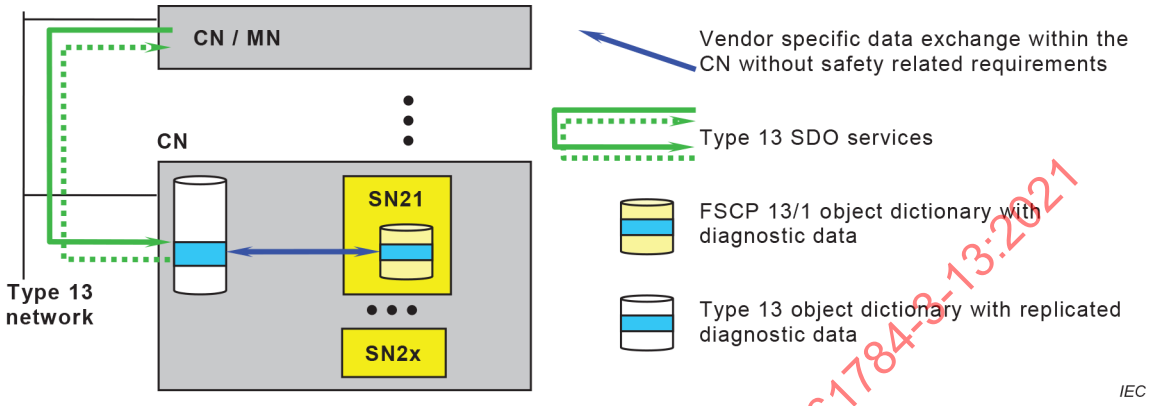


Figure 17 – Diagnostic data representation

Diagnostic data within an SN is stored in the SOD (see Clause 7).

NOTE Usually this diagnostic data is requested from distinguished nodes (e.g. central controller, human machine interface, service equipment) to process the diagnostic data and to report them to the end user.

The CN shall replicate the diagnostic data from the SOD in the Type 13 object dictionary within the CN, where they can access using Type 13 SDO services.

6.3.3 Domain protection and separation

In accordance with 6.1.3.3.2 and 6.1.3.3.3, the communication layer shall support proper security features to protect the domain against attacks from outside and to avoid interferences between separated domains.

7 Safety communication layer protocol

7.1 Safety PDU format

7.1.1 Structure of Safety PDUs

7.1.1.1 General

A safety PDU is transferred within the data part of a Type 13 APDU. One Type 13 APDU may contain zero to many safety PDUs, each reflecting a producer / consumer relationship between SNs, see Figure 18. Safety and non safety data can be mixed within a Type 13 APDU.

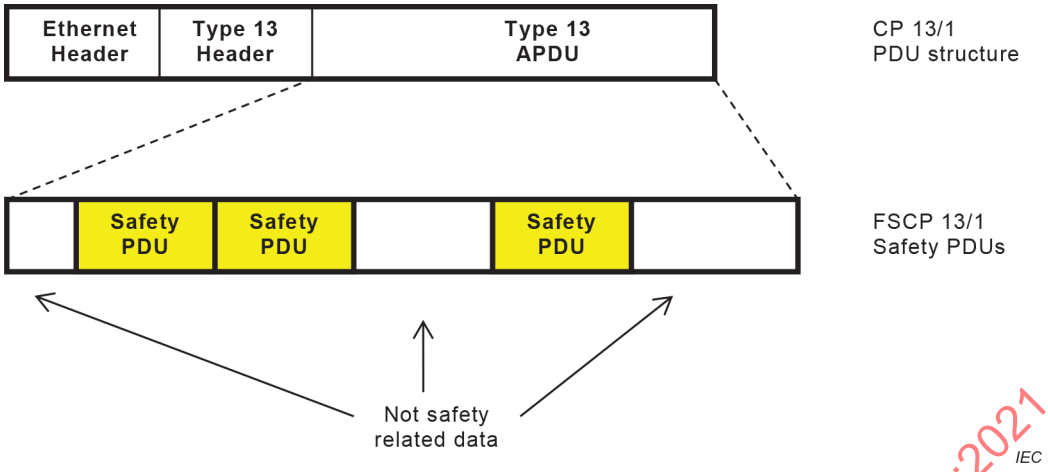


Figure 18 – Safety PDUs inside a CP 13/1 PDU

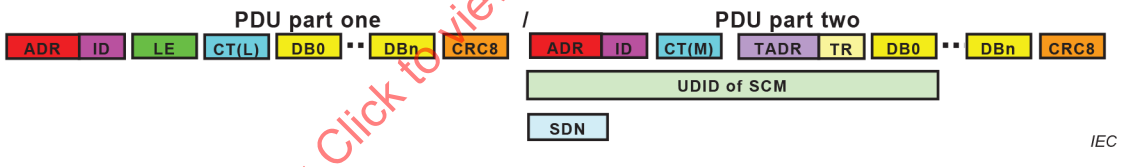
7.1.1.2 Basic Safety PDU

The structure of the Basic Safety PDU shall be as shown in Figure 19 and Figure 20. The Basic Safety PDU shall consist of two sub parts and is able to transport up to 240 octets of payload data. For the residual error rate with respect to the safety PDU length, see 9.5.6.

Depending on payload length two types of Basic Safety PDUs shall be used:

– 1 – 8 octets of payload: PDU format as shown in Figure 19 shall be used (with an 8 bit CRC);

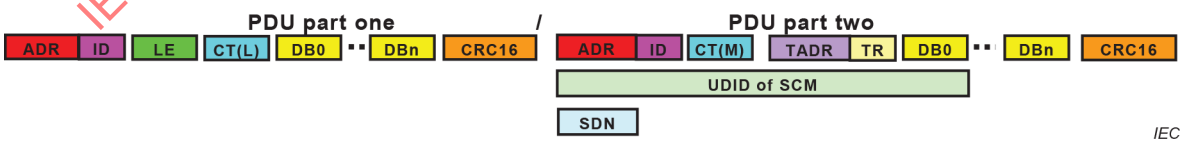
– From 9 octets of payload: PDU format as shown in Figure 20 shall be used (with a 16 bit CRC).



Key

n number of payload octets

Figure 19 – Basic Safety PDU for *n* = 0 – 8 octet payload data



Key

n number of payload octets

Figure 20 – Basic Safety PDU from 9 octet payload data

Table 4 specifies the Basic Safety PDU. The PDU shall be constructed in two parts, called PDU part one and PDU part two. The data of PDU part two of SPDO and SSDO telegrams (not SNMT telegrams) are additionally coded with the UDID of the SCM using a logical XOR operation (see 7.1.10).

Table 4 – Basic Safety PDU format

Octet offset [a]	Bit offset							
	7	6	5	4	3	2	1	0
0	ADR (Bit 0-7), see 7.1.2							
1	ID, see 7.1.3						ADR (8,9)	
2	LE, see 7.1.4							
3	CT (Bit 0-7), see 7.1.5							
4 – n+3	DB 0 to DB n, see 7.1.6							
n+4 – n+4+o	CRC-8 / CRC-16, see 7.1.7							
n+5+o	ADR (Bit 0-7) XOR SDN (Bit 0-7)							
n+6+o	ID						ADR (8,9) XOR SDN (8,9)	
n+7+o	CT (Bit 8-15)							
n+8+o	TADR (Bit 0-7), see 7.1.8							
n+9+o	TR, see 7.1.9						TADR (8,9)	
n+10+o – 2n+9+o	DB 0 – DB n							
2n+10+o – 2n+10+2o	CRC-8 / CRC-16							
NOTE The grey colored lines indicates the data of PDU part two.								
[a] Relative to start of the Basic Safety PDU. n shall be the number of payload octets in the field DB 0 to DB n, where $0 \leq n \leq 240$. o shall be the CRC correction offset, if $0 \leq n \leq 8$ then o = 0 (CRC 8), if $9 \leq n \leq 240$ then o = 1 (CRC-16).								

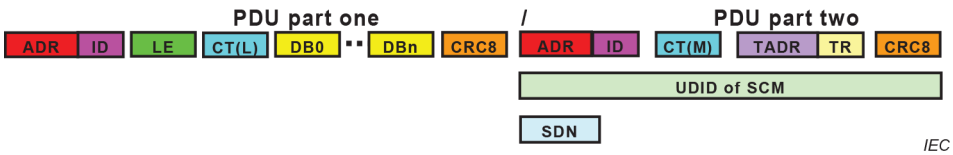
The fields of the Basic Safety PDU are further defined from 7.1.2 onwards.

7.1.1.3 Slim Safety PDU

The structure of the Slim Safety PDU shall be as shown in Figure 21 and Figure 22. The Slim Safety PDU shall consist of two sub parts and is able to transport up to 240 octets of payload data. For the residual error rate with respect to the safety PDU length, see 9.5.6.

Depending on payload length two types of Slim Safety PDUs shall be used:

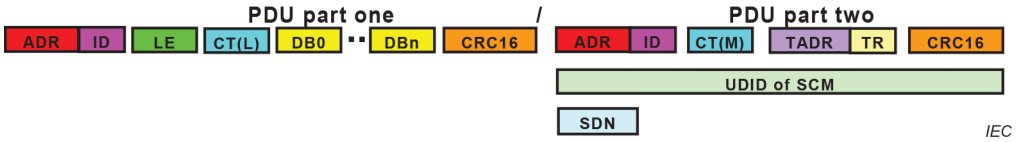
– 1 – 8 octets of payload: PDU format as shown in Figure 21 shall be used (with an 8 bit CRC);
– From 9 octets of payload: PDU format as shown in Figure 22 shall be used (with a 16 bit CRC).



Key

n number of payload octets

Figure 21 – Slim Safety PDU for *n* = 0 – 8 octet payload data



Key

n number of payload octets

Figure 22 – Slim Safety PDU from 9 octet payload data

Table 5 specifies the Slim Safety PDU. The PDU shall be constructed in two parts, called PDU part one and PDU part two. The data of PDU part two are additionally coded with the UDID of the SCM using a logical XOR operation (see 7.1.10).

Table 5 – Slim Safety PDU format

Octet offset [a]	Bit offset							
	7	6	5	4	3	2	1	0
0	ADR (Bit 0-7), see 7.1.2							
1	ID, see 7.1.3						ADR (8,9)	
2	LE, see 7.1.4							
3	CT (Bit 0-7), see 7.1.5							
4 – n+3	DB 0 to DB n, see 7.1.6							
n+4 – n+4+o	CRC-8 / CRC-16, see 7.1.7							
n+5+o	ADR (Bit 0-7) XOR SDN (Bit 0-7)							
n+6+o	ID						ADR (8,9) XOR SDN (8,9)	
n+7+o	CT (Bit 8-15)							
n+8+o	TADR (Bit 0-7), see 7.1.8							
n+9+o	TR, see 7.1.9						TADR (8,9)	
n+10+o – n+10+2o	CRC-8 / CRC-16							

NOTE The grey colored lines indicates the data of PDU part two.

[a] Relative to start of the Slim Safety PDU. n shall be the number of payload octets in the field DB 0 to DB n, where $0 \leq n \leq 240$. o shall be the CRC correction offset, if $0 \leq n \leq 8$ then o = 0 (CRC 8), if $9 \leq n \leq 240$ then o = 1 (CRC-16).

The fields of the safety PDU are further defined from 7.1.2 onwards.

7.1.1.4 Safety PDU with 40 bit counter

FSCP 13/1 may use a 40 bit counter. The 40 bits are built by the 16 bits of explicitly reserved CT fields and additional 24 bits that are encoded onto the PDU part two structure. Only SPDO data telegrams that are not used for time synchronization shall be used for 40 bit counters.

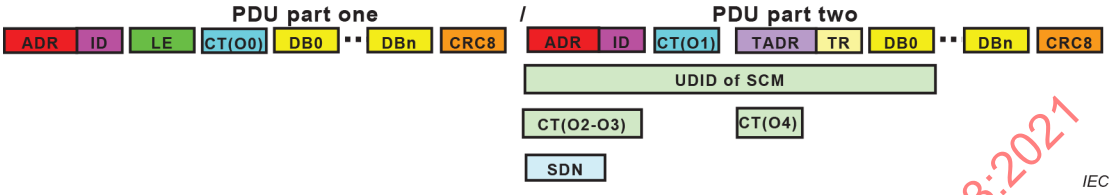
Because the encoding for PDUs with a 40 bit counter differs from that of the ones with a 16-bit counter, an initialization phase is needed, that is described in more detail in 7.6.6.

The structure of the Safety PDU with 40 bit counter shall be as shown in Figure 23 and Figure 24. The Safety PDU with 40 bit counter shall consist of two sub parts and is able to transport up to 240 octets of payload data. For the residual error rate with respect to the safety PDU length, see 9.5.6.

Depending on payload length two types of Safety PDUs with 40 bit counter shall be used:

– 1 – 8 octets of payload: PDU format as shown in Figure 23 shall be used (with an 8 bit CRC);

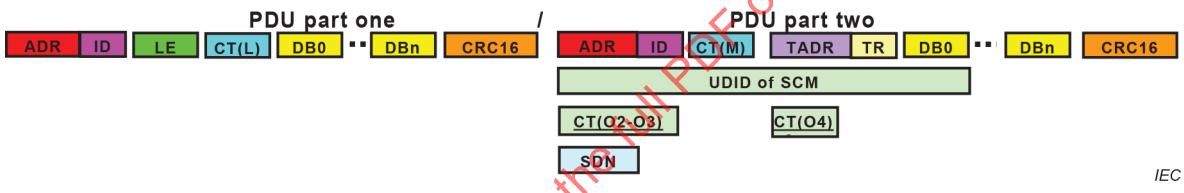
– From 9 octets of payload: PDU format as shown in Figure 24 shall be used (with a 16 bit CRC).



Key

n number of payload octets

Figure 23 – Safety PDU with 40 bit counter for *n* = 0 – 8 octet payload data



Key

n number of payload octets

Figure 24 – Safety PDU with 40 bit counter from 9 octet payload data

Table 6 specifies the Safety PDU with 40 bit counter. The PDU shall be constructed in two parts, called PDU part one and PDU part two. The data of PDU part two are additionally coded with the UDID of the SCM and the bits 16 to 39 of the CT using logical XOR operations (see 7.1.10 and 7.1.5).

Table 6 – Safety PDU with 40 bit counter

Octet offset [a]	Bit offset							
	7	6	5	4	3	2	1	0
0	ADR (Bit 0-7), see 7.1.2							
1	ID, see 7.1.3						ADR (8,9)	
2	LE, see 7.1.4							
3	CT (Bit 0-7), see 7.1.5							
4 – n+3	DB 0 to DB n, see 7.1.6							
n+4 – n+4+o	CRC-8 / CRC-16, see 7.1.7							
n+5+o	ADR (Bit 0-7) XOR CT (BIT16-23) XOR SDN (Bit 0-7)							
n+6+o	ID XOR CT (26-31)						ADR (8,9) XOR CT (24,25) XOR SDN (8,9)	
n+7+o	CT (Bit 8-15)							
n+8+o	TADR (Bit 0-7) XOR CT (BIT32-39), see 7.1.8							
n+9+o	TR, see 7.1.9						TADR (8,9)	
n+10+o – n+10+2o	CRC-8 / CRC-16							
NOTE The grey colored lines indicates the data of PDU part two.								
[a] Relative to start of the Safety PDU with 40 bit CT. n shall be the number of payload octets in the field DB 0 to DB n, where 0 ≤ n ≤ 240. o shall be the CRC correction offset, if 0 ≤ n ≤ 8 then o = 0 (CRC 8), if 9 ≤ n ≤ 240 then o = 1 (CRC-16).								

The fields of the safety PDU are further defined from 7.1.2 onwards.

7.1.2 Address field (ADR)

The ADR field defines the Safety Address (SADR) of the Safety Node (SN). The field shall be a 10 bit value. ADR shall support a value range from 1 to 1 023 (0 is reserved and shall not be used).

The ADR field within PDU part two shall also be used to carry the Safety Domain Number (SDN) information by using a logical XOR operation (see 7.1.10).

The ADR field shall contain the SADR of the SN from which the PDU is sent (source address) or for which the PDU is designated (destination address). The source address shall be the SADR of the SN which sends the PDU; the destination address shall be the SADR of the SN which the telegram is sent to.

According to the content of the ADR field of the PDU part one, the non safe communication layer is able to transport the data to the designated destination node without additional analyzing the ID field or other PDU attributes.

In case of a Safety PDU with 40 bit counter the ADR field within PDU part two shall also be used to carry parts of the CT value by using a logical XOR operation (see 7.1.5).

7.1.3 PDU identification field (ID)

The ID field shall identify the PDU type. Table 7 specifies the coding of the ID field. Reserved codes may be used in future editions of this document. Currently, they shall not be used but shall be ignored if received. For identification of the PDU type, bits 5, 6 and 7 shall be used. Each PDU type has different telegram types which shall be specified bit 2, 3 and 4 of the ID field.

Table 7 – PDU identification field (ID)

Bit Offset								PDU type
7	6	5	4	3	2	1	0	
0	0	0	0	0	x [a]	bit 9 of the address field	bit 8 of the address field	Not allowed (illegal PDU)
0	x	x	x	x	x			reserved
1	0	0	x	x	x			reserved
1	0	1	x	x	x			SNMT
1	1	0	x	x	x			SPDO
1	1	1	x	x	x			SSDO

[a] An x within this table specifies the telegram type for a specific PDU type.

Table 8 specifies all valid ID field combinations.

Table 8 – Used ID field combinations

Bit Offset								Telegram type
7	6	5	4	3	2	1	0	
1	0	1	0	0	0	bit 9 of the address field	bit 8 of the address field	SNMT_Request_UDID
1	0	1	0	0	1			SNMT_Response_UDID
1	0	1	0	1	0			SNMT_Assign_SADR
1	0	1	0	1	1			SNMT_SADR_assigned
1	0	1	1	0	0			SNMT_Service_Request
1	0	1	1	0	1			SNMT_Service_Response
1	0	1	1	1	1			SNMT_SN_reset_guarding_SCM
1	1	0	0	0	x[a]			SPDO_Data_Only
1	1	0	0	1	x[a]			SPDO_Data_with_Time_Request
1	1	0	1	0	x[a]			SPDO_Data_with_Time_Response
1	1	1	0	0	0			SSDO_Service_Request
1	1	1	0	0	1			SSDO_Service_Response
1	1	1	0	1	0			SSDO_Service_Request_Slim
1	1	1	0	1	1			SSDO_Service_Response_Slim

[a] connection valid bit, see 7.2.2.

SSDO and SNMT PDUs shall be divided into request and response telegrams as shown in Table 9. Bit 2 of the ID field shall be used to identify whether the telegram is a request or a response. Bits 0 and 1 of the octet where the ID field is located are used by the address field (see Table 4).

Table 9 – Request / response identification

Bit Offset								Description
7	6	5	4	3	2	1	0	
x	x	x	x	x	0			SSDO / SNMT Request
x	x	x	x	x	1			SSDO / SNMT Response

7.1.4 Length field (LE)

The LE field specifies the number of payload data octets within the PDU. The valid value range shall be 0 – 240. The value of LE shall determine the type of CRC used for the PDU as specified in Table 10.

Table 10 – Type of CRC depending on LE

LE value	Used CRC
0 – 8	CRC-8
9 – 240	CRC-16
241 – 255	reserved

7.1.5 Consecutive Time field (CT)

7.1.5.1 General

The time base of the CT field shall be defined in SOD object 1200h sub index 03h (see 7.5.4.13).

7.1.5.2 Consecutive Time field (CT) – 16 bit

The Consecutive Time (CT) field shall be divided in an LSB part which is sent in PDU part one and an MSB part which is sent in PDU part two. Both octets build the CT data set with a number range of 0 – 65 535. The CT field shall be used for the internal timer value of the SN when sending the PDU in case of an SPDO (see 7.2), in case of an SSDO, the CT field describes the SOD access request number (see 7.3).

7.1.5.3 Consecutive Time field (CT) – 40 bit

For the lower 16 bits the same fields as for the 16 bit CT shall be used, i.e. bit 0-7 shall be transmitted in the CT field of PDU part one, and bit 8-15 shall be transmitted via the CT field in PDU part two.

Bit 16-31 shall be encoded with a XOR operation on the first two bytes of PDU part two (ADR and ID) and bit 32-39 shall be encoded with a XOR operation on the lower 8 bit of the TADDR in PDU part two.

7.1.6 Payload data field (DB0 to DBn)

This field shall contain the payload data. A payload data length from 0 up to 240 octets shall be supported. Payload data length of 0 may be used for time synchronization services (see 7.2.4, 7.2.5).

7.1.7 Cyclic Redundancy Check field (CRC-8 / CRC-16)

The CRC field shall contain the cyclic redundancy check value for the PDU part it belongs to which shall be calculated using one of the polynoms referenced in Table 11. The type (CRC-8 or CRC-16) depends on the values of the length field (LE) and the PDU identification field (ID).

Table 11 – CRC polynoms for SPDUs

Intended use	Polynom	Polynom	Polynom according [35]
Payload data up to 8 octets	$X^8+X^5+X^3+X^2+X+1$	$12F_{hex}$	0x97
Payload data from 9 octets, SLIM SSDO services only	$X^{16}+X^{14}+X^{12}+X^{11}+X^8+X^5+X^4+X^2+1$	15935_{hex}	0xAC9A
Payload data from 9 octets, all services except SLIM SSDO	$X^{16}+X^{14}+X^{13}+X^{12}+X^{10}+X^8+X^6+X^4+X^3+X+1$	$1755B_{hex}$	0xBAAD

The following rules for CRC generation shall apply:

1) Seed value is 0 for CRC-8 as well as for each CRC-16 polynom;
2) The most significant bit of the first octet is shifted in first;
3) The CRC shall be calculated over all fields of the PDU part excluding the CRC;
   i) PDU part one: ADR, ID, LE, CT(L/0-7), DB0–DBn;
   ii) PDU part two: ADR, ID, CT(H/8-15), TADR, TR, DB0–DBn.

The CRC-8 shall comply with the encoding rules of an UNSIGNED8 data type.

The CRC-16 shall comply with the encoding rules of an UNSIGNED16 data type.

An example code for calculating the CRC can be found in Clause A.1.

7.1.8 Time Request Address field (TADR)

The TADR field specifies:

– the Safety Address (SADR) of an SN which responds to the Time Synchronization Request in case of SPDO (see 7.2), or
– the SADR (source or destination) in case of SSDO (see 7.3) and SNMT (see 7.4).

7.1.9 Time Request Distinctive Number field (TR)

The TR field shall be a number in the range 0 – 63, which shall be mirrored by the device in order to distinguish this message from other time request telegrams.

Additionally, this field shall be used to negotiate the Safety PDU feature bits (see 7.6.6.2).

7.1.10 UDID of SCM coding (UDID of SCM)

The 6 octets UDID of the SCM shall be coded by a bitwise exclusive or operation with the first 6 octets of the payload data of PDU part two of SPDO and SSDO telegrams. SNMT telegrams shall not be coded in this way.

The UDID of the SCM is known for every device (see 7.4.3.12), the SN shall test every received SPDO and SSDO telegrams whether the correct UDID was used for coding the telegrams.

NOTE By doing this, mismatches in the domain separation (see 6.1.3.3.3) can be detected.

7.2 Safety Process Data Object (SPDO)

7.2.1 General

Safety Process Data Objects (SPDOs) shall be used for transmitting safety-critical process data as well as time synchronization request/response data from an SPDO producer to the SPDO consumers. Reception of an SPDO by an SPDO consumer shall be determined by the address of the SPDO producer.

Only Basic Safety PDUs shall be used for SPDOs.

7.2.2 SPDO telegram types

The ID field (bit 2-4) shall identify the SDPO telegram type. Three telegram types shall be defined:

– data only telegram,
– data with Time Request telegram, and
– data with Time Response telegram.

Table 12 specifies the coding of the ID field.

Table 12 – SPDO telegram types (ID field, bits 2, 3 and 4)

Bit number of ID field								PDU Type
7	6	5	4	3	2	1	0	
1	1	0	0	0	X[a]	bit 8,9 of the Address field		SPDO_Data_Only
1	1	0	0	1	X[a]			SPDO_Data_with_Time_Request
1	1	0	1	0	X[a]			SPDO_Data_with_Time_Response
1	1	0	1	1	X[a]			Reserved

[a] Bit 2 shall be used as a "connection valid bit". If the time synchronization of all RxSPDOs to be synchronized over the TxSPDO is ok and the payload data in the TxSPDO is valid, the bit shall be set to 1_b, otherwise it shall be set to 0_b.

Example Connection valid bit

A SN has 3 RxSPDOs and 2 TxSPDOs. RxSPDO 1 and 2 are synchronized using TxSPDO 1, RxSPDO 3 is synchronized using TxSPDO 2.

The synchronization of RxSPDO 1 is not ok.

The synchronization of RxSPDO 2 and 3 is ok.

The TxSPDO 1 is sent with connection valid bit set to 0_b (RxSPDO 1 not ok & RxSPDO 2 ok -> not ok).

The TxSPDO 2 is sent with the connection valid bit set to 1_b (RxSPDO 3 ok).

7.2.3 Data Only telegram

The Data Only telegram (see Figure 25) shall be used to transmit process data from an SN (SPDO producer) without a time synchronization request or response. The data can be consumed by any other SN (SPDO consumer).

Together with the data (Data), the SPDO producer shall provide its SN address (ADR) as well as the number of payload data octets transmitted (LE) and the current value of its internal timer (CT).

In an SPDO communication with 40 bit CT values, additionally the upper 24 bit are encoded on various parts of the telegram. See 7.1.5.3 for details on the encoding.

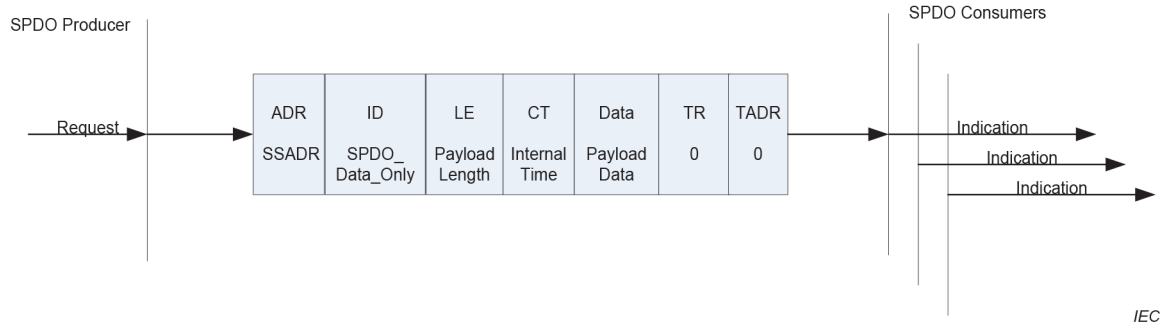


Figure 25 – SPDO_Data_Only telegram

Table 13 specifies the fields of an SPDO_Data_Only telegram.

Table 13 – Fields of SPDO_Data_Only telegram

Field	Information	Content / Value
ADR	Address of SPDO producer	SSADR
ID	SPDO_Data_only	11000xxxb
LE	Number of payload data octets	0 – 240 (see 7.1.4)
CT	Internal time value of SPDO producer	Time
Data	Payload data	Data
TR	Not used/Feature request (see 7.6.6.2)	0
TADR	Not used	0

7.2.4 Data with Time Request telegram

The Data with Time Request telegram (see Figure 26) shall be used to transmit process data from an SN (SPDO producer) with a Time Synchronization Request to another SN addressed by TADR. The data can be consumed by any other SN (SPDO consumer).

Together with the data (Data), the SPDO producer provides its SN Address (ADR), the telegram type (ID) as well as the number of payload data octets (LE) transmitted, the current value of its internal timer (CT), a counter value for indicating the number of time requests (TR) and the address of the SN which is requested to provide its current internal time (TADR).

In an SPDO communication with 40 bit CT values, the ID field of PDU part two shall be encoded with a XOR operation with the confirmation to the SPDO feature request, by the SPDO producer, which mirrors the request information.

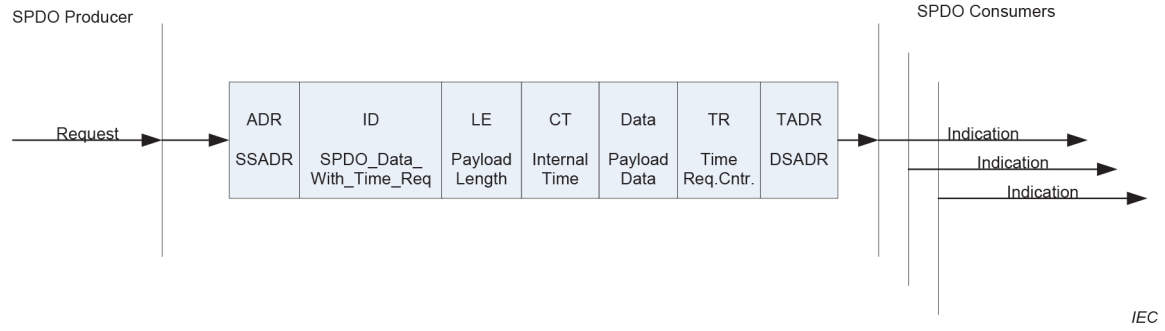


Figure 26 – SPDO_Data_with_Time_Request telegram

Table 14 specifies the fields of an SPDO_Data_with_Time_Request telegram.

Table 14 – Fields of SPDO_Data_with_Time_Request telegram

Field	Information	Content / Value
ADR	Address of SPDO producer	SSADR
ID	SPDO_Data_with_Time_Request	11001xxxb
LE	Number of payload data octets	0 – 240 (see 7.1.4)
CT	Internal time value of SN	Time
Data	Payload data	Data
TR	Internal Time Request Counter	0 – 63
TADR	Address of SN from which time information is requested	DSADR

7.2.5 Data with Time Response telegram

The Data with Time Response telegram (see Figure 27) shall be used to transmit process data from an SN (SPDO producer) together with the internal timer value of the requested node. The data can be consumed by any other SN (SPDO consumer).

Together with the data (Data), the SPDO producer provides its SN address (ADR), the telegram type (ID) as well as the number of payload data octets (LE) transmitted, the current value of its internal timer (CT), the counter value for indicating the number of the time requests (TR) and the address of the SN which has issued the request (TADR).

In an SPDO communication with 40 bit CT values no additional encoding in the frame takes place.

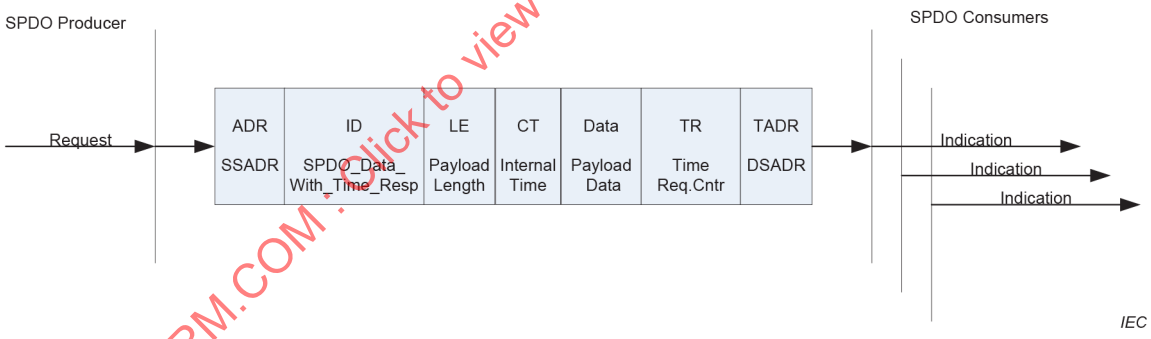


Figure 27 – SPDO_Data_with_Time_Response telegram

Table 15 specifies the fields of an SPDO_Data_with_Time_Response telegram.

Table 15 – Fields of SPDO_Data_with_Time_Response telegram

Field	Information	Content / Value
ADR	Address of SPDO producer	SSADR
ID	SPDO_Data_with_Time_Response	11010xxxb
LE	Number of payload data octets	0 – 240 (see 7.1.4)
CT	Internal time value of SPDO producer	Time
Data	Payload data	Data
TR	Time Request Counter value of corresponding request	0 – 63
TADR	Address of the SN which issued the time information request	DSADR

7.3 Safety Service Data Object (SSDO)

7.3.1 General

Safety Service Data Objects (SSDOs) shall be used for accessing the Object Directory (SOD) of an SSDO-Server by an SSDO-Client.

SSDO telegrams shall be identified by bits 7, 6 and 5 (111b) of the ID. Bit 4 of the ID shall identify an SOD-access (0b). Bit 3 shall identify the type of Safety PDU used which shall be either a Basic Safety PDU (0b) or a Slim Safety PDU (1b) depending on the type of SSDO service. Bit 2 shall differentiate between request (0b) and response (1b) telegrams.

7.3.2 SSDO telegram types

The ID field identifies the telegram. The following telegram types are specified:

– SSDO Service Request
– SSDO Service Response

Table 16 specifies the coding of the ID field.

Table 16 – SSDO telegram types (ID field, bits 2, 3 and 4)

Bit number of ID field								Telegram type
7	6	5	4	3	2	1	0	
1	1	1	0	0	0			SSDO_Service_Request
1	1	1	0	0	1			SSDO_Service_Response
1	1	1	0	1	0			SSDO_Service_Request_Slim
1	1	1	0	1	1			SSDO_Service_Response_Slim

The SSDO Service Request telegram (SSDO_Service_Request) shall be used by an SSDO client to access the SOD of an SSDO server. With an SSDO_Service_Request, an SSDO client shall provide the server SN address (ADR), the length of the payload data (LE) and the Access Request Number SANo (CT field). The first payload data octet shall be used as "Command Octet" (SACmd, see Table 17) and shall specify access type (Read/Write), transfer status (successful/abort), transfer type (expedited/segmented), toggle bit, initialize or last segment of the transfer and transfer mode (normal/block). In data octets 1 and 2, the index of the accessed object dictionary entry, in data octet 3 the sub index of the accessed entry shall be specified. The SN address of the SSDO client shall be provided in TADR.

The SSDO Service Response telegram (SSDO_Service_Response) shall be used by an SSDO server to respond to a request from an SSDO client. The SOD Access Request Number (SANo) shall be unique for each connection and incremented by one with every new SOD Access Request telegram. The response telegram shall contain the SANo value of the corresponding service request.

The maximum length of payload data in a single telegram shall be 240 octets.

The SOD Access Command (SACmd) bit encoding is specified in Table 17.

Table 17 – SOD Access Command (SACmd) – bit coding

Bit No.	Name	Value	Meaning
0	Access Type	0	SOD Read Access
		1	SOD Write Access
1	Preload	0	No Preload in SSDO transfer (default)
		1	Enable Preload in SSDO transfer
2	Abort transfer	0	Successful transfer
		1	Abort transfer
3	Segmentation	0	Expedited SOD access
		1	Segmented SOD access
4	Toggle	0	This bit shall alternate for each subsequent segment and not change its value in case of a repeated transmission. The first segment shall have the toggle bit set to 0. The toggle bit shall be equal for the request and the response telegram.
		1	
5	Initiate transfer	0	No initiate transfer
		1	Initiate transfer
6	End segment transfer	0	More segments to transfer
		1	No more segments to transfer
7	Block transfer (No longer in use)	0	Reserved

7.3.3 SSDO services and protocols

There are two types of SDO services using different Safety PDUs:

– The normal SSDO service type using the Basic Safety PDU may be used for all SSDO services.

– The slim SSDO service type using the Slim Safety PDU may only be used to download initializing data to a SN. The lack of having the data twice in the Safety PDU shall be compensated by verifying the data. This shall be done by only switching the SN to operational if the SOD CRC is correct.

The following SSDO services shall be defined:

– SSDO Download, which is subdivided into:
  - SSDO Download Initiate,
  - SSDO Download Initiate with Preload,
  - SSDO Download Segment,
  - SSDO Download Segment with Preload,
  - SSDO End Segment;
– SSDO Upload, which is subdivided into:
  - SSDO Upload Initiate,
  - SSDO Upload Initiate with Preload,
  - SSDO Upload Segment,
  - SSDO Upload Segment with Preload,
  - SSDO End Segment;
– SSDO Abort.

Figure 28 defines the SSDO segmented and expedited download protocols as well as their preload counterparts.

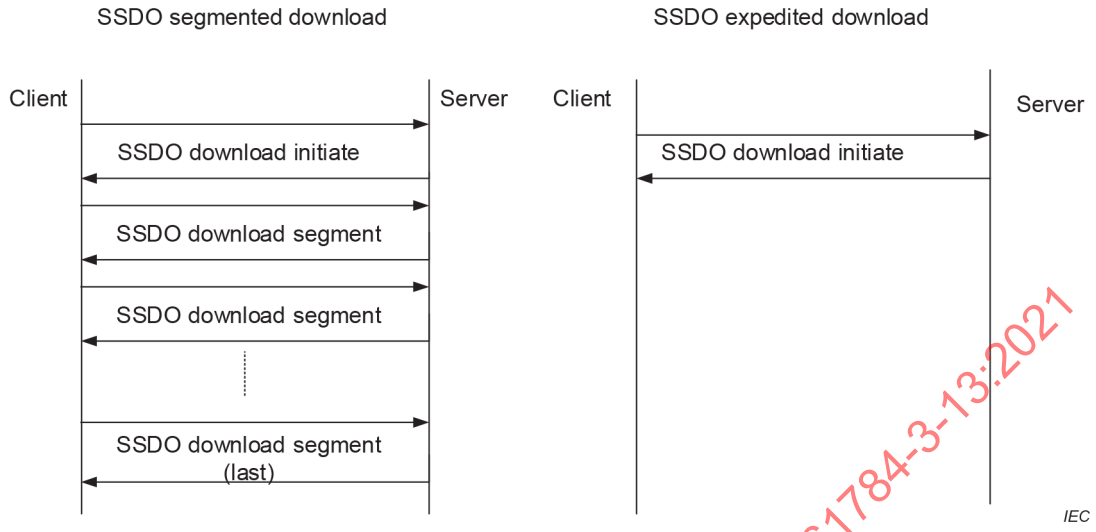


Figure 28 – SSDO download protocols

Figure 29 defines SSDO segmented and expedited upload protocols.

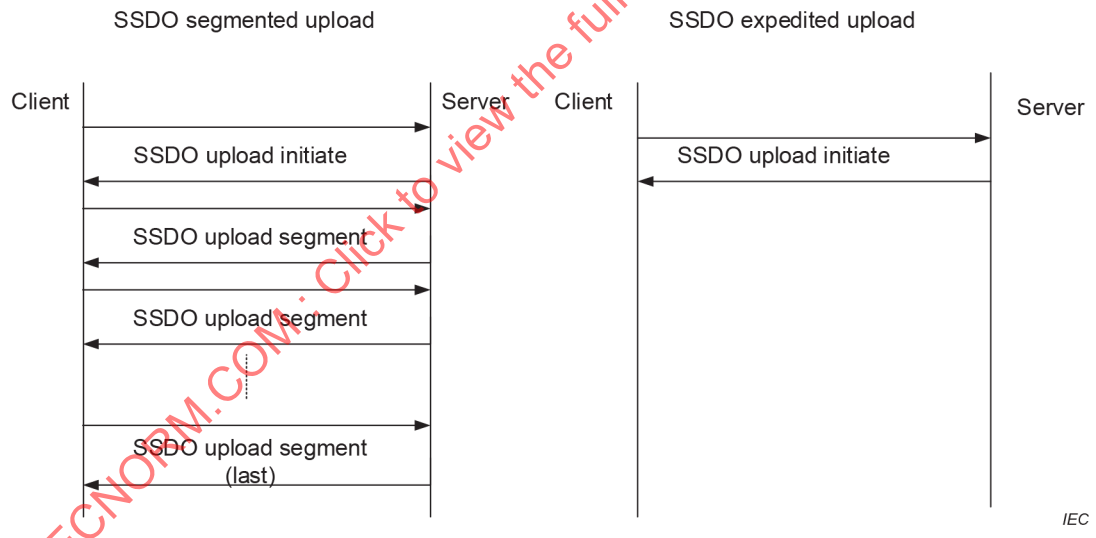


Figure 29 – SSDO upload protocols

7.3.4 SSDO Download Initiate

Figure 30 specifies the download initiate protocol.

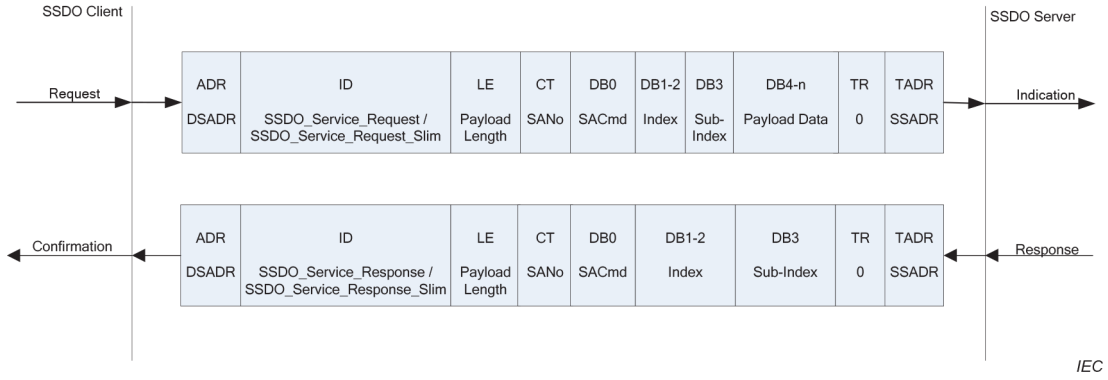


Figure 30 – SSDO Download Initiate protocol

Table 18 specifies the fields of a Download Initiate SSDO_Service_Request telegram.

Table 18 – Fields of Download Initiate SSDO_Service_Request telegram

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Length of payload data	4 – 240 (see 7.1.4)
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited or segmented	00100001b or 00101001b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
DB4 – DBn	Payload data (In case of a segmented download, DB4-DB7 contains the data size of the payload data to be downloaded. If the data size is 0, the size is not indicated.)	Payload data
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 19 specifies the fields of a Download Initiate SSDO_Service_Response telegram.

Table 19 – Fields of Download Initiate SSDO_Service_Response telegram

Field	Information of	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Length of payload data	4
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited or segmented	00100001b or 00101001b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
TR	Not used	0
TADR	Address of SSDO Server	SSADR

7.3.5 SSDO Download Segment

Figure 31 specifies the Download Segment protocol.

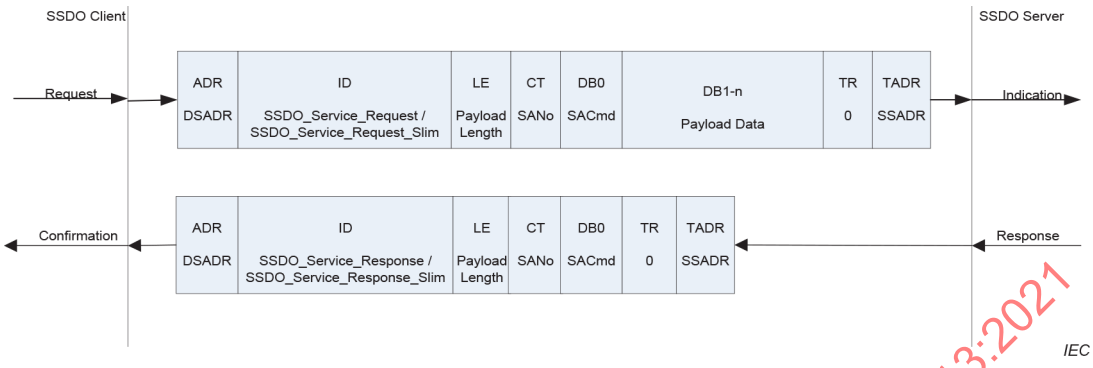


Figure 31 – SSDO Download Segment protocol

Table 20 specifies the fields of a Download Segment SSDO_Service_Request telegram.

Table 20 – Fields of Download Segment SSDO_Service_Request telegram

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Length of payload data	1 – 240 (see 7.1.4)
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment or end segment	000x1001b or 010x1001b
DB1 – DBn	Payload data	Payload data
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 21 specifies the fields of a Download Segment SSDO_Service_Response telegram.

Table 21 – Fields of Download Segment SSDO_Service_Response telegram

Field	Information	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Length of payload data	1
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment or end segment	000x1001b or 010x1001b
TR	Not used	0
TADR	Address of SSDO Server	SSADR

7.3.6 SSDO Download Initiate with Preload

Figure 32 specifies the download initiate protocol with preload.

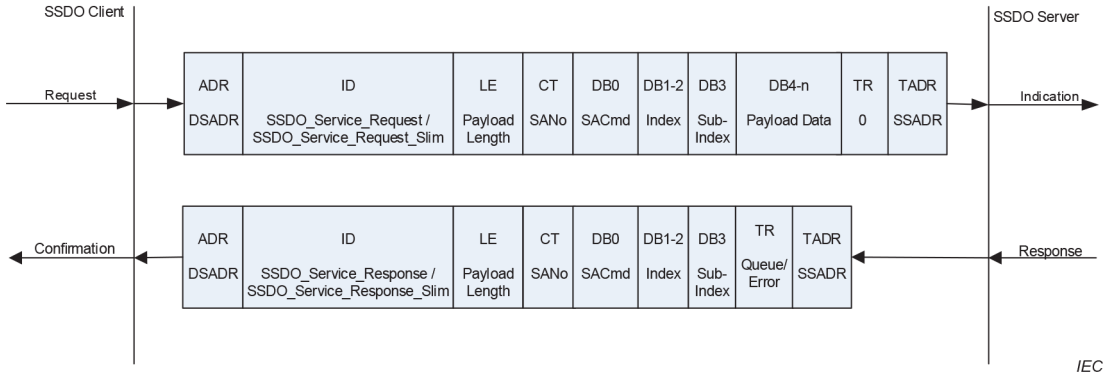


Figure 32 – SSDO Download Initiate protocol with Preload

Table 22 specifies the fields of a Download Initiate SSDO_Service_Request telegram with Preload.

Table 22 – Fields of Download Initiate SSDO_Service_Request telegram with Preload

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Length of payload data	4 – 240 (see 7.1.4)
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited or segmented with preload	00100011b or 00101011b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
DB4 – DBn	Payload data (In case of a segmented download, DB4-DB7 contains the data size of the payload data to be downloaded. If the data size is 0, the size is not indicated.)	Payload data
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 23 specifies the fields of a Download Initiate SSDO_Service_Response telegram with Preload.

Table 23 – Fields of Download Initiate SSDO_Service_Response telegram with Preload

Field	Information of	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Length of payload data	4
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited or segmented	00100011b or 00101011b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
TR	Bit 2-5 – Queue size Bit 6-7 – Error indicator	0 – 15 0xC

Field	Information of	Content / Value
TADR	Address of SSDO Server	SSADR

7.3.7 SSDO Download Segment with Preload

Figure 33 specifies the Download Segment protocol with preload.

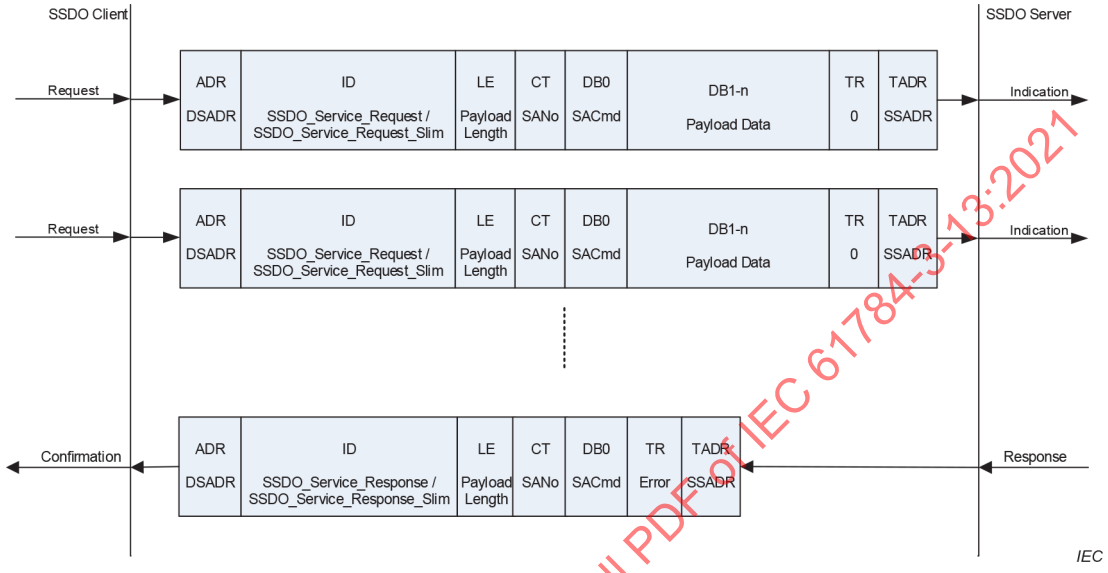


Figure 33 – SSDO Download Segment protocol with Preload

Table 24 specifies the fields of a Download Segment SSDO_Service_Request telegram with Preload.

Table 24 – Fields of Download Segment SSDO_Service_Request telegram with Preload

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Length of payload data	1 – 240 (see 7.1.4)
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment or end segment	000x1011b or 010x1011b
DB1 – DBn	Payload data	Payload data
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 25 specifies the fields of a Download Segment SSDO_Service_Response telegram with Preload.

Table 25 – Fields of Download Segment SSDO_Service_Response telegram with Preload

Field	Information	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Length of payload data	1
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment or end segment	000x1011b or 010x1011b
TR	Error indicator	Error: 1xxxxxxxb Success: 0xxxxxxxb
TADR	Address of SSDO Server	SSADR

7.3.8 SSDO Upload Initiate

Figure 34 specifies the Upload Initiate protocol. The Upload Initiate telegram shall always be expedited. Based on the response the SSDO client shall continue with expedited or segmented transfer.

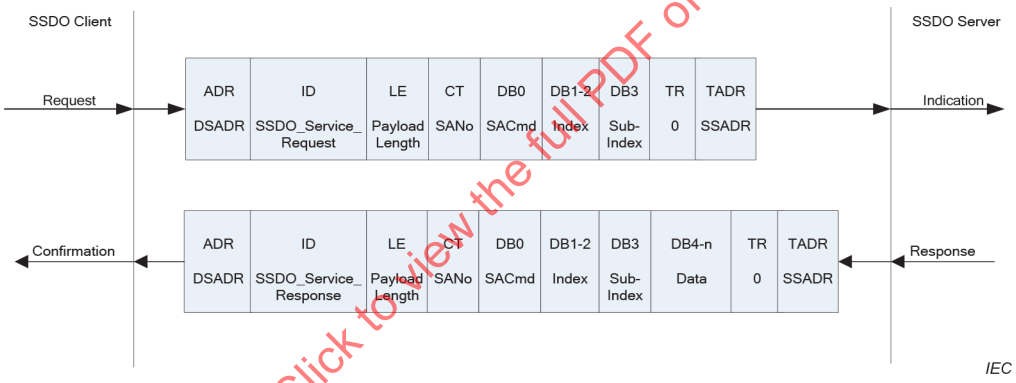


Figure 34 – SSDO Upload Initiate protocol

Table 26 specifies the fields of an Upload Initiate SSDO_Service_Request telegram.

Table 26 – Fields of Upload Initiate SSDO_Service_Request telegram

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Length of payload data	4
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited	00100000b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 27 specifies the fields of an Upload Initiate SSDO_Service_Response telegram.

Table 27 – Fields of Upload Initiate SSDO_Service_Response telegram

Field	Information	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Length of payload data	4 – 240 (see 7.1.4)
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited or segmented	00100000b or 00101000b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
DB4 – DBn	Payload data (In case of a segmented upload, DB4-DB7 contains the data size of the payload data of the block. If the data size is 0, the size is not indicated.)	Payload data
TR	Not used	0
TADR	Address of SSDO Server	SSADR

7.3.9 SSDO Upload Segment

Figure 35 specifies the Upload Segment protocol. The SSDO client always requests a middle segment. Based on the response, the SSDO client receives from the SSDO Server, the SSDO client shall decide if the SSDO server sent a middle segment or the end segment.

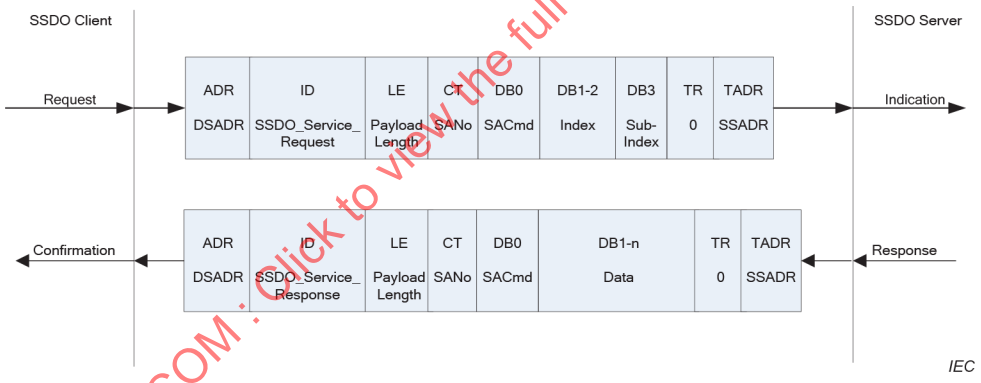


Figure 35 – SSDO Upload Segment protocol

Table 28 specifies the fields of a Upload Segment SSDO_Service_Request telegram.

Table 28 – Fields of Upload Segment SSDO_Service_Request telegram

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Length of payload data	4
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment	000x1000b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 29 specifies the fields of an Upload Segment SSDO_Service_Response telegram.

Table 29 – Fields of Upload Segment SSDO_Service_Response telegram

Field	Information	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Length of payload data	1 – 240 (see 7.1.4)
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment or end segment	000x1000b or 010x1000b
DB1 – DBn	Payload data	Payload data
TR	Not used	0
TADR	Address of SSDO Server	SSADR

7.3.10 SSDO Upload Initiate with Preload

Figure 36 specifies the Block Upload Initiate protocol with preload. The Block Upload Initiate telegram shall always be expedited. Based on the response the SSDO client shall continue with expedited or segmented transfer. The queue size transmitted by the server shall represent the maximum number of requests the server is able to queue for upload.

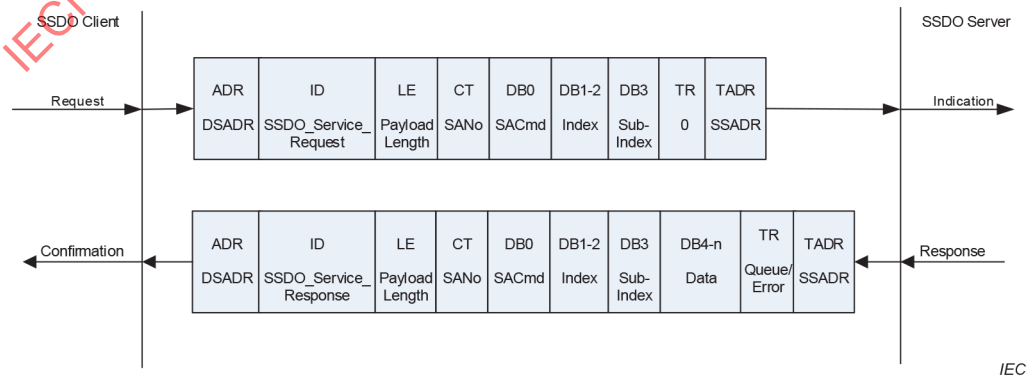


Figure 36 – SSDO Upload Initiate protocol with Preload

Table 30 specifies the fields of an Upload Initiate SSDO_Service_Request telegram with Preload.

Table 30 – Fields of Upload Initiate SSDO_Service_Request telegram with Preload

Field	Information	Content / Value
ADR	Address of SSDO Server to be accessed	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Length of payload data	4
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited	00100010b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 31 specifies the fields of an Upload Initiate SSDO_Service_Response telegram with Preload.

Table 31 – Fields of Upload Initiate SSDO_Service_Response telegram with Preload

Field	Information	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Length of payload data	4 – 240 (see 7.1.4)
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) expedited or segmented	00100010b or 00101010b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub index of SOD entry to be accessed	0 – 255
DB4 – DBn	Payload data (In case of a segmented upload, DB4-DB7 contains the data size of the payload data of the block. If the data size is 0, the size is not indicated.)	Payload data
TR	Bit 2-5 – Queue size Bit 6-7 – Error indicator	0 – 15 0xC
TADR	Address of SSDO Server	SSADR

7.3.11 SSDO Upload Segment with Preload

Figure 37 specifies the Block Upload Segment protocol. The SSDO client always requests a middle segment. Based on the response, the SSDO client receives from the SSDO Server, the SSDO client shall decide if the SSDO server sent a middle segment or the end segment.

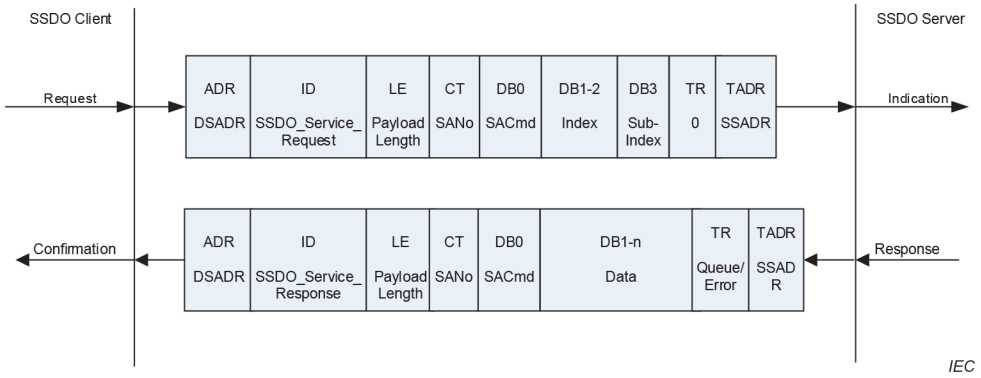


Figure 37 – SSDO Upload Segment protocol with Preload

Table 32 specifies the fields of an Upload Segment SSDO_Service_Request telegram with Preload.

Table 32 – Fields of Upload Segment SSDO_Service_Request telegram with PreLoad

Field	Information	Content / Value
ADR	Address of SSDO Server to be responded	DSADR
ID	SSDO_Service_Requesst	111000xxb
LE	Length of payload data	4
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment	000x1010b
DB1, DB2	Index of SOD entry to be accessed	0 – 65 535
DB3	Sub Index of SOD entry to be accessed	0 – 255
TR	Not used	0
TADR	Address of SSDO Client	SSADR

Table 33 specifies the fields of an Upload Segment SSDO_Service_Response telegram with Preload.

Table 33 – Fields of Upload Segment SSDO_Service_Response telegram with Preload

Field	Information	Content / Value
ADR	Address of SSDO Client to be responded	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Length of payload data	4 – 240 (see 7.1.4)
CT	Echo of SOD Access Request number	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd) middle segment or end segment	000x1010b or 010x1010b
DB1 – DBn	Payload data	Payload data
TR	Error indicator	Error: 1xxxxxxxb Success: 0xxxxxxxb
TADR	Address of SSDO Server	SSADR

7.3.12 SSDO Abort

Figure 38 specifies the SSDO Abort protocol.

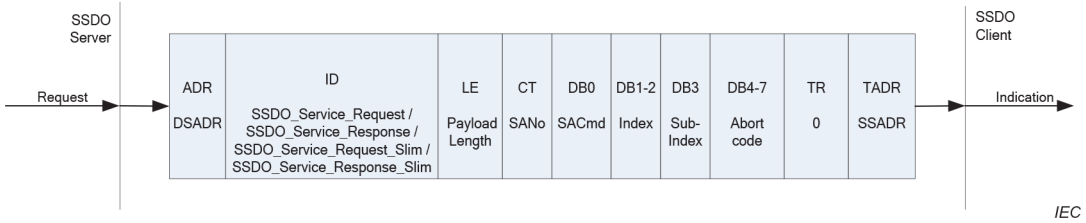


Figure 38 – SSDO Abort protocol

Table 34 specifies the fields of an SSDO Abort telegram.

Table 34 – Fields of SSDO Abort telegram

Field	Information	Content / Value
ADR	Address of the SSDO Client to be informed	DSADR
ID	SSDO_Service_Request SSDO_Service_Response SSDO_Service_Request_Slim SSDO_Service_Response_Slim	111000xxb 111000xxb 111010xxb 111011xxb
LE	Length of payload data	8
CT	SOD Access Request number (SANo)	1 – 65 535, 0 not allowed
DB0	SOD Access Command (SACmd)	00000100b
DB1, DB2	Index of SOD entry	0 – 65 535
DB3	Sub index of SOD entry	0 – 255
DB4 – DB7	Abort code, see Table 35	Abort code
TR	Not used	0
TADR	Address of the SSDO Server initiating the abort	SSADR

The payload data of the SSDO Abort telegram shall contain an error code as specified in Table 35. The abort code shall be encoded as UNSIGNED32 value. Abort codes not listed in Table 35 shall be reserved.

Table 35 – SSDO Abort codes

Abort code [a]	Description
05030000h	Reserved
05040000h	SSDO protocol timed out
05040001h	Client/server Command ID not valid or unknown
05040002h	Invalid block size
05040003h	Invalid sequence number
05040004h	Reserved
05040005h	Out of memory
06010000h	Unsupported access to an object
06010001h	Attempt to read a write-only object
06010002h	Attempt to write a read-only object
06020000h	Object does not exist in the object dictionary
06040041h	Object cannot be mapped to the SPDO
06040042h	The number and length of the objects to be mapped would exceed SPDO length
06040043h	General parameter incompatibility

Abort code [a]	Description
06040047h	General internal incompatibility in the device
06060000h	Access failed due to a hardware error
06070010h	Data type does not match, length of service parameter does not match
06070012h	Data type does not match, length of service parameter too high
06070013h	Data type does not match, length of service parameter too low
06090011h	Sub index does not exist
06090030h	Value range of parameter exceeded (only for write access)
06090031h	Value of parameter written too high
06090032h	Value of parameter written too low
06090036h	Maximum value is less than minimum value
08000000h	General error
08000020h	Data cannot be transferred or stored to the application
08000021h	Data cannot be transferred or stored to the application because of local control
08000022h	Data cannot be transferred or stored to the application because of the present device state
08000023h	Data cannot be transferred or stored to the application because application is busy
[a] The Abort code is shown in hexadecimal notation, as designated by the "h" suffix.	

7.3.13 SSDO Preload

Preload is a method, by which the SSDO server informs the SSDO client of the incoming message buffer size, which allows the client to send messages until the buffer is filled up. The result is a better utilization of asynchronous transport channels.

The SSDO Preload is an optional feature. If the feature is requested by the client but not supported by the server, client and server shall use the conventional segmented transfer techniques to exchange data.

SSDO Preload shall use the TR field, to transport information from SSDO server towards the client regarding the buffer size as well as missing messages.

For SSDO Preload the TR field shall be utilized as defined in Table 36.

Table 36 – TR field usage for SSDO Preload

Bit number of TR field							
7	6	5	4	3	2	1	0
Error		Server buffer size				Used by TADR	

Table 37 specifies the bit fields inside the TR field utilized by SSDO Preload.

Table 37 – Bit fields inside TR field for SSDO Preload

Field	Information
Error	Bits 6-7 shall be set to all high (0xCx) if the package received by the server was not the expected as determined by the SANo numbering. The SANo number of the frame, in which bits 6-7 are set to all high, shall be the number of the last package that was successfully received. The client shall retransmit frames from that point on, beginning with the next frame after that SANo number.
Server buffer size	Bits 2-5 are given by the Initiate Response and define the buffer size on the server side. The value given shall start counting with 0, where 0 is defined as a buffer size of 1, which allows for a maximum buffer size of 16.

7.4 Safety Network Management (SNMT)

7.4.1 General

The Safety Network Management (SNMT) is node oriented and follows a logical Master/Slave structure. The SNMT Master shall be implemented by an SCM or a configuration tool. The following SNMT services shall be provided:

– UDID request,
– node address assignment,
– extended Services, which shall be designated by the DB0 field comprising the following services:
  • communication state control,
  • node guarding,
  • additional node address assignment.

Only Basic Safety PDUs shall be used for SNMT.

7.4.2 SNMT telegram types

The ID field (bits 2 – 4) shall identify the SNMT telegram type. Table 38 specifies the SNMT telegram types:

Table 38 – SNMT telegram types (ID field, bits 2, 3 and 4)

Bit number of ID field								Telegram type	Service
7	6	5	4	3	2	1	0		
1	0	1	0	0	0			SNMT_Request_UDID	UDID request
1	0	1	0	0	1			SNMT_Response_UDID	
1	0	1	0	1	0			SNMT_Assign_SADR	Node address assignment
1	0	1	0	1	1			SNMT_SADR_assigned	
1	0	1	1	0	0			SNMT_Service_Request	Extended Services
1	0	1	1	0	1			SNMT_Service_Response	
1	0	1	1	1	1			SNMT_SN_reset_guarding_SCM	Reset guarding timeout

7.4.3 SNMT services and protocols

7.4.3.1 UDID Request / Response

The UDID Request / Response shall enable an SNMT Master to retrieve the UDID of a specific SNMT Slave. For this, the SNMT Master shall send the UDID Request to one specific SNMT Slave. The SNMT Slave shall reply, even if the value within the ADR field does not match the SNs internal SADR.

NOTE 1 This can happen if there was no prior SADR assignment to the SN.

The TADR field of the response shall contain the SADR of the UDID Request.

NOTE 2 In case the SN does not have a valid SADR and it would reply with its own SADR the response cannot get assigned to the related request correctly.

An error while executing this protocol shall be indicated by UDID 00-00-00-00-00-00 in the response.

Figure 39 specifies the UDID Request protocol.

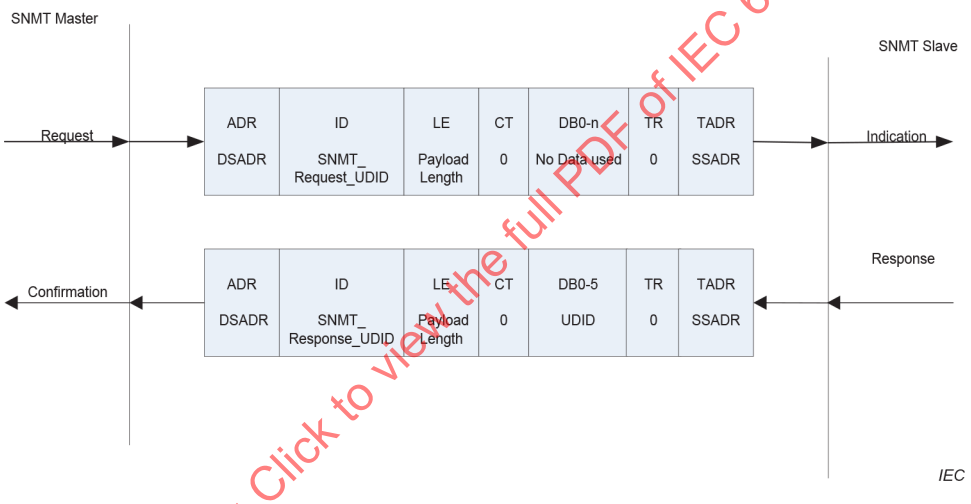


Figure 39 – UDID Request / Response protocol

Table 39 specifies the fields of an SNMT_Request_UDID telegram.

Table 39 – Fields of SNMT_Request_UDID telegram

Field	Information	Content / Value
ADR	Address of the SNMT Slave to be addressed	DSADR
ID	SNMT_Request_UDID	101000xxb
LE	Length of payload data	0
CT	Not used	0
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 40 specifies the fields of an SNMT_Response_UDID telegram.

Table 40 – Fields of SNMT_Response_UDID telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT_Response_UDID	101001xxb
LE	Length of payload data	6
CT	Not used	0
DB0 – DB5	UDID of the SN	UDID
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.4.3.2 SADR assignment

The SADR Assignment shall be used by an SNMT Master to set the SADR of a specific SNMT Slave. The protocol further shall assign the safety domain number (SDN) to the SNMT Slave. The SDN shall be set inside the address field of the PDU part two using a logical XOR operation (see 7.1). A SN shall send a response only if the received UDID matches the SN's own UDID.

If the SDN/SADR requested by the SNMT Master can not be assigned, the SNMT Slave shall reply its current SDN/SADR. The SNMT Master shall then compare the replied SDN/SADR and UDID with the set SDN/SADR and UDID to check whether the assignment was successful.

NOTE Unsuccessful assignments can occur if the nodes SDN/SADR are hard coded using a switch and do not match the SCM value.

An error while executing the service shall be indicated with UDID 00-00-00-00-00-00 in the response.

Figure 40 specifies the SADR Assignment protocol.

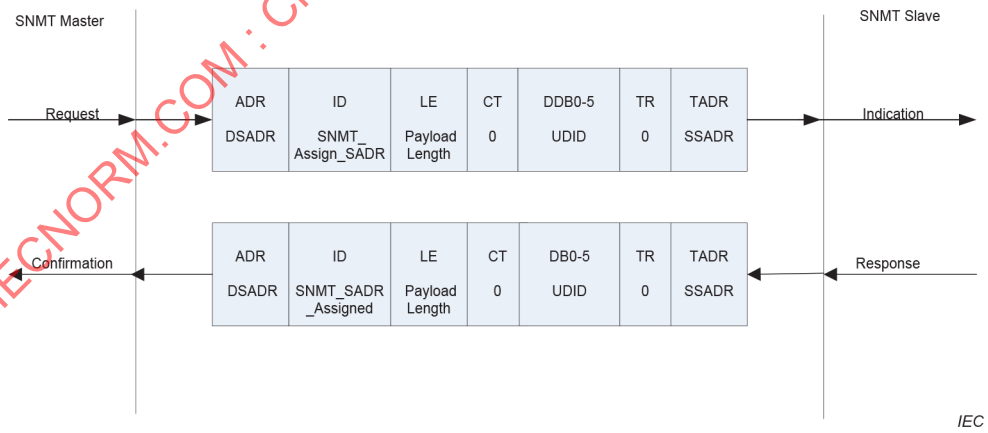


Figure 40 – SADR Assignment protocol

Table 41 specifies the fields of an SNMT_Assign_SADR telegram.

Table 41 – Fields of SNMT_Assign_SADR telegram

Field	Information	Content / Value
ADR	Address of the SNMT Slave to be assigned	DSADR
ID	SNMT_Assign_SADR	101010xxb
LE	Length of payload data	6
CT	Not used	0
DB0 – DB5	UDID of the SN	UDID
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 42 specifies the fields of an SNMT_SADR_ Assigned telegram.

Table 42 – Fields of SNMT_SADR_ Assigned telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT_SADR_ Assigned	101011xxb
LE	Length of payload data	6
CT	Not used	0
DB0 – DB5	UDID of the SN	UDID
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.4.3.3 Reset Node Guarding Time

The service Reset Node Guarding Time shall be used by the SN to trigger a node guarding on the SCM. The SCM shall guard all SNs whether the SADR or SDN in the telegram are matching the SADR and SDN of the SCM or not.

The telegram shall always be sent with SADR and SDN being 1.

NOTE This is due to the fact that the telegram is sent before an SADR or SDN is assigned to the node.

Figure 41 specifies the Reset Node Guarding Time protocol.

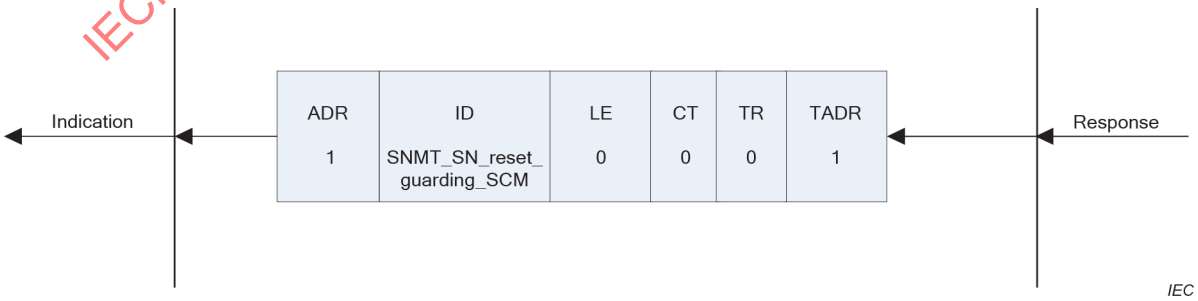


Figure 41 – Reset Node Guarding Time protocol

Table 43 specifies the fields of an SNMT_SN_reset_guarding_SCM telegram.

Table 43 – Fields of SNMT_SN_reset_guarding_SCM telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	1
ID	SNMT_SN_reset_guarding_SCM	$101111xx_b$
LE	Length of payload data	0
CT	Not used	0
TR	Not used	0
TADR	Address of SNMT Slave	1

7.4.3.4 SNMT Extended Services

SNMT Extended Services is specified by the ID field and the DB0 field of the FSPC 13/1 PDU (see Table 44 and Table 45).

Table 44 – SNMT request telegram types

Bit number of DB0								SNMT telegram type
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	SNMT_SN_set_to_PRE_OP
0	0	0	0	0	0	1	0	SNMT_SN_set_to_OP
0	0	0	0	0	1	0	0	SNMT_SCM_set_to_STOP
0	0	0	0	0	1	1	0	SNMT_SCM_set_to_OP
0	0	0	0	1	0	0	0	SNMT_SCM_guard_SN
0	0	0	0	1	0	1	0	SNMT_assign_additional_SADR
0	0	0	0	1	1	0	0	SNMT_SN_ACK
0	0	0	0	1	1	1	0	SNMT_assign_UDID_SCM
0	0	0	1	0	0	0	0	SNMT_assign_Init_CT

Table 45 – SNMT response telegram types

Bit number of DB0								SNMT telegram type
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	SNMT_SN_status_PRE_OP
0	0	0	0	0	0	1	1	SNMT_SN_status_OP
0	0	0	0	0	1	0	1	SNMT_assigned_additional_SADR
0	0	0	0	0	1	1	1	SNMT_SN_FAIL
0	0	0	0	1	0	0	1	SNMT_SN_busy
0	0	0	0	1	1	1	1	SNMT_assigned_UDID_SCM
0	0	0	1	0	0	0	1	SNMT_assigned_Init_CT

7.4.3.5 SN set to Pre-Operational

The SN set to Pre-Operational service shall be used to switch an SNMT Slave from the Operational to the Pre-Operational communication state.

Figure 42 specifies the SN set to Pre-Operational protocol.

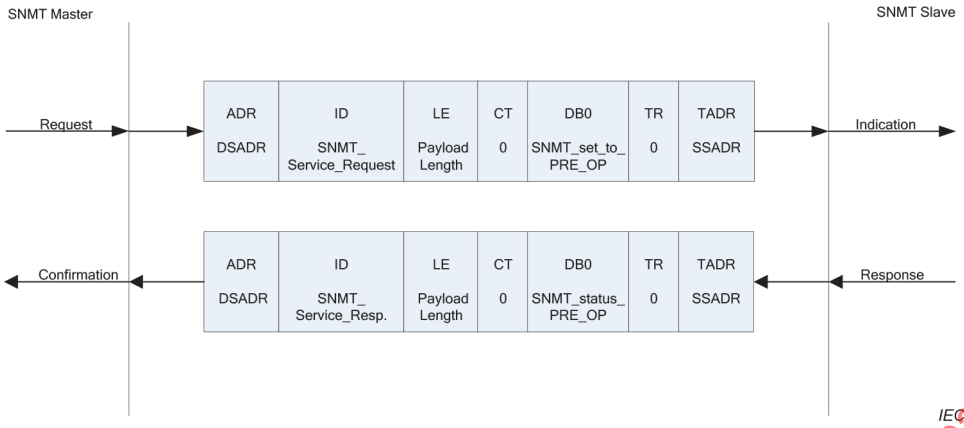


Figure 42 – SN set to Pre-Operational protocol

Table 46 specifies the fields of an SNMT_SN_set_to_PRE_OP telegram.

Table 46 – Fields of SNMT_SN_set_to_PRE_OP telegram

Field	Information	Content / Value
ADR	Address of the SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SN_set_to_PRE_OP	00000000b
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 47 specifies the fields of an SNMT_SN_status_PRE_OP telegram.

Table 47 – Fields of SNMT_SN_status_PRE_OP telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SN_status_PRE_OP	00000001b
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.4.3.6 SN set to Operational

The service SN set to Operational shall be used to switch an SNMT Slave from Pre-Operational to Operational communication state. The response shall either be the information that the SN has entered the Operational state, or error information which the SCM reports to its application.

Figure 43 specifies the SN set to Operational protocol.

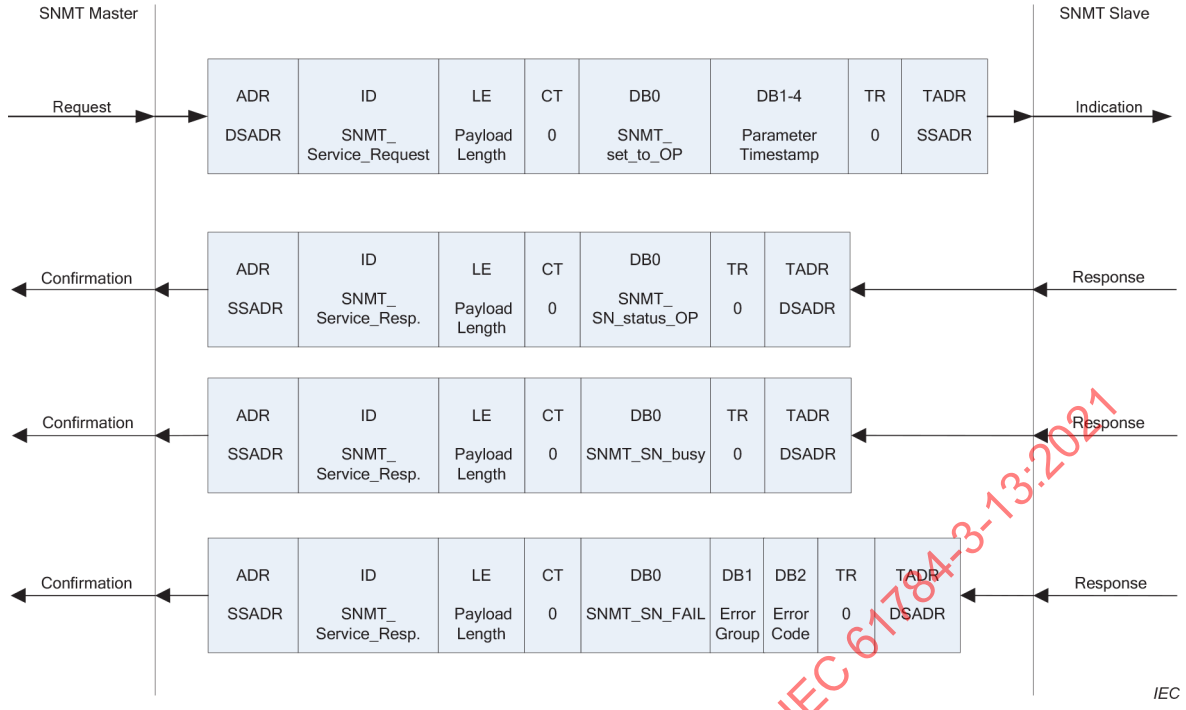


Figure 43 – SN set to Operational protocol

Table 48 specifies the fields of an SNMT_SN_set_to_OP telegram.

Table 48 – Fields of SNMT_SN_set_to_OP telegram

Field	Information	Content / Value
ADR	Address of SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	5
CT	Not used	0
DB0	SNMT_SN_set_to_OP	00000010b
DB1 – DB4	Parameter timestamp	See 7.5.4.9
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 49 specifies the fields of an SNMT_SN_status_OP telegram.

Table 49 – Fields of SNMT_SN_status_OP telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SN_status_OP	00000011b
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

Table 50 specifies the fields of an SNMT_SN_busy telegram.

Table 50 – Fields of SNMT_SN_busy telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SN_busy[a]	00001001b
TR	Not used	0
TADR	Address of SNMT Slave	SSADR
[a] SNMT_SN_busy shall be replied by the application (e.g. during calculation of the CRC or saving parameters).		

Table 51 specifies the fields of an SNMT_SN_FAIL telegram.

Table 51 – Fields of SNMT_SN_FAIL telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	3
CT	Not used	0
DB0	SNMT_SN_FAIL	00000111b
DB1	Error Group (see Table 52)	0 – 255
DB2	Error Code (see Table 53)	0 – 255
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

Table 52 specifies SNMT_SN_FAIL Error Group values.

Table 52 – SNMT_SN_FAIL Error Group values

Value	Error Group
0	Device
1	Application
2	Parameter
3	Vendor specific
4	FSCP 13/1 Stack
5	Additional parameter needed
6 – 255	Reserved for future use

Table 53 specifies SNMT_SN_FAIL Error Code values.

Table 53 – SNMT_SN_FAIL Error Code values

Value	Error Code
0	Default
1 – 255	Vendor specific

An SN may implement additional parameter sets. If an SN returns an SN_FAIL command with an ErrorGroup code of 5, the Error Code defines which set of additional parameters the node is requesting.

In case of an erroneous transmission of the parameter set, the node shall respond with a corresponding error code, indicating which parameter was not transmitted correctly.

Table 54 specifies SNMT_SN_FAIL Error Code values in case of Error Group Code is 5 (see Table 52).

Table 54 – SNMT_SN_FAIL Error Code values in case of Error Group Code 5

Value	Error Code
00h – 0Fh	Parameter sets 1-16
10h – EFh	Reserved
F0h – FFh	Failure during reception of parameter set 1-16

7.4.3.7 SNMT SN Acknowledge

The service SN Acknowledge shall be used by the SCM to acknowledge a reported error (see 7.4.3.6) to the SN.

Figure 44 specifies the SN Acknowledge protocol.

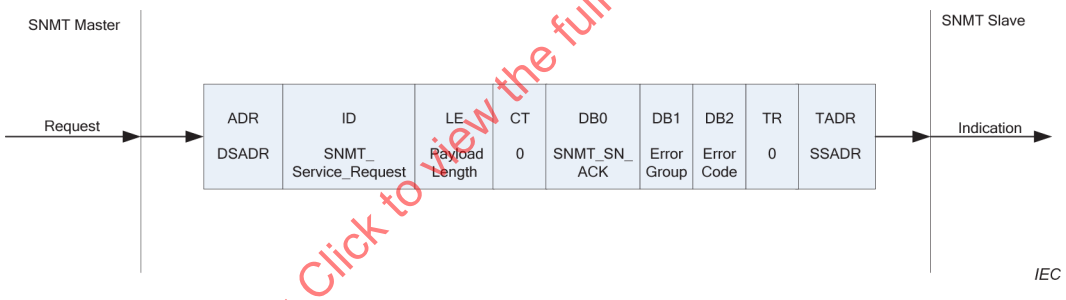


Figure 44 – SN Acknowledge protocol

Table 55 specifies the fields of an SNMT_SN_ACK telegram.

Table 55 – Fields of SNMT_SN_ACK telegram

Field	Information	Content / Value
ADR	Address of SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	3
CT	Not used	0
DB0	SNMT_SN_ACK	00001100b
DB1	Error Group (see Table 52)	0 – 255
DB2	Error Code (see Table 53)	0 – 255
TR	Not used	0
TADR	Address of SNMT Master	SSADR

7.4.3.8 SCM set to stop

The SCM set to stop service shall be used to switch an SCM off (from Operational to Stopped communication state) for system configuration by means of an external tool.

Figure 45 specifies the SCM set to stop protocol.

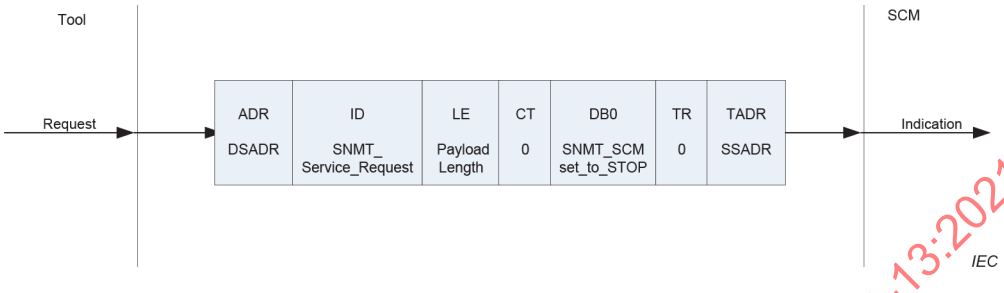


Figure 45 – SN set to stop protocol

Table 56 specifies the fields of an SNMT_SCM_set_to_STOP telegram.

Table 56 – Fields of SNMT_SCM_set_to_STOP telegram

Field	Information	Content / Value
ADR	Address of the SCM to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SCM_set_to_STOP	00000100b
TR	Not used	0
TADR	Address of the TOOL	SSADR

7.4.3.9 SCM set to Operational

The SCM set to Operational service shall be used to switch the SCM from Stopped to Operational communication state. This service may be applied after completion of the configuration by means of an external tool.

Figure 46 specifies the SCM set to Operational protocol.

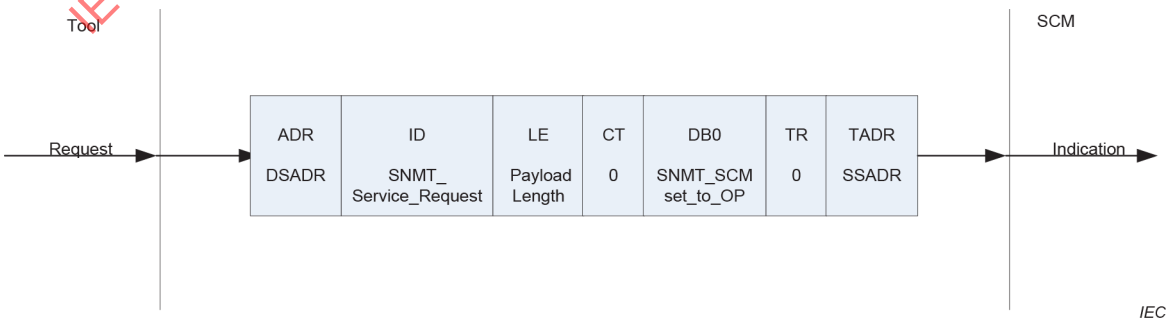


Figure 46 – SCM set to Operational protocol

Table 57 specifies the fields of an SNMT_SCM_set_to_OP telegram.

Table 57 – Fields of SNMT_SCM_set_to_OP telegram

Field	Information	Content / Value
ADR	Address of the SCM to be accessed	DSADR
ID	SNMT_Service_Request	101100xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SCM_set_to_OP	00000110b
TR	Not used	0
TADR	Address of the TOOL	SSADR

7.4.3.10 Node Guarding

The Node Guarding service shall be used to guard an SN in Operational communication state. The SN shall respond with its status. If an SN does not receive an SNMT_SCM_guard_SN telegram within its life time (see 7.5.4.6), it shall switch back to Pre-Operational. Signaling a lost request or response to the application shall be vendor specific.

Figure 47 specifies the Node Guarding protocol.

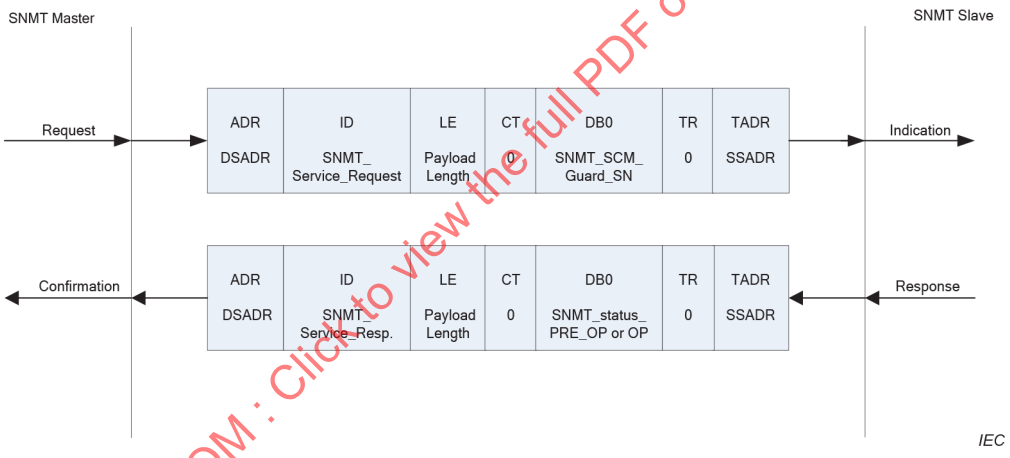


Figure 47 – Node Guarding protocol

Table 58 specifies the fields of an SNMT_SCM_guard_SN telegram.

Table 58 – Fields of SNMT_SCM_guard_SN telegram

Field	Information	Content / Value
ADR	Address of SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SCM_guard_SN	00001000b
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 59 specifies the fields of SNMT_SN_status_PRE_OP / SNMT_SN_status_OP telegrams.

Table 59 – Fields of SNMT_SN_status_OP/SNMT_SN_status_OP telegrams

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	1
CT	Not used	0
DB0	SNMT_SN_status_PRE_OP or SNMT_SN_status_OP	00000001b or 00000011b
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.4.3.11 Additional SADR Assignment

The Additional SADR Assignment service shall be used to assign an additional SADR to one SNMT Slave. This service shall only be used if an SN does support more than one TxSPDO (see 7.5.4.18).

If the SNMT Slave does not support more than one TxSPDO, the SNMT Slave shall reply with 0 as the SADR value (DB1, DB2 in the response telegram).

Figure 48 specifies the Additional SADR Assignment protocol.

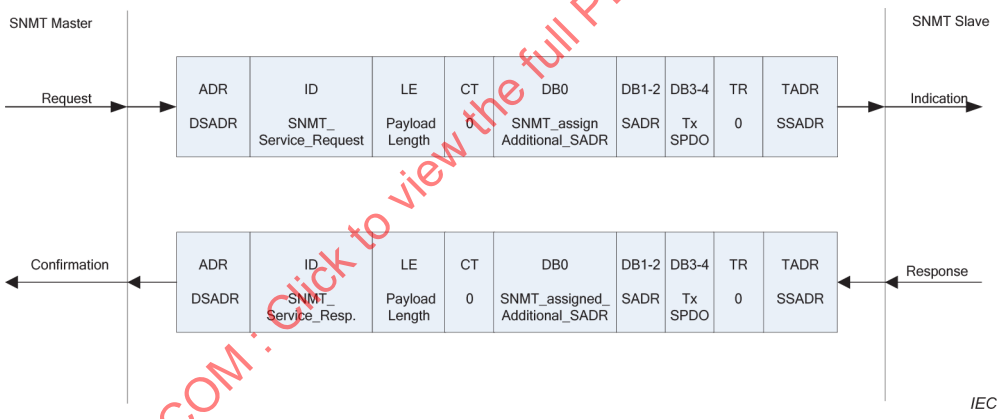


Figure 48 – Additional SADR Assignment protocol

Table 60 specifies the fields of an SNMT_assign_additional_SADR telegram.

Table 60 – Fields of SNMT_assign_additional_SADR telegram

Field	Information	Content / Value
ADR	Address of SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	5
CT	Not used	0
DB0	SNMT_assign_additional_SADR	00001010b
DB1,DB2	Additional SADR to be assigned	SADR
DB3,DB4	TxSPDO to which additional SADR is assigned	2 – 1 023
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 61 specifies the fields of an SNMT_assigned_additional_SADR telegram.

Table 61 – Fields of SNMT_assigned_additional_SADR telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	5
CT	Not used	0
DB0	SNMT_assigned_additional_SADR	00000101b
DB1, DB2	SADR that was assigned	SADR
DB3, DB4	TxSPDO to which SADR was assigned	2 – 1 023
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.4.3.12 UDID of SCM Assignment

The UDID of SCM Assignment service shall be used to assign the UDID of the SCM to the SNMT Slave.

NOTE The UDID of the SCM is coded in all SPDOs and SSDOs. Therefore, the SNMT slave needs the UDID of the SCM to decode those telegrams.

An error while executing the service shall be indicated with UDID value 00-00-00-00-00-00 in the response.

Figure 49 specifies the UDID of SCM Assignment protocol.

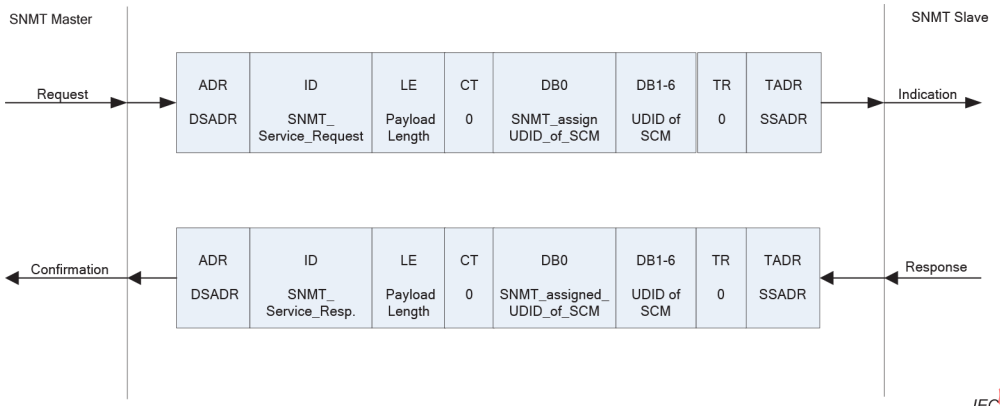


Figure 49 – UDID of SCM Assignment protocol

Table 62 specifies the fields of an SNMT_assign_UDID_of_SCM telegram.

Table 62 – Fields of SNMT_assign_UDID_of_SCM telegram

Field	Information	Content / Value
ADR	Address of SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	7
CT	Not used	0
DB0	SNMT_assign_UDID_of_SCM	00001110b
DB1 – DB6	UDID of the SCM	UDID
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 63 specifies the fields of an SNMT_assigned_UDID_of_SCM telegram.

Table 63 – Fields of SNMT_assigned_UDID_of_SCM telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	7
CT	Not used	0
DB0	SNMT_assigned_UDID_of_SCM	00001111b
DB1 – DB6	UDID of the SCM	UDID
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.4.3.13 Initialize CT on SN

In case the 40 bit SPDO counter (see 7.1.1.4) is used, the Init CT assignment service shall be used to seed the CT clock for a given SN with a randomized value from the SCM. This initial value shall be the new start value for the CT timer on the SN.

Otherwise the SNs already provide a CT value of their own or Init CT assignment service may be used too.

Figure 50 specifies the Init CT value assignment protocol.

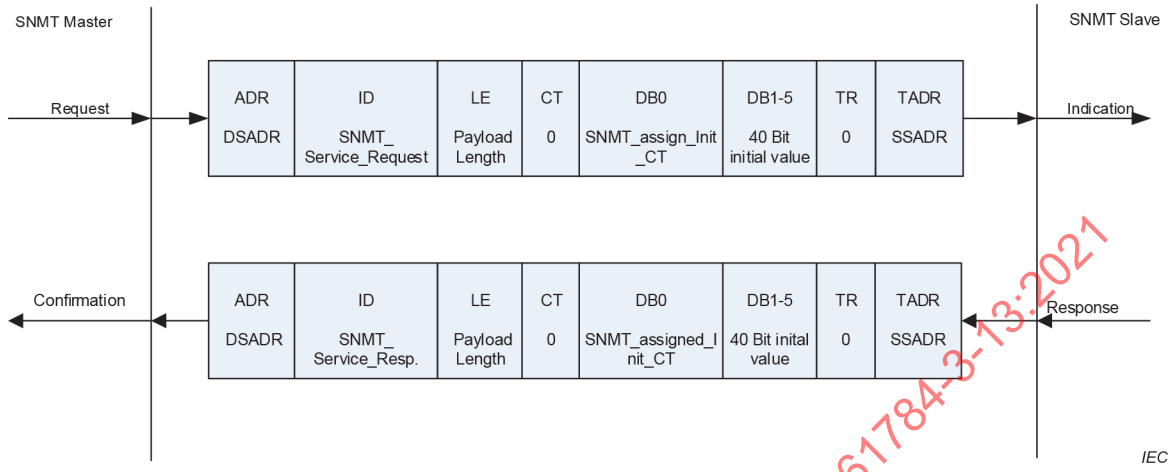


Figure 50 – Init CT value assignment protocol

Table 64 specifies the fields of an SNMT_assign_Init_CT telegram.

Table 64 – Fields of SNMT_assign_Init_CT telegram

Field	Information	Content / Value
ADR	Address of SNMT Slave to be accessed	DSADR
ID	SNMT Service Request	101100xxb
LE	Length of payload data	7
CT	Not used	0
DB0	SNMT_assign_Init CT	00010000b
DB1 – DB5	40 bit initial value	UDID
TR	Not used	0
TADR	Address of SNMT Master	SSADR

Table 65 specifies the fields of an SNMT_assigned_Init_CT telegram.

Table 65 – Fields of SNMT_assigned_Init_CT telegram

Field	Information	Content / Value
ADR	Address of SNMT Master	DSADR
ID	SNMT Service Response	101101xxb
LE	Length of payload data	7
CT	Not used	0
DB0	SNMT_assigned_Init CT	00010001b
DB1 – DB5	40 bit initial value	UDID
TR	Not used	0
TADR	Address of SNMT Slave	SSADR

7.5 Safety Object dictionary (SOD)

7.5.1 General

The following subclasses specify the Safety Object Dictionary structure and entries which shall be common for all devices.

7.5.2 Object dictionary entry definition

7.5.2.1 General

An Object Dictionary entry shall be defined by the following items:

- index,
- object,
- object type,
- name,
- data type,
- category,
- access,
- value range,
- default value,
- SPDO mapping.

7.5.2.2 Index

Index shall denote the objects position within the Object Dictionary. This acts as a kind of address to reference the desired data field. An Index shall be declared as hexadecimal value.

7.5.2.3 Object

An Object may be subdivided to sub indices. The sub index shall be used to reference data fields within a compound object such as an array or record. For sub index definition, see 7.5.2.14.

7.5.2.4 Object type

Object type shall denote what kind of object is at that particular index within the Object Dictionary. Object types shall be as specified in Table 66.

Table 66 – Object type definition

Object type	Comments	Code
NULL	A dictionary entry with no data fields	0
DOMAIN	Large variable amount of data e.g. executable program code	2
DEFTYPE	Denotes a basic data type definition (e.g. BOOLEAN, UNSIGNED16, REAL32)	5
DEFSTRUCT	Defines a record type	6
VAR	A single value (e.g. UNSIGNED8, BOOLENA, REAL32, INTEGER16, VISIBLE STRING)	7
ARRAY	A multiple data field object where each data field is a simple variable of the SAME basic data type e.g. array of UNSIGNED16 etc. Sub index 0 is of UNSIGNED8 and therefore not part of the ARRAY data	8
RECORD	A multiple data field object where the data fields may be any combination of simple variables. Sub index 0 is an UNSIGNED8 and therefore not part of the RECORD data	9

7.5.2.5 Name

Name shall provide a simple textual description of the function of that particular object. Name shall comply with IEC 61131-3. A name shall consist of:

- domain prefix, indicating the association of the object to a functional domain, up to 4 uppercase characters followed by underline,
- name (verbally) composed of words, each word shall be leaded by an uppercase character followed by lowercase characters or digits, no underlines or spaces,
- data type postfix, indicating the data type of the object (underline followed by up to 5 uppercase characters or digits).

Total length of name shall not exceed 32 characters.

7.5.2.6 Data type

Data type shall provide information about the data type of the object. It includes the following pre-defined basic data types: BOOLEAN, floating point number, UNSIGNED Integer, Signed Integer, visible/octet string and DOMAIN. It also includes the pre-defined compound data types and may also include types which are either manufacturer or device specific.

The following shall not be defined:

- records of records,
- arrays of records, or
- records with arrays as fields of that record.

In the case where an object is an array or a record, the sub index shall be used to reference one basic data type data field within the object.

Data type shall be a basic data type for ARRAY type objects and a compound data type for RECORD type objects.

7.5.2.7 Category

Category defines whether the object is Mandatory (M) or Optional (O). A mandatory object shall be implemented on a device. An optional object may be implemented on a device. The support of certain objects or features however may require the implementation of related objects. In this case, the relations are described in the detailed object specification.

Category may be Conditional (Cond) if the M/O category of an object depends on the implementation of another object.

Category shall refer to a compound data type object as a whole, but not to the particular sub index. A mandatory object may be composed of mandatory and optional sub indices. This means that the object shall be supported but some of its sub indices are optional. Defining optional objects with mandatory sub indices is also permitted. This means that these sub indices shall be supported if the object is implemented.

7.5.2.8 Access

Access defines the access rights for a particular object. The point of view is from the outside into the device.

Access rights are specified in Table 67.

Table 67 – Access attributes for data objects

Access right	Description
rw	read and write access
wo	write only access
ro	read only access
Const	read only access, value is constant

Optional objects shall be listed in the Object Dictionary with the attribute rw may be implemented as read only. Exceptions are defined in the detailed object specification.

7.5.2.9 Value range

This entry shall indicate the value range of the respective object. It may consist of one or more distinct values or ranges of values. If the item shown is a data type identifier, the complete value range of the mentioned data type shall be allowed.

7.5.2.10 Default value

This entry shall indicate the initialization value that shall be assigned by the profile implementation. For mapable objects, this value shall also represent the safe state value.

7.5.2.11 SPDO mapping

This entry indicates whether an entry may be mapped to an SPDO message.

SPDO mapping values shall be as specified in Table 68.

Table 68 – SPDO mapping attributes for data objects

Value	Description
Opt	Object may be mapped into an SPDO
No	Object shall not be mapped into an SPDO

7.5.2.12 Example basic data type object definition

A complete basic data type object definition (Object Type = VAR or DOMAIN) example is shown in Table 69.

Table 69 – Basic data type object definition example

Item	Value
Index	1234h
Name	SSDO_VarDummyParameter_S16
Object type	VAR
Data type	INTEGER16
Value range	-15 000 – 10 000
Default value	0
Category	M
Access	rw
SPDO mapping	Opt

7.5.2.13 Example compound data type object definition

Compound data type object definitions (Object Type = ARRAY or RECORD) are of reduced form, because Access, Value Range, Default Value, and SPDO mapping shall be defined individually for the sub indices.

Table 70 gives a compound data type object definition example.

Table 70 – Compound data type object definition example

Item	Value
Index	2345h
Name	SSDO_ArrayDummyParameter_AU16
Object type	ARRAY
Data type	UNSIGNED16
Category	M

7.5.2.14 Sub index definition

7.5.2.14.1 General

Compound object types (ARRAY, RECORD objects) shall be composed of up to 256 data items. Each data item shall be addressed by an UNSIGNED8 type sub index.

Sub indices shall be interpreted as specified in Table 71.

Table 71 – Sub index interpretation

Sub index	Name
00h	NumberOfEntries
01h – FEh	Object Specific Data
FEh	ChecksumOfObject
FFh	StructureOfObject

7.5.2.14.2 Sub index 00h – NumberOfEntries

NumberOfEntries shall describe the highest available sub index, not considering FFh. This entry shall be encoded as UNSIGNED8, regardless of the type of the object. If the object exists, NumberOfEntries shall be mandatory. Data Type and Category shall not be denoted in NumberOfEntries descriptions.

In this document, NumberOfEntries is specified in the form as shown in Table 72. Elements depicted by <...> shall be set according the specific context.

Table 72 – NumberOfEntries sub index specification

Item	Value
Sub index	00h
Name	NumberOfEntries
Value range	<valid values, for example 1 – 15>
Default value	<default value, for example 15>
Access	<Access, for example rw>
SPDO mapping	No

For ARRAYs, NumberOfEntries shall express the number of occupied items in a list (Access shall be rw). For RECORDs, NumberOfEntries shall be constant (Access shall be ro or Const).

7.5.2.14.3 Sub index 01h – FEh – Object specific data

Sub indices between 01h and FEh shall hold the object's payload data. The highest accessible index shall be given by NumberOfEntries.

In this document, Object Specific Data of RECORD type objects are specified in the form as shown in Table 73. Elements depicted by <…> shall be set according the specific context.

Table 73 – RECORD type object sub index specification

Item	Value
Sub index	<sub index in hexadecimal notation, for example 01h>
Name	<name>
Data type	<data type, for example UNSIGNED8>
Value range	<valid values, for example 1 – 255>
Default value	<default value, for example 1>
Category	<mandatory or optional, for example M>
Access	<Access, for example rw>
SPDO mapping	<mapping possible, for example Opt>

Name shall be composed of a descriptive text followed by a data type postfix. Subclause 7.5.2.5 shall apply, with the single exception that the domain prefix is omitted.

If a name entry is defined in a data type definition, the sub index name shall be equal to it.

Category shall refer to the respective sub index only. Mandatory (M) shall denote that the sub index shall be implemented if the object is implemented. A mandatory sub index shall not force the parent object to be mandatory.

In this document, Object Specific Data of ARRAY type objects are specified in the form as shown in Table 74. Elements depicted by <…> shall be set according the specific context.

Table 74 – ARRAY type object sub index specification

Item	Value
Sub index	<sub index in hexadecimal notation, for example 01h>
Name	<name>
Value range	<data type, for example UNSIGNED16>
Default value	<default value, for example 0>
Category	<mandatory or optional, for example M>
Access	<Access, for example rw>
SPDO mapping	<mapping possible, for example Opt>

Name shall be a descriptive text. Subclause 7.5.2.5 shall apply, with the exceptions that, the domain prefix and data type postfix are omitted. If a name entry is defined in a data type definition, the sub index name shall be equal to it.

Category shall refer to the respective sub index only. Mandatory (M) shall denote that the sub index shall be implemented if the object is implemented. A mandatory sub index shall not force the parent object to be mandatory.

NOTE Despite the functional equivalence of all sub indices in an array, more than one sub index entry can be defined to a particular object to show differences of the Category, Access and / or SPDO Mapping options of the sub indices.

7.5.2.14.4 Sub index FFh – StructureOfObject

Sub index FFh may denote the structure of the object by providing the data type and the object type. Regardless of the type of the object, it is encoded as UNSIGNED32.

Encoding of StructureOfObject is specified by Table 75.

Table 75 – StructureOfObject encoding

	MSB [a]			LSB [b]
Bit-No.	31 – 24	23 – 16	15 – 8	7 – 0
Value	00h	Data Type (refer to Table 76)		Object Type (refer to Table 66)
		(High octet)	(Low octet)	
[a] Most significant octet.				
[b] Least significant octet.				

NOTE It is optional to support sub index FFh. If it is supported throughout the Object Dictionary and the structure of the compound data types is provided as well, it is possible to upload the entire structure of the Object Dictionary.

If StructureOfEntry is not supported, sub index FFh shall be reserved.

StructureOfObject is not shown in object definitions in this specification.

7.5.3 Data type entry specification

7.5.3.1 General

The basic data types shall be placed in the Object Dictionary for definition purposes only. However, indices in the range 0001h – 0007h may be mapped in order to define the appropriate space of the RxSPDO (Dummy Mapping) as not being used by the device (do not care). The indices 0009h to 000Bh and 000Fh shall not be mapped to an SPDO.

Index range 0001h to 04FFh shall be used for data type specifying object dictionary entries.

Table 76 specifies the data types used in the SOD.

Table 76 – Object dictionary data types

Index	Object	Name
0001h	DEFTYPE	BOOLEAN
0002h	DEFTYPE	INTEGER8
0003h	DEFTYPE	INTEGER16
0004h	DEFTYPE	INTEGER32
0005h	DEFTYPE	UNSIGNED8
0006h	DEFTYPE	UNSIGNED16
0007h	DEFTYPE	UNSIGNED32
0008h	DEFTYPE	REAL32
0009h	DEFTYPE	VISIBLE_STRING
000Ah	DEFTYPE	OCTET_STRING
000Bh	DEFTYPE	UNICODE_STRING
000Ch	reserved	
000Dh	reserved	
000Eh	reserved	
000Fh	DEFTYPE	DOMAIN
0010h	DEFTYPE	INTEGER24
0011h	DEFTYPE	REAL64
0012h	DEFTYPE	INTEGER40
0013h	DEFTYPE	INTEGER48
0014h	DEFTYPE	INTEGER56
0015h	DEFTYPE	INTEGER64
0016h	DEFTYPE	UNSIGNED24
0017h	reserved	
0018h	DEFTYPE	UNSIGNED40
0019h	DEFTYPE	UNSIGNED48
001Ah	DEFTYPE	UNSIGNED56
001Bh	DEFTYPE	UNSIGNED64
001Ch – 001Fh	reserved	
0020h	DEFSTRUCT	SSDO_LifeGuardRecord_TYPE
0021h	DEFSTRUCT	SINF_DevVendorInfoRecord_TYPE
0022h	DEFSTRUCT	SCOM_CommonCommParamRecord_TYPE
0023h	DEFSTRUCT	SSDO_CommParamRecord_TYPE
0024h	DEFSTRUCT	SNMT_CommParamRecord_TYPE
0025h	DEFSTRUCT	SPDO_RxCommParamRecord_TYPE
0026h	DEFSTRUCT	SPDO_TxCommParamRecord_TYPE
0027h	DEFSTRUCT	SSCM_SADRDVIListRecord_TYPE
0028h	DEFSTRUCT	SSCM_UDIDSADRListRecord_TYPE
0029h	DEFSTRUCT	SSCM_SpecificParametersRecord_TYPE
002Ah – 003Fh	reserved	

7.5.3.2 Basic data types

For basic data types (Object Type = DEFTYPE) see 5.5.2.

A device may optionally provide the length of the basic data types encoded as UNSIGNED32 for read access of the index that refers to the data type.

EXAMPLE Index 0007h (UNSIGNED32) contains the value 20h=32d because the data type UNSIGNED32 is encoded using a bit sequence of 32 bit. If the length is variable (e.g. 000Fh = Domain), the entry contains 0h.

7.5.3.3 Compound data types

For the supported compound data types, a device may optionally provide the structure of that data type for read access at the corresponding data type index. Sub index 0 then provides the number of entries at this index not counting sub indices 0 and 255; the following sub indices contain the data type according to Table 76 encoded as UNSIGNED16.

Table 77 and Table 78 give an example of a compound data type definition describing the structure of the identity object at index 0021h.

Table 77 – 0021h Compound data type description example

Item	Value
Index	0021h
Name	SINF_DevVendorInfoRecord_TYPE
Object type	DEFSTRUCT

Table 78 – 0021h Compound sub index descriptions example

Sub index	Component name	Value	Data Type
00h	NumberOfEntries	06h	UNSIGNED16
01h	VendorId_U32	07h	UNSIGNED32
02h	ProductCode_U32	07h	UNSIGNED32
03h	RevisionNo_U32	07h	UNSIGNED32
04h	SerialNo_U32	07h	UNSIGNED32
05h	FirmWareChecksum_U32	07h	UNSIGNED32
06h	ParameterChecksum_U16	06h	UNSIGNED16
07h	ParameterTimestamp_U32	07h	UNSIGNED32

Standard (simple) and compound manufacturer specific data types shall be distinguished by attempting to read sub index 1h: For a compound data type, the device shall return a value and sub index 0h contains the number of sub indices that follow; For a standard data type, the device shall abort the SOD Access because no sub index 1h is available.

7.5.4 Object description

7.5.4.1 General

This and the following subclasses describe the standard object dictionary entries for an SN and SCM. The implementation of these objects may differ from the description within the specification.

EXAMPLE The Main SADR might be a read only parameter in case it is to be set using a dip switch.

Table 79 to Table 89 specify the FSCP 13/1 standard objects.

Table 79 – Standard objects

Index	Object type	Name	Data type	Acc	M/O
1001h	VAR	Error register	UNSIGNED8	ro	M
1002h	VAR	Manufacturer status register	UNSIGNED32	ro	O
1003h	ARRAY	Pre-defined error field	UNSIGNED32	rw	O
:::	:::	:::	:::	:::	:::
100Ch	RECORD	Life guarding	SSDO_LifeGuardRecord_TYPE (20h)	-	M
100Dh	VAR	Refresh Time for Reset Guarding	UNSIGNED32	rw	M
100Eh	VAR	Number of Retries for Reset Guarding	UNSIGNED8	rw	O
1018h	RECORD	Device vendor information	SINF_DevVendorInfoRecord_TYPE (21h)	-	M
1019h	VAR	Unique device id	OCTET_STRING	ro	M
101Ah	DOMAIN	Parameter download	DOMAIN	rw	O
101Bh	RECORD	SCM specific parameters	SSCM_SpecificParametersRecord_TYPE (29h)	-	O

Table 80 – Common communication objects

Index	Object type	Name	Data type	Acc	M/O
1200h	RECORD	common communication parameters	SCOM_CommonCommParamRecord_TYPE (22h)	-	M
1201h	RECORD	SSDO communication parameters	SSDO_CommParamRecord_TYPE (23h)	-	C[a]
1202h	RECORD	SNMT communication parameters	SNMT_CommParamRecord_TYPE (24h)	-	C
[a] The objects shall only be available if the node can be an SSDO client.					

Table 81 – Receive SPDO communication objects

Index	Object type	Name	Data type	Acc	M/O
1400h	RECORD	1st RxSPDO communication parameters	SPDO_RxCommParamRecord_TYPE (25h)	rw	M[a]
1401h	RECORD	2nd RxSPDO communication parameters	SPDO_RxCommParamRecord_TYPE (25h)	rw	M[a]
:::	:::	:::	:::	:::	:::
17FEh	RECORD	1 023rd RxSPDO communication parameters	SPDO_RxCommParamRecord_TYPE (25h)	rw	M[a]
[a] If a device supports SPDOs, the corresponding SPDO communication parameter and SPDO mapping entries in the Object dictionary are mandatory.					

Table 82 – Receive SPDO mapping objects

Index	Object type	Name	Data type	Acc	M/O
1800h	ARRAY	1st RxSPDO mapping parameters	UNSIGNED32	rw	M[a]
1801h	ARRAY	2nd RxSPDO mapping parameters	UNSIGNED32	rw	M[a]
:::	:::	:::	:::	:::	:::
1BFEh	ARRAY	1 023rd RxSPDO mapping parameters	UNSIGNED32	rw	M[a]
[a] If a device supports SPDOs, the corresponding SPDO communication parameter and SPDO mapping entries in the Object dictionary are mandatory.					

Table 83 – Transmit SPDO communication objects

Index	Object type	Name	Data type	Acc	M/O
1C00h	RECORD	1st TxSPDO communication parameters	SPDO_TxCommParamRecord_TYPE (26h)	rw	M[a]
1C01h	RECORD	2nd TxSPDO communication parameters	SPDO_TxCommParamRecord_TYPE (26h)	rw	M[a]
:::	:::	:::	:::	:::	:::
1FFEh	RECORD	1 023rd TxSPDO communication parameters	SPDO_TxCommParamRecord_TYPE (26h)	rw	M[a]
[a] If a device supports SPDOs, the corresponding SPDO communication parameter and SPDO mapping entries in the Object dictionary are mandatory.					

Table 84 – User parameter (writeable at any time)

Index	Object type	Name	Data type	Acc	M/O
2800h	ARRAY	1st User parameter	UNSIGNED32	rw	O
:::	:::	:::	:::	:::	O
2FFFh	ARRAY	2 048th User parameter	UNSIGNED32	rw	O

Table 85 – Transmit SPDO mapping objects

Index	Object type	Name	Data type	Acc	M/O
C000h	ARRAY	1st TxSPDO mapping parameters	UNSIGNED32	rw	M[a]
C001h	ARRAY	2nd TxSPDO mapping parameters	UNSIGNED32	rw	M[a]
:::	:::	:::	:::	:::	:::
C3FEh	ARRAY	1 023rd TxSPDO mapping parameters	UNSIGNED32	rw	M[a]
[a] If a device supports SPDOs, the corresponding SPDO communication parameter and SPDO mapping entries in the Object dictionary are mandatory.					

Table 86 – SADR DVI list

Index	Object type	Name	Data type	Acc	M/O
C400h	RECORD	1st SADR-DVI List	SSCM_SADRDVIListRecord_TYPE (27h)	rw	C[a]
C401h	RECORD	2nd SADR-DVI List	SSCM_SADRDVIListRecord_TYPE (27h)	rw	C[a]
:::	:::	:::	:::	:::	:::
C7FEh	RECORD	1 023rd SADR-DVI List	SSCM_SADRDVIListRecord_TYPE (27h)	rw	C[a]
[a] Only needs to be implemented if the device has SCM functionality.					

Table 87 – Additional SADR list

Index	Object type	Name	Data type	Acc	M/O
C801h	ARRAY	1st SADR	UNSIGNED16	rw	C[a]
C802h	ARRAY	2nd SADR	UNSIGNED16	rw	C[a]
:::	:::	:::	:::	:::	:::
CBFFh	ARRAY	1 023rd SADR	UNSIGNED16	rw	C[a]
[a] Only needs to be implemented if the device has SCM functionality.					

Table 88 – SADR UDID list

Index	Object type	Name	Data type	Acc	M/O
CC01h	RECORD	1st SADR-UDID List	SSCM_UDIDSADRListRecord_TYPE (28h)	rw	C[a]
CC02h	RECORD	2nd SADR-UDID List	SSCM_UDIDSADRListRecord_TYPE (28h)	rw	C[a]
:::	:::	:::	:::	:::	:::
CFFFh	RECORD	1 023rd SADR-UDID List	SSCM_UDIDSADRListRecord_TYPE (28h)	rw	C[a]
[a] Only needs to be implemented if the device has SCM functionality.					

Table 89 – Additional parameter list

Index	Object type	Name	Data type	Acc	M/O
E400h	RECORD	1st Additional Parameter List	SSCM_ADDPARAListRecord_TYPE (2Ah)	rw	C[a]
:::	:::	:::	:::	:::	:::
E7FEh	RECORD	1 023rd Additional Parameter List	SSCM_ADDPARAListRecord_TYPE (2Ah)	rw	C[a]
[a] Only needs to be implemented if the device has SCM functionality.					

7.5.4.2 Object 1001h: Error Register

This object shall be an error register for the device. The device may map internal errors in this octet. If the value of the object is zero, the device shall be ok, all other values shall signal that the device is in an error state.

Object 1001h Error Register is specified in Table 90.

Table 90 – Object 1001h Error Register

Item	Value
Index	1001h
Name	SERR_GeneralError_U8
Object type	VAR
Data type	UNSIGNED8
Value range	UNSIGNED8
Default value	0
Category	M
Access	ro
SPDO mapping	Yes
Relevant for SOD checksum	No

The value interpretation for Object 1001h Error Register is specified in Table 91.

Table 91 – Object 1001h Error Register value interpretation

Bit	M/O	Description
0	M	generic error
1	O	current
2	O	voltage
3	O	temperature
4	O	communication error (overrun, error state)
5	O	device profile specific
6	O	Reserved (always 0)
7	O	manufacturer specific

7.5.4.3 Object 1002h: Manufacturer status register

This object shall be a status register for manufacturer purposes. In this document, only the size and location of this object shall be defined.

Object 1002h Manufacturer status register is specified in Table 92.

Table 92 – Object 1002h Manufacturer status register

Item	Value
Index	1002h
Name	SERR_ManufacturerStatus_U32
Object type	VAR
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	O
Access	ro
SPDO mapping	Yes
Relevant for SOD checksum	No

7.5.4.4 Object 1003h: Pre defined error field

The object at index 1003h shall store the errors that have occurred on the device. Every new error shall be stored at sub index 01h, older errors shall be moved to the next higher sub index. In doing so, it provides an error history. Sub index 0 shall provide the number of errors in the list. Setting sub index 0 to zero shall be used for deleting the list.

Object 1003h Pre defined error field is specified in Table 93 to Table 96.

Table 93 – Object 1003h Pre defined error field

Item	Value
Index	1003h
Name	SERR_ErrorHistory_AU32
Object type	ARRAY
Data type	UNSIGNED32
Category	O
Relevant for SOD checksum	No

Table 94 – Object 1003h sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	1 – 254
Default value	1
Access	rw
SPDO mapping	No
[a] Number of errors.	

Table 95 – Object 1003h sub index 01h

Item	Value
Sub index	01h
Name	Error_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Error within the history list.	

Table 96 – Object 1003h sub index 02h to FEh

Item	Value
Sub index	02h – FEh
Name	Error_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	O
Access	ro
SPDO mapping	No
[a] Error within the history list.	

7.5.4.5 Object 1004h: Error statistics

Object 1004h specifies error statistic counters in Table 97 to Table 112.

Table 97 – Object 1003h Error statistics

Item	Value
Index	1004h
Name	SERR_ErrorStatistics_AU32
Object type	ARRAY
Data type	UNSIGNED32
Category	O
Relevant for SOD checksum	No

Table 98 – Object 1004h sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	0Eh
Default value	0Eh
Access	rw
SPDO mapping	No
[a] Number of error statistic entries	

Table 99 – Object 1004h sub index 01h

Item	Value
Sub index	01h
Name	SFS_Length [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Defined payload length for the PDU differs from the received data.	

Table 100 – Object 1004h sub index 02h

Item	Value
Sub index	02h
Name	SFS_too_long [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Safety PDU is too long.	

Table 101 – Object 1004h sub index 03h

Item	Value
Sub index	03h
Name	SFS_PDU_ID [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] PDU IDs differ between PDU part one and two.	

Table 102 – Object 1004h sub index 04h

Item	Value
Sub index	04h
Name	SFS_SADR_invalid [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Invalid SADR	

Table 103 – Object 1004h sub index 05h

Item	Value
Sub index	05h
Name	SFS_SDN_invalid [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Invalid SDN	

Table 104 – Object 1004h sub index 06h

Item	Value
Sub index	06h
Name	SFS_TADR_invalid [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Invalid TADR	

Table 105 – Object 1004h sub index 07h

Item	Value
Sub index	07h
Name	SFS_CRC1_invalid [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] CRC of PDU part one is invalid.	

Table 106 – Object 1004h sub index 08h

Item	Value
Sub index	08h
Name	SFS_CRC2_invalid [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] CRC of PDU part two is invalid.	

Table 107 – Object 1004h sub index 09h

Item	Value
Sub index	09h
Name	SFS_Data [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Payload differs between PDU part one and two.	

Table 108 – Object 1004h sub index 0Ah

Item	Value
Sub index	0Ah
Name	CYC_Reject [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Cyclic telegrams rejected because not applicable for SN.	

Table 109 – Object 1004h sub index 0Bh

Item	Value
Sub index	0Bh
Name	CYC_Error [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Generic cyclic telegram errors, telegrams intended for SN	

Table 110 – Object 1004h sub index 0Ch

Item	Value
Sub index	0Ch
Name	ACYC_Reject [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] SNMT/SSDO telegrams rejected because not applicable for SN	

Table 111 – Object 1004h sub index 0Dh

Item	Value
Sub index	0Dh
Name	ACYC_Retry [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Number of SNMT/SSDO retries	

Table 112 – Object 1004h sub index 0Eh

Item	Value
Sub index	0Eh
Name	COMMON_Counter [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Common error counter	

7.5.4.6 Object 100Ch: Life Guarding

This object defines the Life Guarding protocol. The guard time shall be the time interval in which the SN should receive a life time signal from the SCM. The life time factor multiplied by the guard time results in the life time of the node. The time interval for sending the life time signal to the SNs shall be calculated by dividing the GuardTime_U32 by the number of nodes. The guard time interval for an SN within the SCM is started after receiving the last guarding response from the SN.

Object 100Ch Life Guarding is specified in Table 113 to Table 116.

Table 113 – Object 100Ch Life Guarding

Item	Value
Index	100Ch
Name	SSDO_LifeGuard_REC
Object type	RECORD
Data type	SSDO_LifeGuardRecord_TYPE
Category	M
Relevant for SOD checksum	Yes

Table 114 – Object 100Ch sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	2
Default value	2
Access	ro
SPDO mapping	No
[a] Number of Entries within this index.	

Table 115 – Object 100Ch sub index 01h

Item	Value
Sub index	01h
Name	GuardTime_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	10 000 000
Category	M
Access	rw
SPDO mapping	No
[a] Guard time for the lifetime protocol. The time base of this parameter is defined by CommonCommParam_REC. ConsecutiveTimeBase_I8.The values' range may be limited to a smaller value than 232. This limitation is vendor specific and/or due to implementation reasons.	

Table 116 – Object 100Ch sub index 02h

Item	Value
Sub index	02h
Name	LifeTimeFactor_U8 [a]
Data type	UNSIGNED8
Value range	1 – 255
Default value	2
Category	M
Access	rw
SPDO mapping	No
[a] Number of times the GuardTime_U32 may elapse before the node switches back from Operational to Pre-Operational.	

7.5.4.7 Object 100Dh: Refresh Interval of Reset Guarding

This object contains the refresh interval of the Reset Guarding (see 7.7.6.3). The refresh interval shall be the time delay between each individual SNMT_SN_reset_guarding_SCM signal sent by the SN to it's accompanying SCM during operational and pre-operational phases. The number of attempts during boot-up may be defined by 100Eh (see 7.5.4.8).

Object 100Dh Refresh Interval of Reset Guarding is specified in Table 117.

Table 117 – Object 100Dh Refresh Interval of Reset Guarding

Item	Value
Index	100Dh
Name	SNMT_ResetGuarding_U32 [a]
Object type	VAR
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	100 000
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	No
[a] The sub index provides the refresh time for the SNMT_SN_reset_guarding_SCM service. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

7.5.4.8 Object 100Eh: Number of Retries for Reset Guarding

This object contains the number of retries a SN performs to contact its accompanying SCM using Reset Guarding (see 7.7.6.3) during Pre-Operational.

NOTE The time interval between each message is defined by object 100Dh (see 7.5.4.7).

Object 100Eh is optional and if omitted an infinite number of retries shall be assumed.

Object 100Eh Number of Retries for Reset Guarding is specified in Table 118.

Table 118 – Object 100Eh Number of Retries for Reset Guarding

Item	Value
Index	100Eh
Name	SNMT_NoResetGuarding_U8 [a]
Object type	VAR
Data type	UNSIGNED8
Value range	UNSIGNED8
Default value	Manufacturer specific
Category	O
Access	rw
SPDO mapping	No
Relevant for SOD checksum	No
[a] The sub index provides maximum number of Reset Guarding telegrams sent in Pre-Operational (see 7.7.6.3). If the value is set to 255 the Reset Guarding telegram shall be repeated infinitely. If the value is set to 0 – 254 the Reset Guarding telegrams shall not be sent anymore after the set number of repetitions.	

7.5.4.9 Object 1018h: Device Vendor Information

7.5.4.9.1 General

This object shall contain the vendor specific device information. It shall also define compatible devices. Any device with an identical Vendor-ID, Product code, Major Revision Number and a Minor Revision Number greater or equal to another device shall be compatible. The manufacturer of a device shall verify and ensure the compatibility for minor revision changes on a safety related device.

Object 1018h Device Vendor Information is specified in Table 119 to Table 129.

Table 119 – Object 1018h Device Vendor Information

Item	Value
Index	1018h
Name	SINF_DevVendorInfo_REC
Object type	RECORD
Data type	SINF_DevVendorInfoRecord_TYPE
Category	M
Relevant for SOD checksum	No

Table 120 – Object 1018h sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	9
Default value	9
Access	ro
SPDO mapping	No
[a] Number of Entries within this index.	

Table 121 – Object 1018h sub index 01h

Item	Value
Sub index	01h
Name	VendorID_32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Vendor ID according EPSG Vendor List.	

Table 122 – Object 1018h sub index 02h

Item	Value
Sub index	02h
Name	ProductCode_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] The manufacturer-specific product code identifies a specific device version.	

Table 123 – Object 1018h sub index 03h

Item	Value
Sub index	03h
Name	RevisionNumber_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] See 7.5.4.9.	

Table 124 – Object 1018h sub index 04h

Item	Value
Sub index	04h
Name	SerialNumber_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Serial number of the device.	

Table 125 – Object 1018h sub index 05h

Item	Value
Sub index	05h
Name	FirmWareChecksum_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Checksum used to test the integrity of the firmware which is stored on the device. Calculation is vendor specific.	

Table 126 – Object 1018h sub index 06h

Item	Value
Sub index	06h
Name	ParameterChecksum_DOM [a]
Data type	DOMAIN
Value range	DOMAIN
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] This object contains the SOD parameter checksum container that shall be calculated as described in 7.5.4.9.3.	

Table 127 – Object 1018h sub index 07h

Item	Value
Sub index	07h
Name	ParameterTimestamp_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] The Parameter Timestamp shall be the same as in ParameterChecksum_DOM. The parameter timestamp is created by the configuration tool and is verified in each device at start-up by the SCM (see 7.7.12.2). The timestamp format ist seconds since 1.1.1970 (UNIX timestamp).	

Table 128 – Object 1018h sub index 08h

Item	Value
Sub index	08h
Name	Stack_ChkSum_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Checksum, of FSCP 13/1 implementation used for creating the device firmware.	

Table 129 – Object 1018h sub index 09h

Item	Value
Sub index	09h
Name	Stack_Version_U32 [a]
Data type	UNSIGNED16
Value range	UNSIGNED16
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] See 7.5.4.9.4.	

7.5.4.9.2 Manufacturer Revision Number

The manufacturer-specific Revision Number shall consist of a major revision number and a minor revision number. The major revision number shall identify specific device behaviour. If the device functionality is expanded, the major revision shall be incremented. The minor revision number shall identify different versions with the same device behaviour.

Table 130 specifies the encoding of RevisionNumber_U32.

Table 130 – Structure of Revision Number

31	16	15	0
Major revision number		Minor revision number	
MSB			LSB

7.5.4.9.3 Parameter checksum

The parameter checksum object is a domain which shall contain the timestamp and n 32 bit CRCs as specified in Table 131. The timestamp shall be shown also in object 1018h sub index 7h. Each CRC shall be used to verify the correctness of up to 4 k bit (4096 bit / 512 octet) of SOD variables. The polynom that shall be used for the 32 bit CRCs in the parameter checksum is specified in Table 132.

Table 131 – Structure of parameter checksum domain

Timestamp
CRC32
CRC32
…

Table 132 – CRC polynom for parameter checksum

Polynom	Polynom	Polynom according [41]
$x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{23}+x^{22}+x^{20}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+x^{0}$	$11EDC6F41_{hex}$	0x8F6E37A0

When the SN is switched to operational, the whole SOD shall be checked against the CRCs contained in this object. Switching to operational shall only be allowed, if the timestamp of the SNMT command and the CRCs are correct.

For calculating the CRC over the SOD only the actual data of relevant objects shall be used. Objects marked as not relevant for the CRC calculation shall not be considered for the calculation. Due to the fact that a maximum of 4 k bit of data is allowed to be used for each CRC, more than one CRC may be necessary for securing the SOD of a large device.

It is not scope of this document to describe how the parameters and the checksum container are written to the device. It is only necessary to ensure that the entire SN parameters have been written to the SN. Due to the fact that the timestamp and the CRCs are coupled together in the object 1018h sub index 6h, the SCM only has to verify the timestamp, to ensure that the correct parameter/CRC set is on the SN.

7.5.4.9.4 Stack Version

The Stack Version contains the version for the FSCP 13/1 stack used to create the device firmware and shall consist of a major version number and a minor version number.

Table 133 specifies the encoding of Stack_Version_U32.

Table 133 – Structure of Stack Version

16	0
Major version number	Minor version number
MSB	LSB

7.5.4.10 Object 1019h: Unique Device ID

The object at index 1019h shall contain the Unique Device Identification (UDID) for the device. The UDID shall be composed of the MAC Vendor ID and a vendor's unique device ID.

NOTE Since the uniqueness of the UDID of connected SNs is verified by the SCM, the allocation of the UDID during the vendor's production is not a safety critical production step.

The composition of the UDID shall be as following:

3 octets MAC Vendor Id + 3 octet vendor-specific unique device number (e.g. serial number).

EXAMPLE MAC Vendor ID: 06065h and Serial Number: 471112h gives UDID: 00-60-65-47-11-12.

UDID 00-00-00-00-00-00 shall be reserved.

Object 1019h Unique Device ID is specified in Table 134.

Table 134 – Object 1019h Unique Device ID

Item	Value
Index	1019h
Name	SINF_UniqueDeviceID_OSTR6 [a]
Object type	VAR
Data type	OCTET_STRING6
Value range	
Default value	0
Category	M
Access	Const
SPDO mapping	No
Relevant for SOD checksum	No
[a] The UDID shall be a globally unique identifier which consists of 6 octets.	

7.5.4.11 Object 101Ah: Parameter Download

This object shall be used for receiving an application specific parameter set and also additional parameter sets from the SCM. The SN shall determine which type of parameter set is received and react accordingly.

Object 101Ah Parameter Download is specified in Table 135.

Table 135 – Object 101Ah Parameter Download

Item	Value
Index	101Ah
Name	SSDO_ParameterDownload_DOM
Data type	DOMAIN
Value range	DOMAIN
Default value	0
Category	O
Access	rw
SPDO mapping	No
Relevant for SOD checksum	No

Table 136 specifies the format in which the parameters shall be transferred.

Table 136 – Format of Parameter Download

	Data Type	Len	Description
Item 1	UINT16	2	Index of the object
	UINT8	1	Sub index of the object
	UINT32	4	Length of the object
	DOMAIN	n	Data for the object
Item 2	UINT16	2	Index of the object
	UINT8	1	Sub index of the object
	UINT32	4	Length of the object
	DOMAIN	n	Data for the object
	…	…	…

Table 137 specifies the additional header for the additional parameter sets.

Table 137 – Format of additional parameter header

Data Type	Len	Description
UINT8	1	Parameter set number
UINT8	1	Header version (reserved for future use)
UINT16	2	SAddr of SN
UINT32	4	Octet size for parameter set without header
UINT32	4	CRC for parameter set without header
UINT32	4	Timestamp for parameter set

7.5.4.12 Object 101Bh: SCM Parameters

The object at index $101B_h$ shall contain the SCM specific parameters.

Object 101Bh SCM Parameters is specified in Table 138, Table 139 and Table 140.

Table 138 – Object 101Bh SCM Parameters

Item	Value
Index	101Bh
Name	SSCM_SpecificParameters_REC
Object type	RECORD
Data type	SSCM_SpecificParametersRecord_TYPE
Category	O
SPDO mapping	No

Table 139 – Object 101Bh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	1
Default value	1
Access	ro
SPDO mapping	No
[a] Number of Entries within this index.	

Table 140 – Object 101Bh sub index 01h

Item	Value
Sub index	01h
Name	ConfigurationMode_U8
Data type	UNSIGNED8
Value range	0 – 1
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] SCM configuration mode according to 6.2.2, 0 = ACM, 1 = MCM.	

7.5.4.13 Object 1200h: Common Communication Parameter

This object shall contain the common communication parameters for the device.

Sub index 3 shall contain the consecutive time base for the module. It is not necessary for a device to support all possible time bases. The minimum requirement shall support a time base of 100 µs (type 2).

Object 1200h Common Communication Parameter is specified in Table 141 to Table 146.

Table 141 – Object 1200h Common Communication Parameter

Item	Value
Index	1200h
Name	SCOM_CommonCommParam_REC
Object type	RECORD
Data type	SCOM_CommonCommParamRecord_TYPE
Category	M
Relevant for SOD checksum	Yes

Table 142 – Object 1200h sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	4
Default value	4
Access	ro
SPDO mapping	No
[a] Number of Entries within this index.	

Table 143 – Object 1200h sub index 01h

Item	Value
Sub index	01h
Name	SDN_U16 [a]
Data type	UNSIGNED16
Value range	0 – 1 023
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Safety Domain Number of the device.	

Table 144 – Object 1200h sub index 02h

Item	Value
Sub index	02h
Name	SADR_U16 [a]
Data type	UNSIGNED16
Value range	0 – 1 023
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Safety Address of the SCM within the SD.	

Table 145 – Object 1200h sub index 03h

Item	Value
Sub index	03h
Name	ConsecutiveTimeBase_I8 [a]
Data type	INTEGER8
Value range	0 – 3
Default value	2
Category	M
Access	rw
SPDO mapping	No
[a] Time base identification for CT field, 0 = 1 µs, 1 = 10 µs, 2 = 100 µs, 3 = 1 000 µs.	

Time base of CT field shall be calculated by Formula (1).

$$TimeBase = 1\ \mu s \times 10^{ConsecutiveTimeBase\_I8} \quad (1)$$

where

TimeBase is the time base of CT field in µs;

ConsecutiveTimeBase_I8 is the time base identification.

Table 146 – Object 1200h sub index 04h

Item	Value
Sub index	04h
Name	UDIDofSCM_ OSTR6 [a]
Data type	OCTET_STRING6
Value range	OCTET_STRING6
Default value	UDID of the own SN
Category	M
Access	rw
SPDO mapping	No
[a] UDID of the SCM.	

7.5.4.14 Object 1201h: SSDO Communication Parameter

This object defines the SSDO communication parameters of an SSDO client.

Object 1201h SSDO Communication Parameter is specified in Table 147 to Table 150.

Table 147 – Object 1201h SSDO Communication Parameter

Item	Value
Index	1201h
Name	SSDO_CommParam_REC
Object type	RECORD
Data type	SSDO_CommParamRecord_TYPE
Category	C
Relevant for SOD checksum	No

Table 148 – Object 1201h sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	2
Default value	2
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 149 – Object 1201h sub index 01h

Item	Value
Sub index	01h
Name	Timeout_U32 [a]
Data type	UNSIGNED32
Value range	$1 – 2^{32}$
Default value	500 000
Category	M
Access	rw
SPDO mapping	No
[a] The sub index provides the timeout for an SSDO telegram. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 150 – Object 1201h sub index 02h

Item	Value
Sub index	02h
Name	Retries_U8 [a]
Data type	UNSIGNED8
Value range	0 – 255
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] This sub index provides the number of retries for an SSDO telegram.	

7.5.4.15 Object 1202h: SNMT Communication Parameter

This object defines the SNMT communication parameters of an SNMT master.

Object 1202h SNMT Communication Parameter is specified in Table 151 to Table 154.

Table 151 – Object 1202h SNMT Communication Parameter

Item	Value
Index	1202h
Name	SNMT_CommParam_REC
Object type	RECORD
Data type	SNMT_CommParamRecord_TYPE
Category	C
Relevant for SOD checksum	No

Table 152 – Object 1202h sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	2
Default value	2
Access	ro
SPDO mapping	No
[a] Number of Entries within this index.	

Table 153 – Object 1202h sub index 01h

Item	Value
Sub index	01h
Name	Timeout_U32 [a]
Data type	UNSIGNED32
Value range	$1 – 2^{32}$
Default value	500 000
Category	M
Access	rw
SPDO mapping	No
[a] The sub index provides the timeout for an SNMT telegram. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 154 – Object 1202h sub index 02h

Item	Value
Sub index	02h
Name	Retries_U8 [a]
Data type	UNSIGNED8
Value range	0 – 255
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] This sub index provides the number of retries for an SNMT telegram.	

7.5.4.16 Object 1400h – 17FEh: RxSPDO Communication Parameter

These objects define from which SADR data shall be received.

One producer may have more SADRs (one for each TxSPDO). Therefore one consumer can have more than one RxSPDO for the same physical producer.

1400h – 17FEh RxSPDO Communication Parameter is specified in Table 155 to Table 168.

Table 155 – Object 1400h – 17FEh RxSPDO Communication Parameter

Item	Value
Index	1400h – 17FEh
Name	SPDO_RxCommParam_XXXh_REC
Object type	RECORD
Data type	SPDO_RxCommParamRecord_TYPE
Category	O
Relevant for SOD checksum	Yes

Table 156 – Object 1400h – 17FEh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	0Ch
Default value	0Ch
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 157 – Object 1400h – 17FEh sub index 01h

Item	Value
Sub index	01h
Name	SADR_U16 [a]
Data type	UNSIGNED16
Value range	0 – 1 023
Default value	0
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	
[a] Defines the SADR from which the RxSPDO data is to be received. Setting the SADR to zero deactivates the SPDO.	

Table 158 – Object 1400h – 17FEh sub index 02h

Item	Value
Sub index	02h
Name	SCT_U32 [a]
Data type	UNSIGNED32
Value range	$1 – 2^{32}$
Default value	1
Category	M
Access	rw
SPDO mapping	No
[a] Defines the SCT timer for the data from this RxSPDO (see 7.7.1.2). The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 159 – Object 1400h – 17FEh sub index 03h

Item	Value
Sub index	03h
Name	NumberOfConsecutiveTReq_U8 [a]
Data type	UNSIGNED8
Value range	1 – 63
Default value	1
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	
[a] The sub index provides the number of consecutive time requests. For dynamic SCT configurations the maximum value shall always be used.	

Table 160 – Object 1400h – 17FEh sub index 04h

Item	Value
Sub index	04h
Name	TimeDelayTReq_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] This sub index provides the time delay between two sets of consecutive time requests. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 161 – Object 1400h – 17FEh sub index 05h

Item	Value
Sub index	05h
Name	TimeDelaySync_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	1
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	
[a] This sub index provides the time delay between successful time synchronization and the next time request. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 162 – Object 1400h – 17FEh sub index 06h

Item	Value
Sub index	06h
Name	MinTSyncPropagationDelay_U16 [a]
Data type	UNSIGNED16
Value range	0 – 65 535 in case of dynamic SCT 1 – 65 535 otherwise
Default value	1
Category	M
Access	rw
SPDO mapping	No
[a] The sub index provides the minimum allowed propagation delay between time request and time response. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 163 – Object 1400h – 17FEh sub index 07h

Item	Value
Sub index	07h
Name	MaxTSyncPropagationDelay_U16 [a]
Data type	UNSIGNED16
Value range	0 – 65 535 in case of dynamic SCT 1 – 65 535 otherwise
Default value	1
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	
[a] The sub index provides the maximum allowed propagation delay between time request and time response. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 164 – Object 1400h – 17FEh sub index 08h

Item	Value
Sub index	08h
Name	MinSPDOPropagationDelay_U16 [a]
Data type	UNSIGNED16
Value range	0 – 65 535 in case of dynamic SCT 1 – 65 535 otherwise
Default value	1
Category	M
Access	rw
SPDO mapping	No
[a] The sub index provides the minimum allowed propagation delay between sending and receiving an SPDO. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 165 – Object 1400h – 17FEh sub index 09h

Item	Value
Sub index	09h
Name	MaxSPDOPropagationDelay_U16 [a]
Data type	UNSIGNED16
Value range	0 – 65 535 in case of dynamic SCT 1 – 65 535 otherwise
Default value	1
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	

[a] The sub index provides the maximum allowed propagation delay between sending and receiving an SPDO. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.

Table 166 – Object 1400h – 17FEh sub index 0Ah

Item	Value
Sub index	0Ah
Name	BestCaseTResDelay_U16 [a]
Data type	UNSIGNED16
Value range	0 – 65 535
Default value	0
Category	M
Access	rw
SPDO mapping	No

[a] The sub index provides the best case delay between sending the time request and setting up the time response. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.

Table 167 – Object 1400h – 17FEh sub index 0Bh

Item	Value
Sub index	0Bh
Name	TimeRequestCycle_U32 [a]
Data type	UNSIGNED32
Value range	$1 – 2^{32}$
Default value	1
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	

[a] The sub index provides the time request cycle time. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.

Table 168 – Object 1400h – 17FEh sub index 0Ch

Item	Value
Sub index	0Ch
Name	TxSPDONo_U16 [a]
Data type	UNSIGNED16
Value range	1 – 1 023
Default value	1
Category	M
Access	rw
SPDO mapping	No
[a] Contains the TxSPDO number with which the time request is to be sent.	

7.5.4.17 Object 1800h – 1BFEh: RxSPDO Mapping Parameter

This object defines the RxSPDO mapping parameters. The mapping parameters may be predefined by the vendor and read only. For a detailed description of SPDO mapping, refer to 7.5.4.24.

Object 1800h – 1BFEh RxSPDO Mapping Parameter is specified in Table 169 to Table 172.

Table 169 – Object 1800h – 1BFEh RxSPDO communication parameter

Item	Value
Index	1800h – 1BFEh
Name	SPDO_RxMappParam_XXXh_AU32
Object type	ARRAY
Data type	UNSIGNED32
Category	O
Relevant for SOD checksum	Yes

Table 170 – Object 1800h – 1BFEh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	0 – 254
Default value	0
Access	rw / ro
SPDO mapping	No
[a] Number of Entries within this index.	

Table 171 – Object 1800h – 1BFEh sub index 01h

Item	Value
Sub index	01h
Name	SPDOMapping_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw / ro
SPDO mapping	No
[a] The sub index provides the mapping parameter for this RxSPDO.	

Table 172 – Object 1800h – 1BFEh sub index 02h – FEh

Item	Value
Sub index	02h – FEh
Name	SPDOMapping_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	O
Access	rw / ro
SPDO mapping	No
[a] The sub index provides the mapping parameter for this RxSPDO.	

7.5.4.18 Object 1C00h – 1FFEh: TxSPDO Communication Parameter

Each SN shall have at least one TxSPDO, either for sending data (producer) or for the time request (consumer).

Object 1C00h – 1FFEh TxSPDO Communication Parameter is specified in Table 173 to Table 177.

Table 173 – 1Object C00h – 1FFEh TxSPDO communication parameter

Item	Value
Index	1C00h – 1FFEh
Name	SPDO_TxCommParam_XXXh_REC
Object type	RECORD
Data type	SPDO_TxCommParamRecord_TYPE
Category	M/O
Relevant for SOD checksum	Yes

Table 174 – Object 1C00h – 1FFEh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	3
Default value	3
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 175 – Object 1C00h – 1FFEh sub index 01h

Item	Value
Sub index	01h
Name	SADR_XXXh_U16 [a]
Data type	UNSIGNED16
Value range	0 – 1 023
Default value	0
Category	M
Access	rw
SPDO mapping	No
Relevant for SOD checksum	
[a] Defines the SADR that is used when broadcasting the data of the corresponding TxSPDO within the network. Setting the SADR to zero deactivates the SPDO.	

Table 176 – Object 1C00h – 1FFEh sub index 02h

Item	Value
Sub index	02h
Name	RefreshPrescale_U16 [a]
Data type	UNSIGNED16
Value range	1 – 32 767
Default value	1
Category	M
Access	rw
SPDO mapping	No
[a] The sub index provides the refresh time of the producer. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 177 – Object 1C00h – 1FFEh sub index 03h

Item	Value
Sub index	03h
Name	NumberOfTRes_U8 [a]
Data type	UNSIGNED8
Value range	UNSIGNED8
Default value	0
Category	M/C
Access	rw
SPDO mapping	No
[a] Defines the number of consecutive TRes for a received TReq. If the parameter is zero, the TxSPDO can only be used for TReq telegrams (for devices which are consumers only).	

7.5.4.19 Object 2800h – 2FFFh: User Parameter (writeable at any time)

These objects define user parameters which may be written at any time. These parameters shall not be part of the SOD CRC.

Access to these parameters using SLIM SSDO Services shall be probibited.

The structure of these objects is vendor specific.

7.5.4.20 Object C000h – C3FEh: TxSPDO Mapping Parameter

This object defines the TxSPDO mapping parameters. The mapping parameters may be predefined by the vendor and read only. For a detailed description of the SPDO mapping, refer to 7.5.4.24.

Object C000h – C3FEh TxSPDO Mapping Parameter is specified in Table 178 to Table 199.

Table 178 – Object C000h – C3FEh TxSPDO mapping parameter

Item	Value
Index	C000h – C3FEh
Name	SPDO_TxMappParam_XXXh_AU32
Object type	ARRAY
Data type	UNSIGNED32
Category	M/O
Relevant for SOD checksum	Yes

Table 179 – Object C000h – C3FEh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	0 – 254
Default value	0
Access	rw / ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 180 – Object C000h – C3FEh sub index 01h

Item	Value
Sub index	01h
Name	SPDOMapping_U32 [a]
Object type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw/ro
SPDO mapping	No
Relevant for SOD checksum	
[a] The sub index provides the mapping parameter for this TxSPDO.	

Table 181 – Object C000h – C3FEh sub index 02h – FEh

Item	Value
Sub index	02h – FEh
Name	SPDOMapping_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	O
Access	rw/ro
SPDO mapping	No
[a] The sub index provides the mapping parameter for this TxSPDO.	

7.5.4.21 Object C400h – C7FEh: SADR-DVI List

This object shall be implemented if the SN has SCM functionality. It shall contain a list of all SNs within the SD. The list shall contain the parameters for each SN which are needed to verify the SN during start up. The maximum number of supported SNs shall be vendor specific. Therefore the first two indices (C400h, C401h) shall be mandatory (one SN for the SCM and at least one additional SN), all other shall be optional.

Object C400h – C7FEh SADR-DVI List is specified in Table 182 to Table 197.

Table 182 – Object C400h – C7FEh SADR-DVI list

Item	Value
Index	C400h – C7FEh
Name	SSCM_SADRDVIList_XXXh_REC
Object type	RECORD
Data type	SSCM_SADRDVIListRecord_TYPE
Category	C
Relevant for SOD checksum	No

Table 183 – Object C400h – C7FEh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	0Fh
Default value	0Fh
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 184 – Object C400h – C7FEh sub index 01h

Item	Value
Sub index	01h
Name	SADR_U16 [a]
Data type	UNSIGNED16
Value range	1 – 1 023
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the expected SADR of the corresponding SN.	

Table 185 – Object C400h – C7FEh sub index 02h

Item	Value
Sub index	02h
Name	VendorId_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the Vendor ID of the corresponding SN.	

Table 186 – Object C400h – C7FEh sub index 03h

Item	Value
Sub index	03h
Name	ProductCode_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the product code of the corresponding SN.	

Table 187 – Object C400h – C7FEh sub index 04h

Item	Value
Sub index	04h
Name	RevisionNumber_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the revision number of the corresponding SN.	

Table 188 – Object C400h – C4FEh sub index 05h

Item	Value
Sub index	05h
Name	ModuleStatus_U8 [a]
Data type	UNSIGNED8
Value range	0 – 255
Default value	0
Category	M
Access	ro
SPDO mapping	No
[a] Status of the corresponding SN, 0 = missing, 1 = invalid, 2 = wrong SADR, 3 = UDID mismatch, 4 = wrong parameters, 5 = error additional parameters, 6 = incompatible version, 7 = error init CT, 128 = valid, 129 = OK, all others reserved for future use.	

Table 189 – Object C400h – C7FEh sub index 06h

Item	Value
Sub index	06h
Name	Reserved [a]
Data type	UNSIGNED16
Value range	UNSIGNED16
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Reserved for compatibility reasons.	

Table 190 – Object C400h – C7FEh sub index 07h

Item	Value
Sub index	07h
Name	ParameterSetTimestamp_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Timestamp of parameter set of the corresponding SN.	

Table 191 – Object C400h – C7FEh sub index 08h

Item	Value
Sub index	08h
Name	MaximumSsdoPayloadLen_U16 [a]
Data type	UNSIGNED16
Value range	8 – 240
Default value	8
Category	M
Access	rw
SPDO mapping	No
[a] Maximum length of the payload of an SSDO telegram. This parameter is important because not all SNs might support long SSDO telegrams.	

Table 192 – Object C400h – C7FEh sub index 09h

Item	Value
Sub index	09h
Name	SnmtCrcPollInterval_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	100
Category	M
Access	rw
SPDO mapping	No
[a] This sub index provides the poll interval for the SNMT_SN_set_to_OP command. The time base of this parameter is defined by CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Table 193 – Object C400h – C7FEh sub index 0Ah

Item	Value
Sub index	0Ah
Name	ParameterSetLength_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	O
Access	rw
SPDO mapping	No
[a] Contains the length of the parameter set for the corresponding SN.	

Table 194 – Object C400h – C7FEh sub index 0Bh

Item	Value
Sub index	0Bh
Name	ParameterSet_DOM [a]
Data type	DOMAIN
Value range	DOMAIN
Default value	0
Category	C
Access	rw
SPDO mapping	No
[a] Contains the parameter set for the corresponding SN. This sub index needs to be implemented if sub index 0Ah was implemented.	

Table 195 – Object C400h – C7FEh sub index 0Ch

Item	Value
Sub index	0Ch
Name	OptionalFeatures_U32 [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	O
Access	rw
SPDO mapping	No
[a] This is a bit field where all optional protocols shall be specified (see Table 196).	

Table 196 – Bit field of optional features

Bit	Value	Description
0	X	reserved
1	0	Do not use 40 bit SDPO counter (default)
	1	Use 40 bit SDPO counter
2 – 31	X	Reserved

Table 197 – Object C400h – C7FEh sub index 0Dh

Item	Value
Sub index	0Dh
Name	Deprecated [a]
Data type	UNSIGNED32
Value range	UNSIGNED32
Default value	0
Category	C
Access	rw
SPDO mapping	No
[a] This sub index is not used anymore	

Table 198 – Object C400h – C7FEh sub index 0Eh

Item	Value
Sub index	0Eh
Name	ParameterChecksum_DOM [a]
Data type	DOMAIN
Value range	DOMAIN
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] This object contains the SOD parameter checksum container that shall be calculated as described in 7.5.4.9.3	

Table 199 – Object C400h – C7FEh sub index 0Fh

Item	Value
Sub index	0Fh
Name	RemoteParChecksum_DOM [a]
Data type	DOMAIN
Value range	DOMAIN
Default value	0
Category	C
Access	rw
SPDO mapping	No
[a] Contains the parameter set for the corresponding SN. This can be used to store the information from 1080h/06h for each SN, as it is read during the verification of the DVI.	

7.5.4.22 Object C801h – CBFFh: Additional SADR list

This Object shall describe the addresses (SADR) of an SN. Sub index 1 shall describe the "Main-SADR" to which the corresponding address belongs. Sub index 2 shall describe an additional TxSPDO address. The "Main-SADR" is the SADR of the 1st TxSPDO and is also used by the SSDO services.

Object C801h – CBFFh Additional SADR List is specified in Table 200 to Table 203.

Table 200 – Object C801h – CBFFh Additional SADR list

Item	Value
Index	C801h – CBFFh
Name	SSCM_SADRMembership_XXXh_AU16
Object type	ARRAY
Data type	UNSIGNED16
Category	O
Relevant for SOD checksum	No

Table 201 – Object C801h – CBFFh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	2
Default value	2
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 202 – Object C801h – CBFFh sub index 01h

Item	Value
Sub index	01h
Name	SADR_U16 [a]
Data type	UNSIGNED16
Value range	1 – 1 023
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the "Main-SADR" which uses the SADR for a TxSPDO.	

Table 203 – Object C801h – CBFFh sub index 02h

Item	Value
Sub index	02h
Name	TxSPDONo_U16 [a]
Data type	UNSIGNED16
Value range	1 – 1 023
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the TxSPDO number, the SADR is used for.	

EXAMPLE

Given the configuration shown in Table 204 below, this is to be interpreted as follows:

1st line: SADR 1 (Index – C800h) belongs to the SN with "Main-SADR" 1 and is used in TxSPDO 1

2nd line: SADR 2 (Index – C800h) belongs to the SN with "Main-SADR" 2 and is used in TxSPDO 1

3rd line: SADR 3 (Index – C800h) belongs to the SN with "Main-SADR" 3 and is used in TxSPDO 1

4th line: SADR 4 (Index – C800h) belongs to the SN with "Main-SADR" 1 and is used in TxSPDO 2

5th line: SADR 5 (Index – C800h) belongs to the SN with "Main-SADR" 1 and is used in TxSPDO 3

6th line: SADR 6 (Index – C800h) belongs to the SN with "Main-SADR" 0, 0 is not a valid SADR -> therefore this SADR does not belong to any SN

Summary: the SN with "Main-SADR" 1 has 3 TxSPDOs using SADR 1, 4 and 5. SN 2 and 3 only have one SADR (therefore only 1 TxSPDO).

Table 204 – Object Additional SADR List Example

Index (hex)	Sub index 0	Sub index 1	Sub index 2
C801	2	1	1
C802	2	2	1
C803	2	3	1
C804	2	1	2
C805	2	1	3
C806	2	0	0

7.5.4.23 Object CC01h – CFFFh: SADR-UDID List

These objects shall contain information specifying which UDIDs can belong to which SADR. One logical SN (described with the "Main-SADR") can be switched between more than one physical device (modular machine part). All valid devices (described by their UDID) are listed within these objects.

Object CC01h – CFFFh SADR-UDID List is specified in Table 205, Table 206 and Table 207.

Table 205 – Object CC01h – CFFFh SADR-UDID list

Item	Value
Index	CC01h – CFFFh
Name	SSCM_SADRUDIDList_XXXh_AOSTR6
Object type	ARRAY
Data type	OCTET_STRING6
Category	O
Relevant for SOD checksum	No

Table 206 – Object CC01h – CFFFh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	1 – 254
Default value	1
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 207 – Object CC01h – CFFFh sub index 01h – FEh

Item	Value
Sub index	01h – FEh
Name	UDID_OSTR6 [a]
Data type	OCTET_STRING6
Value range	OCTET_STRING6
Default value	0
Category	M
Access	rw
SPDO mapping	No
[a] Contains the UDIDs which are allowed for the corresponding SADR.	

EXAMPLE Given the configuration shown in Table 208, this is to be interpreted as follows. For SADR 0001 (CC01h), two UDIDs are allowed. Either UDID 50-aa-c0-43-56-3f or 50-aa-c0-43-56-42 is accepted as UDID for the module with SADR 0001.

Table 208 – SADR-UDID List Example

Index (hex)	Sub index 0	Sub index 1	Sub index 2
CC01	2	50-aa-c0-43-56-3f	50-aa-c0-43-56-42

7.5.4.24 Object E400h – E7FEh: Additional parameter list

This object contains a list of all additional parameter sets per SN within the SD and shall be present, if the SN has SCM functionality. The maximum number of parameter sets per SN shall be 16. Each element shall correspond to it's SADDR, therefore the address E400h references SADDR 1, E401h references SADDR 2 and so on. Foreach parameter list entry, the index of the entry, the length of the parameter set and the parameter set itself shall be specified.

All parameter sets shall be preceeded by their corresponding headers. For these headers see Table 137.

An SN shall check if the additional parameters transmitted are valid and usable. If the SN expects additional paramters and the SCM fails to send them the SN shall not go into operational.

Object E400h – E7FEh SSCM additional parameter list is specified from Table 209 to Table 211.

Table 209 – Object E400h – E7FEh SSCM additional parameter list

Item	Value
Index	E400h – E7FEh
Name	SSCM_AddParaList_*XXX*h_ADO
Object type	ARRAY
Data type	OCTET_STRING6
Category	C
Relevant for SOD checksum	No

Table 210 – Object E400h – E7FEh sub index 00h

Item	Value
Sub index	00h
Name	NumberOfEntries [a]
Value range	0 – 16
Default value	0
Access	ro
SPDO mapping	No
[a] Number of entries within this index.	

Table 211 – Object E400h – E7FEh sub index 01h – 10h

Item	Value
Sub index	01h – 10h
Name	ParameterSet [a]
Data type	DOMAIN
Value range	DOMAIN
Default value	-
Category	O
Access	rw
SPDO mapping	No
[a] Contains a parameter set for this entry.	

7.6 Safety related PDO mapping

7.6.1 General

The real-time data transfer shall be performed by means of Safety Process Data Objects (SPDO).

Data length and mapping of application objects into SPDOs shall be determined by the corresponding SPDO mapping structures within the Safety Object Dictionary (SOD). The mapping of application objects into an SPDO may be transmitted to a device during the device configuration process by applying the SSDO services to the corresponding entries of the Object Dictionary.

The length of SPDOs of a device is application-specific and shall be determined by the corresponding mapping object.

There are two uses for SPDOs. The first is data transmission and the second data reception. Transmit SPDOs (TxSPDOs) and Receive SPDOs (RxSPDOs) are distinguished. Devices supporting TxSPDOs are SPDO producers and devices which are able to receive SPDOs are called SPDO consumers.

SPDOs are described by the SPDO communication parameter and the SPDO mapping parameter. For each SPDO, the pair of communication and mapping parameters shall be mandatory.

SPDO communication parameter shall describe the communication capabilities of the SPDO.

SPDO mapping parameter shall describe mapping for each object contained in SPDO payload data to object dictionary entries and vice versa.

SPDO mapping shall be variable or static. Static mapping shall be pre-defined by the vendor and shall not be modified in any way. Variable mapping shall be configured during commissioning.

7.6.2 Transmit SPDOs

The TxSPDO communication parameters shall be specified by SOD indices 1C00h – 1FFEh (SPDO_TxCommParam_XXXh_REC). An SN may support up to 1 023 TxSPDOs. The implementation of the first TxSPDO shall be mandatory. The amount of additional TxSPDOs is vendor specific. The TxSPDO mapping parameters is described by indices C000h – C3FEh (SPDO_TxMappParam_XXXh_AU32). The data to be transmitted shall be assembled according to the corresponding mapping data.

Sending SPDO data shall be implicitly limited to the Operational state. Refer to 7.5 for the corresponding entries within the SOD.

7.6.3 Receive SPDOs

The RxSPDO communication parameters shall be specified by indices 1400h – 17FEh (RxSPDOCommParam_XXXh_REC). An SN may support up to 1 023 RxSPDOs. The implementation of RxSPDOs shall be optional (e.g. an input module, which is a simple producer, does not need to have an RxSPDO). The RxSPDO mapping parameters is described by indices 1800h – 1BFEh (SPDO_RxMappParam_XXXh_AU32). The received data shall be transferred to the communication objects assigned to them by the RxSPDO mapping parameters. The application accesses the received SPDO data by reading these communication objects.

Receiving SPDO data shall be implicitly limited to the Operational state. Refer to 7.5 for the corresponding entries within the SOD.

7.6.4 SPDO mapping parameter

The indices 1800h – 1BFEh (SPDO_TxMappParam_XXXh_AU32) and C000h – C3FEh (SPDO_RxMappParam_XXXh_AU32) shall contain the mapping parameters for transmitting and receiving SPDOs. For both indices, sub index 0 shall contain the number of valid mapping entries which is also the number of communication variables (SOD entries) to be transmitted or received. The sub indices 01h to NumberOfEntries shall contain information about the mapped application variables. These entries shall describe the SPDO contents by their index, sub index and length. All three values shall be given in hexadecimal notation. The encoding of a mapping entry (sub index 01_h – FE_h) is specified by Table 212. The length entry contains the length of the object in bits (01_h – FF_h). This parameter can be used to check the overall mapping length.

Table 212 – Structure of SPDO mapping entry

Bit 31 – bit 16	Bit 15 – bit 8	Bit 7 – bit 0
Index (16 bit)	sub index (8 bit)	data length (8 bit)

The commissioner shall be responsible for the correctness of the mapping data. If objects are mapped which shall not be mapped according to the object attribute, the SN shall not switch to state Operational. If data within an RxSPDO is not used by the consumer, it shall be mapped to the data types (index 0001h – 0007h), these serve as "dummy entries". The corresponding data within the RxSPDO shall not be evaluated by the device. This feature is useful if a TxSPDO is used for several consumers, each only utilizing parts of the transmitted data. It shall not be possible to map those "dummy entries" into a TxSPDO.

7.6.5 SPDO mapping example

Example devices (see Figure 51) are:

– the SN with "Main-SADR" 12 is an input module with 32 digital inputs,
– the SN with "Main-SADR" 13 is an output module with 8 digital outputs, and
– the SN with "Main-SADR" 14 is an output module with 16 digital outputs.

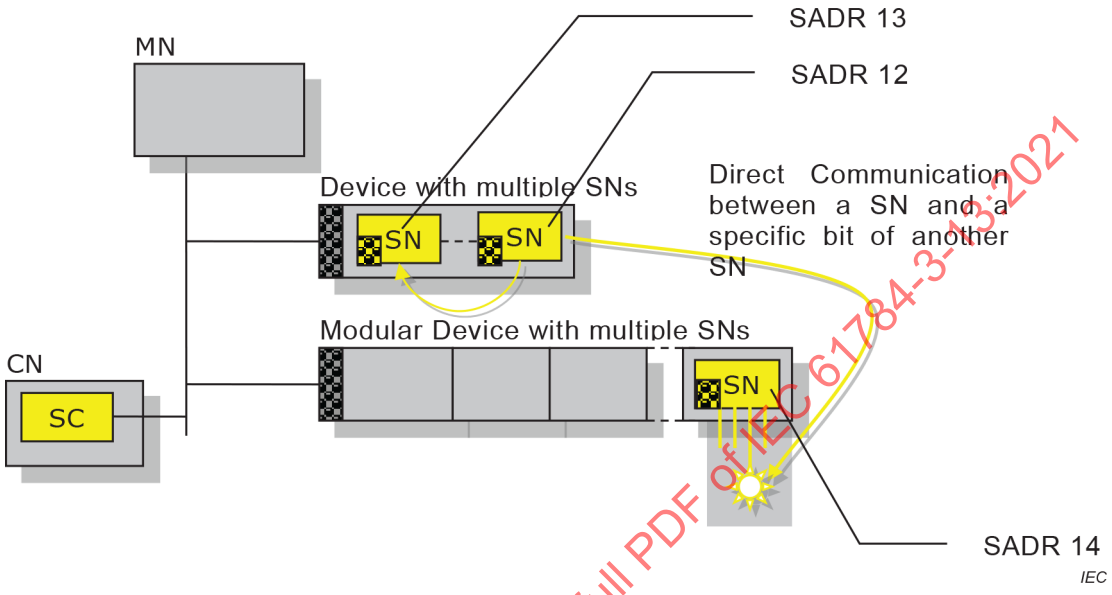


Figure 51 – SPDO mapping example

It is assumed that the inputs are available at SOD-index 6000h sub indices 01h – 04h and that the data type is UNSIGNED8. It is also assumed that the outputs need to be written to SOD index 6200h sub indices 01h respectively 01h and 02h. The data type again is UNSIGNED8. It is then assumed that SN 13 needs the information from the 3rd input octet and that SN 14 needs the information from the 1st and 4th input octet.

The communication parameters from SN 12 (producer) are shown in Table 213.

Table 213 – Mapping example table 1

Index	Sub index	Data	Description
1C00h	1	12	The TxSPDO is broadcast under the SADR 12

The mapping of SN 12 (producer) may be specified as in Table 214.

Table 214 – Mapping example table 2

Index	Sub index	Data	Description
C000h	0	4	four valid mapping entries
	1	60000108h	Mapping entry 1: 8bits of the object from index 6000h sub index 1 are mapped into the SPDO
	2	60000208h	Mapping entry 2: 8bits of the object from index 6000h sub index 2 are mapped into the SPDO
	3	60000308h	Mapping entry 3: 8bits of the object from index 6000h sub index 3 are mapped into the SPDO
	4	60000408h	Mapping entry 4: 8bits of the object from index 6000h sub index 4 are mapped into the SPDO

This TxSPDO mapping could be a predefined mapping. Input 2 is mapped into the TxSPDO although it is not needed by any of the SNs.

According to the mapping parameters, the payload data of the TxSPDO looks as in Table 215.

Table 215 – Mapping example table 3

Octet	Data
1	Input Octet 1
2	Input Octet 2
3	Input Octet 3
4	Input Octet 4

The communication parameter for SN 13 is specified as in Table 216.

Table 216 – Mapping example table 4

Index	Sub index	Data	Description
1400h	1	12	SN 13 will map the data of a received TxSPDO from SADR 12 according to the mapping entries

The mapping of the SN 13 looks as in Table 217.

Table 217 – Mapping example table 5

Index	Sub index	Data	Description
1800h	0	2	two valid mapping entries
	1	00060010h	Mapping entry 1: 16 bits of the RxSPDO are mapped into the object at index 0006h sub index 0
	2	62000108h	Mapping entry 2: 8bits of the RxSPDO are mapped into the object at index 6200h sub index 1
	3	00050008h	Mapping entry 2: 8bits of the RxSPDO are mapped into the object at index 0005h sub index 0

As described above, SN 13 is only interested in the 3rd input octet of SN 12. Therefore, the first two input octets (first two octets of the RxSPDO) are ignored by using "dummy mapping". The 4th octet is also mapped to a "dummy entry". This is required because only RxSPDOs which have the same length as the mapped data are accepted.

The communication parameter of SN 14 is shown in Table 218.

Table 218 – Mapping example table 6

Index	Sub index	Data	Description
1400h	1	12	SN 14 will map the data of a received TxSPDO from SADR 12 according to the mapping entries

The mapping of the SN 14 is shown in Table 219.

Table 219 – Mapping example table 7

Index	Sub index	Data	Description
1800h	0	3	three valid mapping entries
	1	62000108h	Mapping entry 1: 8bits of the RxSPDO are mapped into the object at index 6200h sub index 1
	2	00060010h	Mapping entry 2: 16 bits of the RxSPDO are mapped into the object at index 0006h sub index 0
	3	62000208h	Mapping entry 3: 8bits of the RxSPDO are mapped into the object at index 6200h sub index 2

As described above, SN 14 is only interested in input octets 1 and 4 of SN 12. Input octets 2 and 3 (respectively octet 2 and 3 of the RxSPDO) are ignored by using "dummy mapping".

7.6.6 SDPO Feature Set

7.6.6.1 General

A producer may suggest the usage of additional SPDO features to all listening consumers. This shall only be valid, if the corresponding consumer is also handling time synchronization.

All requested features shall be acknowledged by the corresponding consumer. As soon as the acknowledgment has been received by the producer, the features shall be activated automatically. However if the producer receives more than one time request per TxSPDO, the features shall not be enabled.

7.6.6.2 SPDO Features

Table 220 specifies the supported features.

Table 220 – SPDO telegram features (TR field, bits 2-7)

Bit number of TR field								Frame type
7	6	5	4	3	2	1	0	
1	X	X	X	X	X	Bit 8, 9 of the TADR field		Request 40 bit counter
X	1	X	X	X	X			40 bit counter active

Bits 2-5 are reserved and shall not be used.

7.6.6.3 Feature Set Request

Feature sets may only be requested while the connection valid bit is not set and are valid from the moment they are acknowledged onwards.

Once the usage of a feature set has successfully been agreed between producer and consumer, it shall become invalid as soon as the SPDO, to which the feature set has been applied, switches its connection valid state from set to not-set. The feature set shall also become invalid as soon as either communication partner switches to pre-operational or a powerfail occurs. It may be renegotiated during the next SPDO synchronization cycle, after it has become invalid.

The request shall be transmitted by the producer using an SPDO data only telegram. In this telegram, as long as no time synchronization request has been received, the feature set is encoded via an XOR operation on the TR field of PDU part two. The connection valid bit shall not be set during negotiation. Feature set negotiations on valid connections are not permitted.

If the connection valid bit is set to true, the feature bits shall be still encoded on the TR field as an information to the consumer, and the consumer subsequently shall keep sending the acknowledgment. This is an information to both ends that the feature is activated.

7.6.6.4 Feature Set Acknowledgement

The consumer shall acknowledge a feature before it may be used actively. This shall be done by encoding the feature set on the ID field of PDU part two on all telegrams for time synchronization requests to the producer requesting the feature set.

As soon as the acknowledgement ist sent the consumer shall apply the feature sets to all further SPDO data communications.

7.6.7 SPDO error handling

7.6.7.1 Non-mapable application object

The SN shall ensure that only mapable application objects are accepted. If a non mapable object is recognized, the SN shall not store this mapping data into the SOD.

NOTE As a result, the checksum of the SOD will be different from the expected checksum stored on the SCM and therefore, the SN will not be switched to the Operational state (see 7.7.12.2).

7.6.7.2 Unexpected length of RxSPDO

If an SPDO is received with a length that is not equal to the length of the mapped data, the SPDO shall be ignored. Additional diagnosis data may be provided via the SOD.

7.7 State and sequence diagrams

7.7.1 Safety Process Data Object (SPDO)

7.7.1.1 Safety Process Data Object producer (TxSPDO)

The producer shall start sending its data cyclically or on state changes when switching to Operational.

NOTE The refresh time for sending the data can be adjusted within the SOD.

The Safety Process Data Object Producer state diagram is specified by Figure 52.

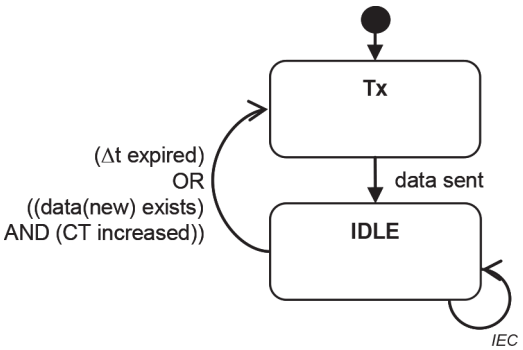
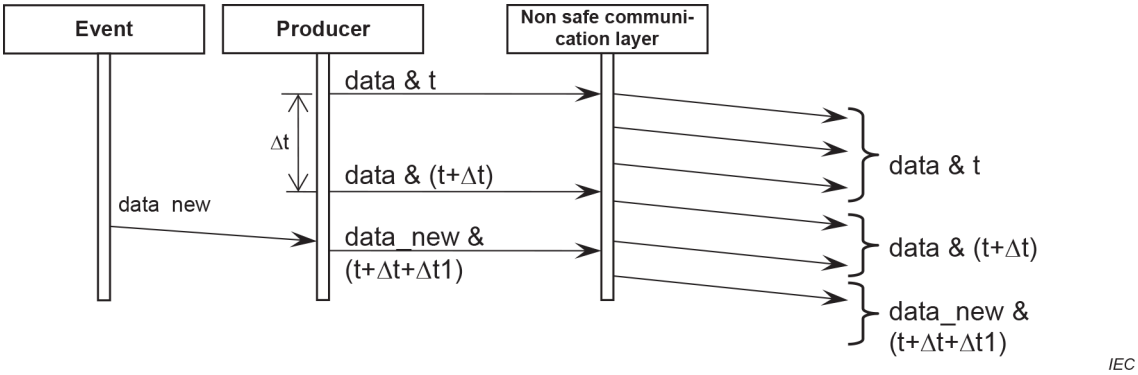


Figure 52 – State diagram TxSPDO

The Safety Process Data Object Producer sequence diagram is specified by Figure 53.



Key

See Table 221.

Figure 53 – SPDO communication producer

Table 221 specifies items used in Figure 53.

Table 221 – SPDO communication producer item description

Item	Description	Min value	Max value	SOD
data_new	New data from safety related application or new TRes received	-	-	-
t	Internal time [CT]	0	65 535	-
Δt	Refresh prescale [CT]	1	32 767	Object Index 1C00h – 1FFEh sub index 02h (see 7.5.4.18)
Δt1	Time difference of new data [CT]	>0	Δt	-

Table 222 specifies the SPDO communication producer states.

Table 222 – SPDO communication producer state description

State	Description
Tx	Sending data
IDLE	Wait until refresh time elapsed or new data available

7.7.1.2 Safety Process Data Object consumer (RxSPDO)

7.7.1.2.1 General behaviour

For each received RxSPDO action as specified in the Safety Process Data Object Producer, state diagram (Figure 54) shall be taken by the consumer. The data shall be processed. Processing data shall include time synchronization and time validation, see 7.7.1.2.2. If a consumer receives valid data from a producer, the SCT timer shall be restarted. If the SCT timer elapses, all data related to the RxSPDO shall be set to a safe state.

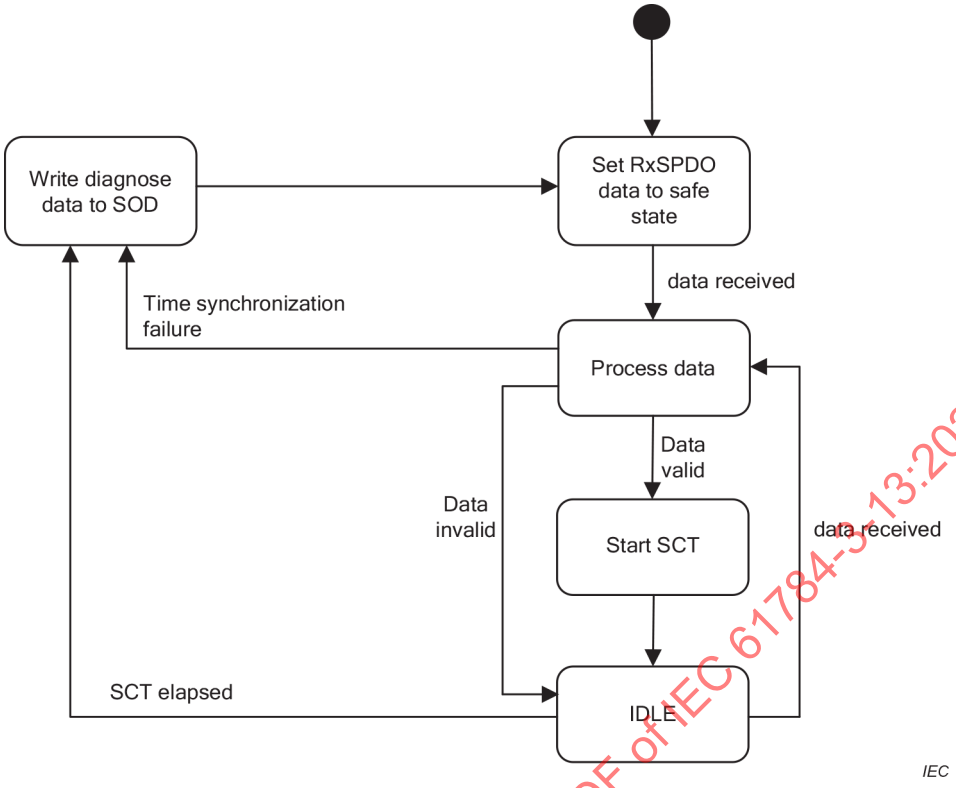
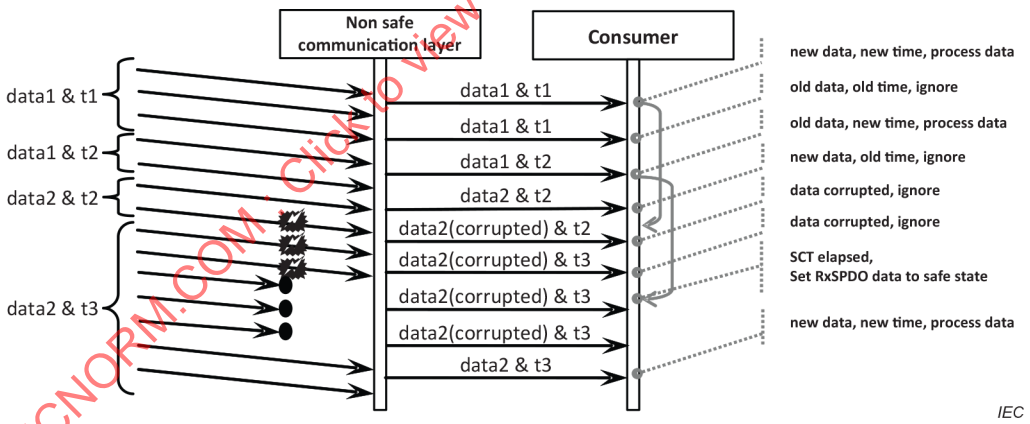


Figure 54 – State diagram RxSPDO

The Safety Process Data Object Consumer sequence diagram is specified by Figure 55.



Key

See Table 223.

Figure 55 – SPDO communication consumer

Table 223 specifies the item used in Figure 55.

Table 223 – SPDO communication consumer item description

Item	Description	Min value	Max value	SOD
SCT	Safety control time [CT]	1	UNSIGNED32	Object 1400h – 17FEh sub index 02h (see 7.5.4.16)

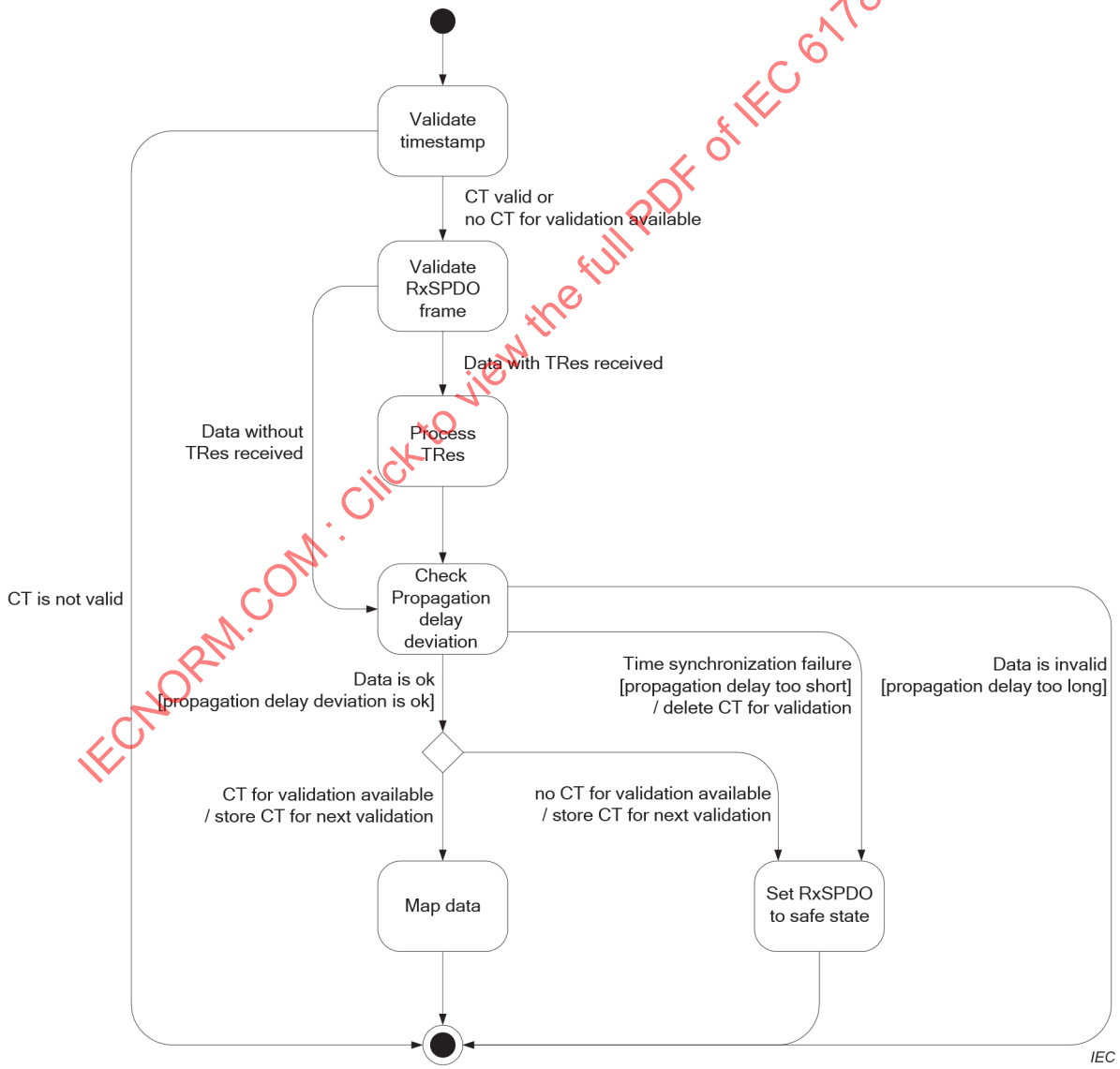
Table 224 specifies the SPDO communication consumer states.

Table 224 – SPDO communication consumer state description

State	Description
IDLE	Waiting for data
Process data	Performing timing validation and time synchronization. Process data according to SPDO mapping
Start SCT	Reset the SCT
Write diagnose data to SOD	The diagnose data has to be available for the user. This is made through the SOD
Set RxSPDO data to safe state	All data related to the producer will be set to zero

7.7.1.2.2 Process data

The process data state diagram is specified by Figure 56.



Key

See Table 225 and Table 226.

Figure 56 – State diagram process data

Table 225 specifies items used in Figure 56.

Table 225 – SPDO communication consumer telegram validation item description

Item	Description	Min value	Max value	SOD
CT	Consecutive Time field	0	65 535	-
TRes	Time synchronization response, consists of TR (Time Request Distinctive Number Field) and TADR (Time Request Address) within the safety PDU	-	-	-

Table 226 specifies the process data states.

Table 226 – SPDO communication consumer telegram validation state description

State	Description
Validate timestamp	If the timestamp (CT) is equal to or less than the timestamp of the previously valid telegram, the telegram shall be ignored
Validate RxSPDO PDU frame	Checks if the PDU contains time synchronization data for this node
Process TRes	See 7.7.2.7
Check propagation delay deviation	Checks timestamp validity
Set RxSPDO to safe state	All cyclic data related to this RxSPDO shall be set to the safe state
Map data	Writes the data into the SOD according to the RxSPDO mapping

7.7.2 Time synchronization and validation

7.7.2.1 General

To verify the performance of the non safety communication layer, FSCP 13/1 shall use a combined time synchronization and validation sequence. This shall be done cyclically.

NOTE Cyclically because of tolerances of the hardware clocks within the safety nodes.

Figure 57 specifies the basic sequences for the time synchronization and validation.

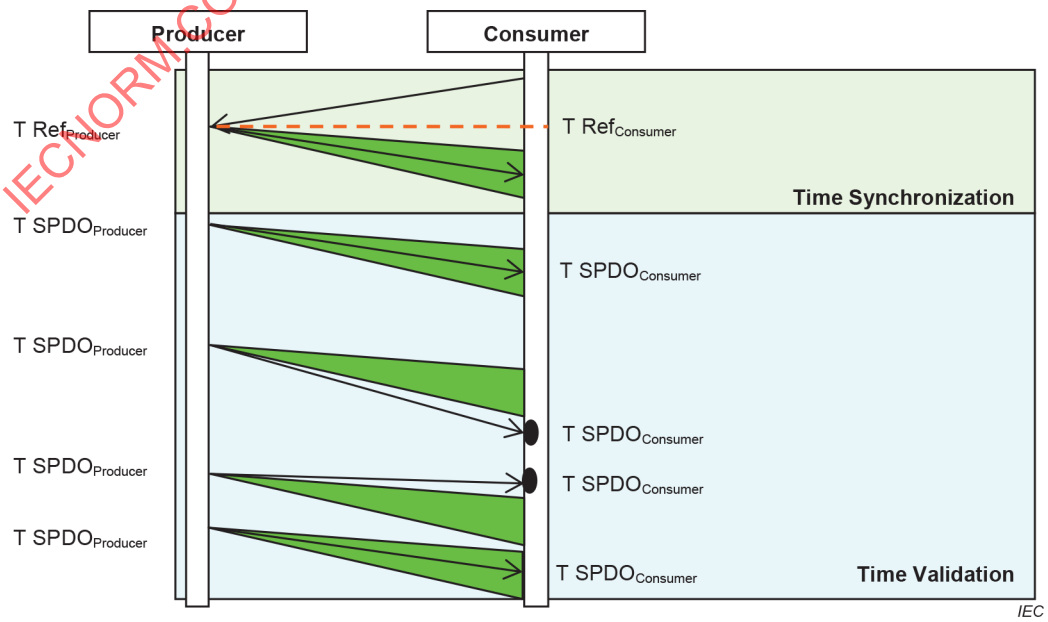
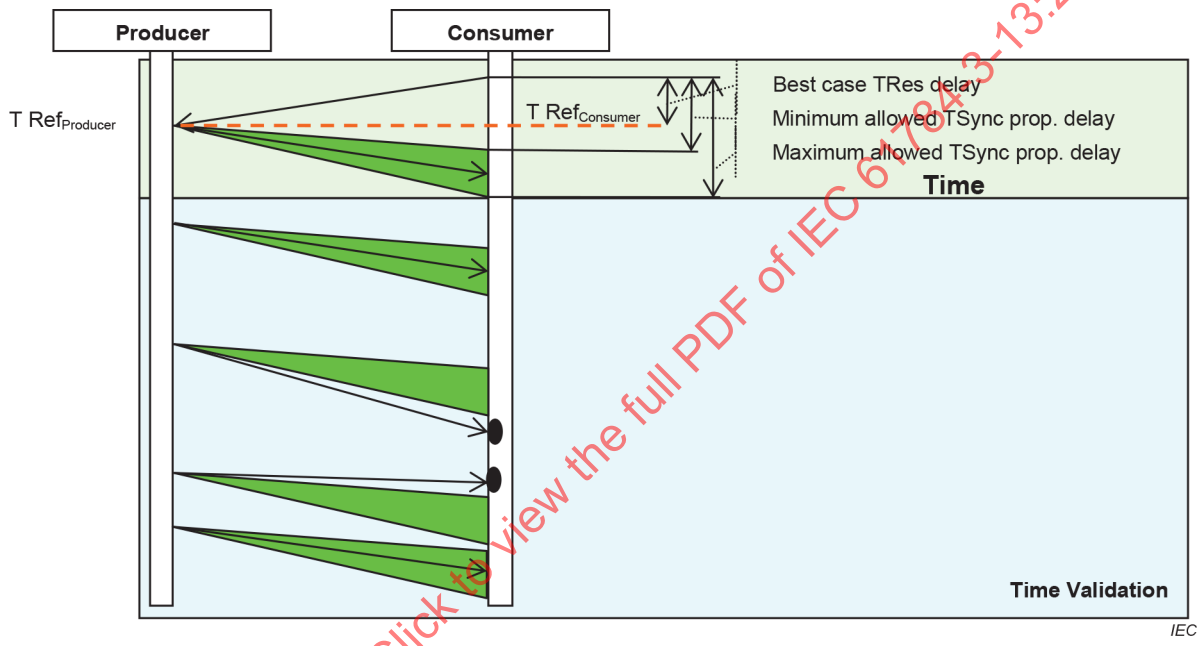


Figure 57 – Time synchronization and validation

7.7.2.2 Time synchronization

The consumer shall start the sequence by sending a set of Time Requests (TReq) to the producer. This shall be done by using a TxSPDO. When receiving the first TReq, the producer shall create a "new data event" (see 7.7.1.1) to reply to the request as soon as possible with the corresponding set of Time Responses (TRes). Then the first received TRes within the consumer shall be checked against the minimum and maximum allowed TSync propagation delay. If the delay is shorter than the minimum allowed delay, the consumer shall enter the safe state, because in this case the best case TRes delay parameter is incorrect. If the delay is longer than the maximum allowed delay, the time response shall be ignored. If delay is within the given limits, time synchronization shall be considered successful and the consumer shall memorize the internal T $Ref_{Producer}$ and T $Ref_{Consumer}$ values.

Figure 58 specifies the synchronization sequence.



Key

See Table 227.

Figure 58 – Time synchronization detail

Table 227 specifies items used in Figure 58.

Table 227 – Time synchronization item description

Item	Description	Min value	Max value	SOD
T $Ref_{Producer}$	Value of CT field within the TRes service supplied by the producer	0	65 535	-
best case TRes delay	User defined value to define the best case minimum time needed to transfer the time request from the consumer to the producer and to set up the time response within the producer. This value is used to calculate the SPDO propagation delay. If the user assumes a larger value than the real value, the time validation would not recognize impermissible delays in the SPDO propagation	0	65 535	Object index 1400h – 17FEh sub index 0Ah (see 7.5.4.16)
T $Ref_{Consumer}$	Time Request start time [CT] + best case TRes delay [CT]. This value is calculated within the consumer and gives a time reference within the consumers CT to T $Ref_{Producer}$	> best case TRes delay	65 535	-
Minimum allowed TSync propagation delay	Minimum allowed propagation delay for time synchronization [CT]	1	65 535	Object index 1400h – 17FEh sub index 06h (see 7.5.4.16)
Maximum allowed TSync propagation delay	Maximum allowed propagation delay for time synchronization [CT]	1	65 535	Object index 1400h – 17FEh sub index 07h (see 7.5.4.16)

7.7.2.3 Time validation

After successful time synchronization, the consumer shall receive SPDOs and calculate the propagation delay as specified by Formulae (2), (3) and (4).

$$\text{SPDO propagation delay} = T\_SDPO\_to\_Ref_{Consumer} - T\_SDPO\_to\_Ref_{Producer} \quad (2)$$

$$T\_SDPO\_to\_Ref_{Consumer} = T\ SPDO_{Consumer} - T\ Ref_{Consumer} \quad (3)$$

$$T\_SDPO\_to\_Ref_{Producer} = T\ SPDO_{Producer} - T\ Ref_{Producer} \quad (4)$$

where

$T\ SPDO_{Producer}$ is the value of CT field within the SPDO supplied by the producer;

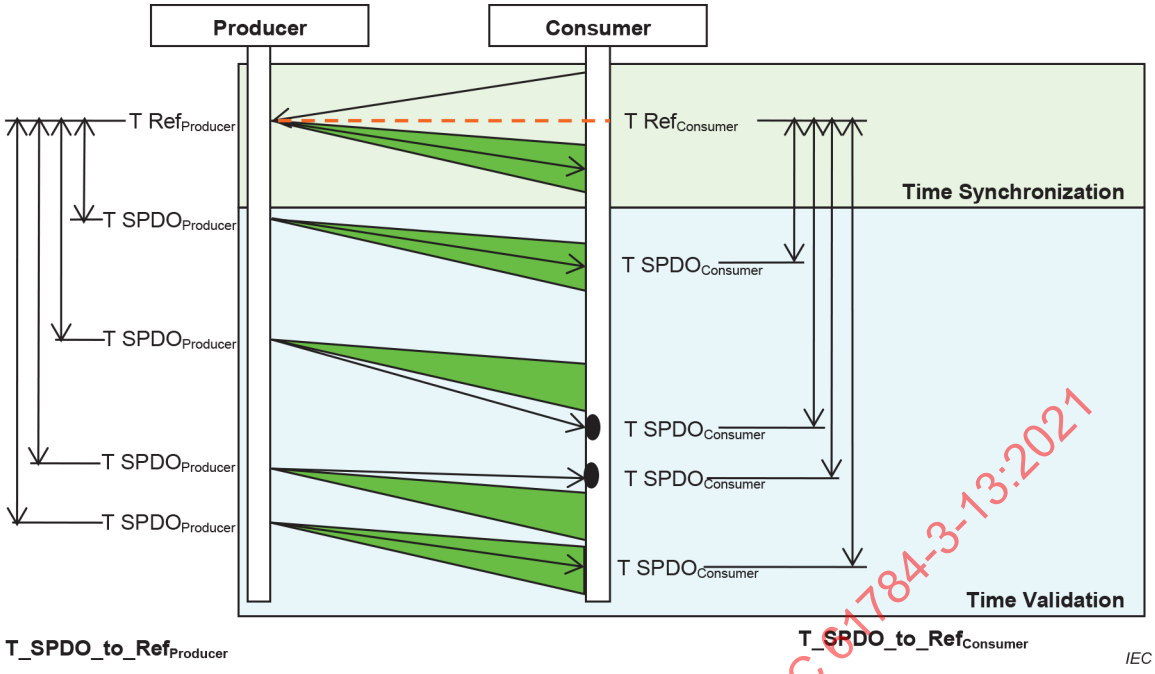
$T\ SPDO_{Consumer}$ is the CT value within the consumer when receiving the SPDO;

$T\ Ref_{Producer}$ is the value of CT field within the TRes service supplied by the producer;

$T\ Ref_{Consumer}$ is the time reference within the consumers CT to T RefProducer (see Table 227).

The result of Formula (2) shall be checked against the minimum and maximum allowed SPDO propagation delay. If the delay is shorter than the minimum allowed delay, the consumer shall enter the safe state, because in this case the best case TRes delay parameter was set wrong. If the delay is longer than the maximum allowed delay, the SPDO shall be ignored. If delay is within the given limits, the SPDO shall be processed.

Figure 59 shows how *SPDO propagation delay* is calculated.

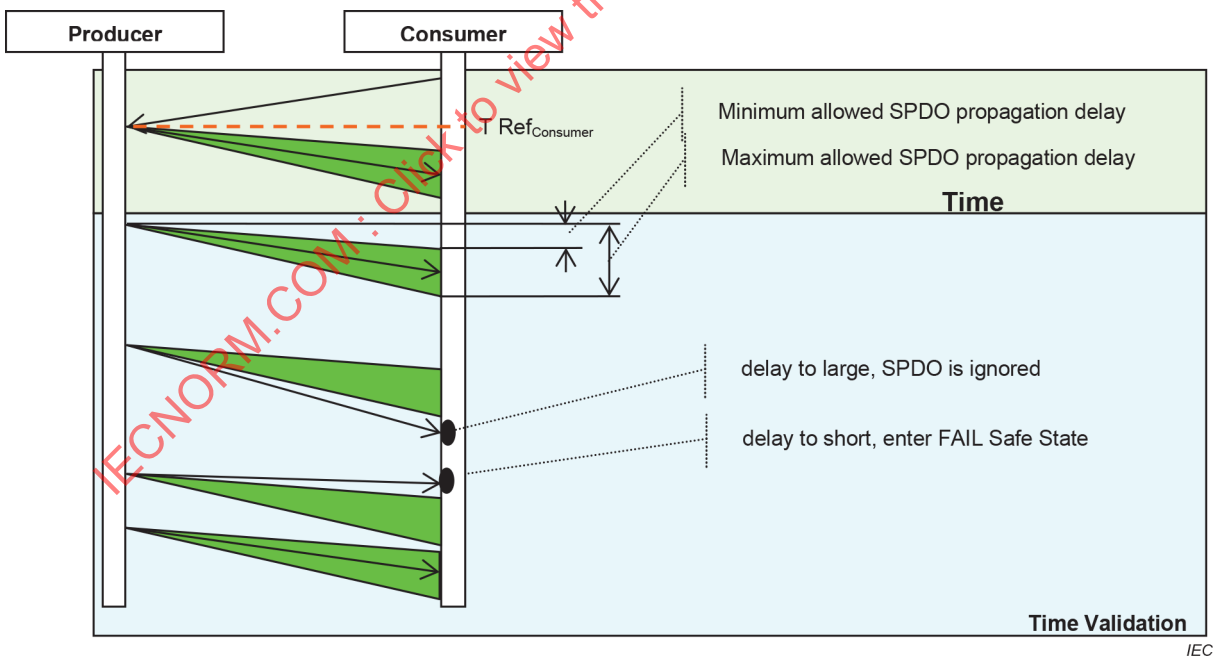


Key

See Table 228.

Figure 59 – Calculation of propagation delay

Figure 60 specifies the time validation sequences.



Key

See Table 228.

Figure 60 – Time validation, propagation delay explanation limits

Table 228 specifies items used in Figure 59 and Figure 60.

Table 228 – Time validation item description

Item	Description	Min value	Max value	SOD
T $SPDO_{Producer}$	Value of CT field within the SPDO supplied by the producer	0	65 535	-
T $SPDO_{Consumer}$	CT value within the consumer when receiving the SPDO	0	65 535	-
SPDO propagation delay	Calculated propagation delay using the time references of the previous time synchronization, see Formula (2)	0	65 535	-
Minimum allowed SPDO propagation delay	Minimum allowed propagation delay for SPDO [CT]	1	65 535	Object index 1400h – 17FEh sub index 8h (see 7.5.4.16)
Maximum allowed SPDO propagation delay	Maximum allowed propagation delay for SPDO [CT]	1	65 535	Object index 1400h – 17FEh sub index 9h (see 7.5.4.16)

7.7.2.4 Time synchronization operation

To increase the immunity of FSCP 13/1 against data loss within the non safe communication layer, the TReq and TRes services shall be sent as bundle of single services.

The consumer shall send m time requests (TxSPDO) to the producer containing the TR (Time Request Distinctive Number). This number shall be incremented with each time request. The producer shall receive the time request from a specific node and shall start answering the time request. The response to a time request shall be to fill the TADR (SADR of the requesting node) and the TR field in the cyclic TxSPDO telegram from the producer.

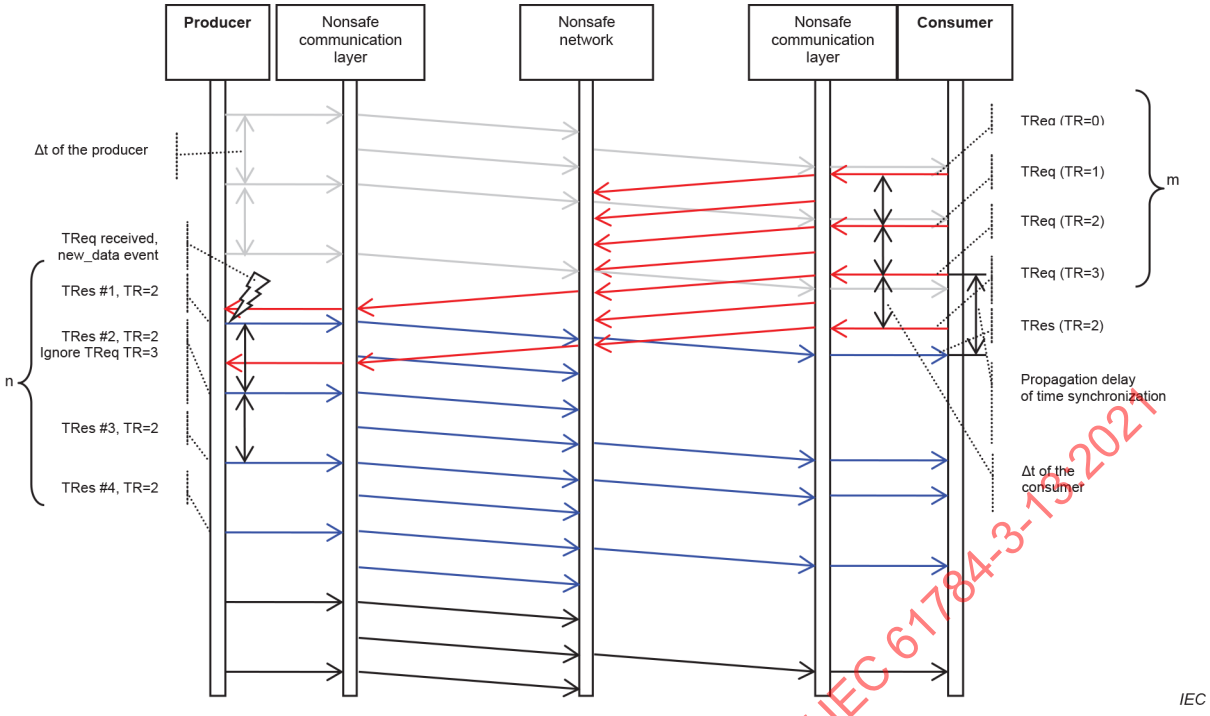
The producer shall repeat the response for a received time request n times although new time requests were already received. This ensures that the consumer which initiates the time request receives the correct answer. During processing the Time Response, the producer shall ignore all other time requests corresponding to this TxSPDO.

If the consumer does not receive the time response within time td (see Table 229) the next set of time requests shall be sent. The number of time responses per time request, the number of requests per consecutive request set, td and the timer ts (see Table 229) which re-initiates a time request shall be adjusted within the SOD. If the propagation delay is too long the TRes shall be ignored. If it is too short, the safe state shall be entered (see 7.7.2.1). If a valid TRes was not received within the Time request cycle, a time synchronization failure shall occur.

The time synchronization for each producer shall start when the consumer is Operational and receives the first RxSPDO from the corresponding producer.

Within a Consumer / Producer relationship, equal CT values shall be mandatory.

Figure 61 specifies time request and response. The consumer shall send a time request to one of its producers. This request is repeated m times. The producer shall answer the first time request, and it receives and repeats the answer n times. If a producer receives a time request during the time it is already answering another time request on the corresponding TxSPDO, the new time request shall be ignored. Figure 61 also shows that the nonsafe communication layer may also drop or delay messages due to different network cycle times or other effects.

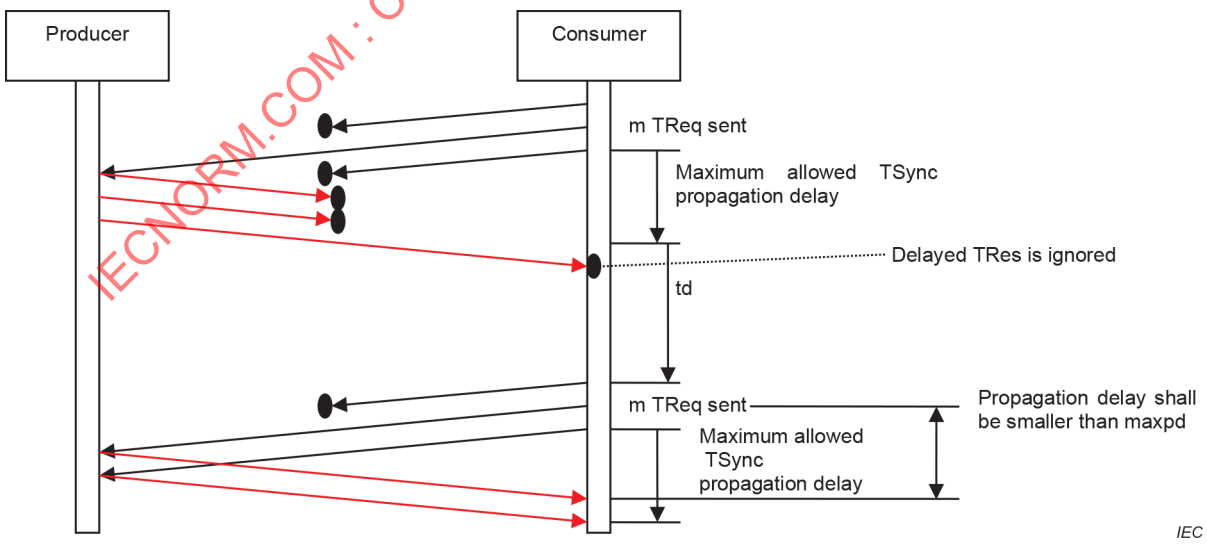


Key

See Table 229.

Figure 61 – Time synchronization on a nonsafe network

Time synchronization is specified by Figure 62. The consumer shall send a set of m time requests (TReq) and then wait for the response. If the response time is shorter than the minimum allowed propagation delay, the safe state shall be entered. If the response (TRes) is within the maximum and minimum allowed propagation delay, it shall be valid. If no response is received within the maximum allowed propagation delay, the consumer shall wait until td has elapsed and shall send the next set of m TReq.

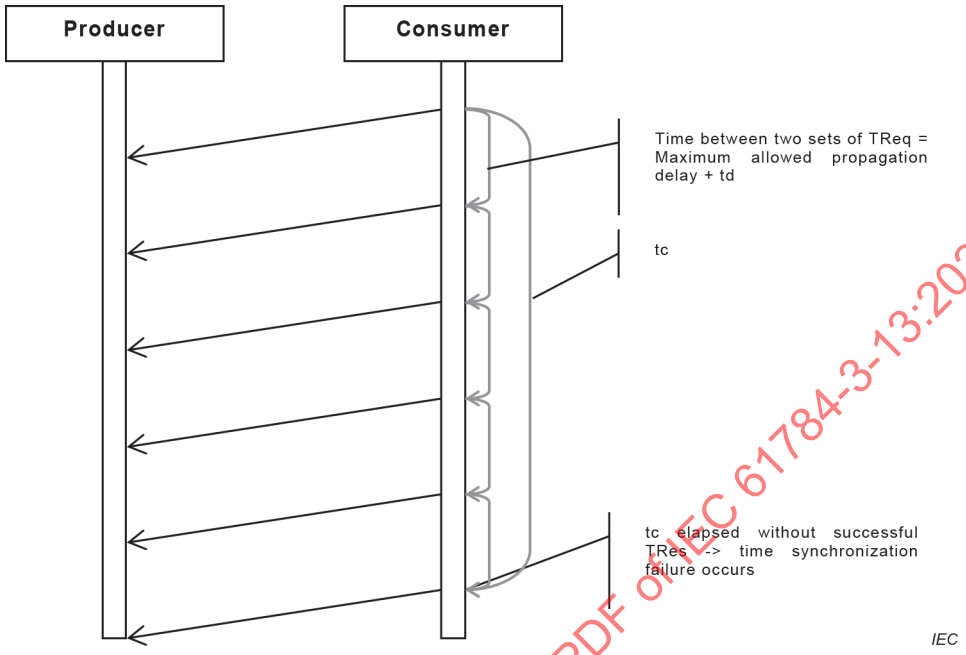


Key

See Table 229.

Figure 62 – Explanation of time synchronization

Figure 63 specifies a time synchronization failure. The consumer sends time requests to the producer. If a valid time response is not received within the entire time synchronization cycle time, a synchronization failure shall occur. This also ensures that the related outputs will enter the safe state.



Key

See Table 229.

Figure 63 – Time synchronization failure

Table 229 specifies items used in Figure 61, Figure 62 and Figure 63.

Table 229 – Extended time synchronization item description

Item	Description	Min value	Max value	SOD
n	Number of TRes from Producer	1	255	Object index 1C00h – 1FFEh sub index 03h (see 7.5.4.18)
TR	Time Request Distinctive Number Consumer: To be incremented each TReq	0	63	-
m	Number of consecutive TReq from consumer	1	63	Object index 1400h – 17FEh sub index 03h (see 7.5.4.16)
td	Time delay between two TReq blocks [CT]	0	UNSIGNED32	Object index 1400h – 17FEh sub index 04h (see 7.5.4.16)
ts	Time delay for new synchronization [CT]. The delay between successful time synchronization and the next TReq	1	UNSIGNED32	Object index 1400h – 17FEh sub index 05h (see 7.5.4.16)
tc	Time request cycle [CT]	1	UNSIGNED32	Object index 1400h – 17FEh sub index 0Bh (see 7.5.4.16)
maxpd	Maximum allowed TSync propagation delay	1	65535	Object index 1400h – 17FEh sub index 07h (see 7.5.4.16)

7.7.2.5 Time synchronization frequency

The frequency of the time synchronization is calculated by Formula (5).

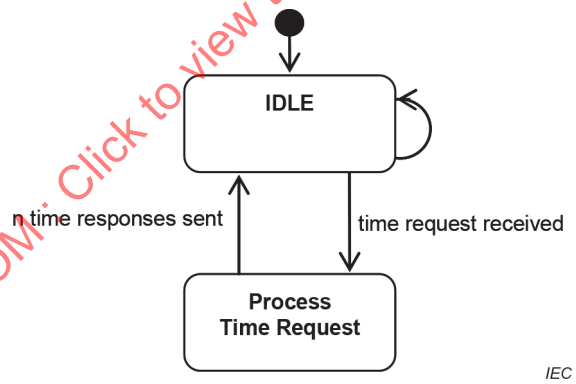
$$\text{TimeRequestCycle}[CT] = \frac{\text{SCT}[CT] - \text{Max. allowed SPDO PropagationDelay}[CT]}{\text{TimeAccuracy}[1]} \quad (5)$$

where

Element	Min value	Max value	SOD
Time request cycle [CT]	1	UNSIGNED32	Object index 1400h – 17FEh sub index Bh (see 7.5.4.16)
SCT – Safety control time [CT]	1	UNSIGNED32	Object 1400h – 17FEh sub index 2h (see 7.5.4.16)
Max. allowed SPDO propagation delay [CT]	1	65 535	Object index 1400h – 17FEh sub index 9h (see 7.5.4.16)
TimeAccuracy: Overall Time Accuracy of Consumer and Producer. This term is mainly determined by the system, the quartz crystal, etc.	-	-	-

7.7.2.6 Time synchronization producer

The time synchronization producer state diagram is specified by Figure 64.



Key

See Table 230 and Table 231.

Figure 64 – State diagram time synchronization producer

Table 230 specifies items used in Figure 64.

Table 230 – Time synchronization producer item description

Item	Description	Min value	Max value	SOD
n	Number of TRes from producer	1	255	Object index 1C00h – 1FFEh sub index 03h (see 7.5.4.18)

Table 231 specifies the time synchronization producer states.

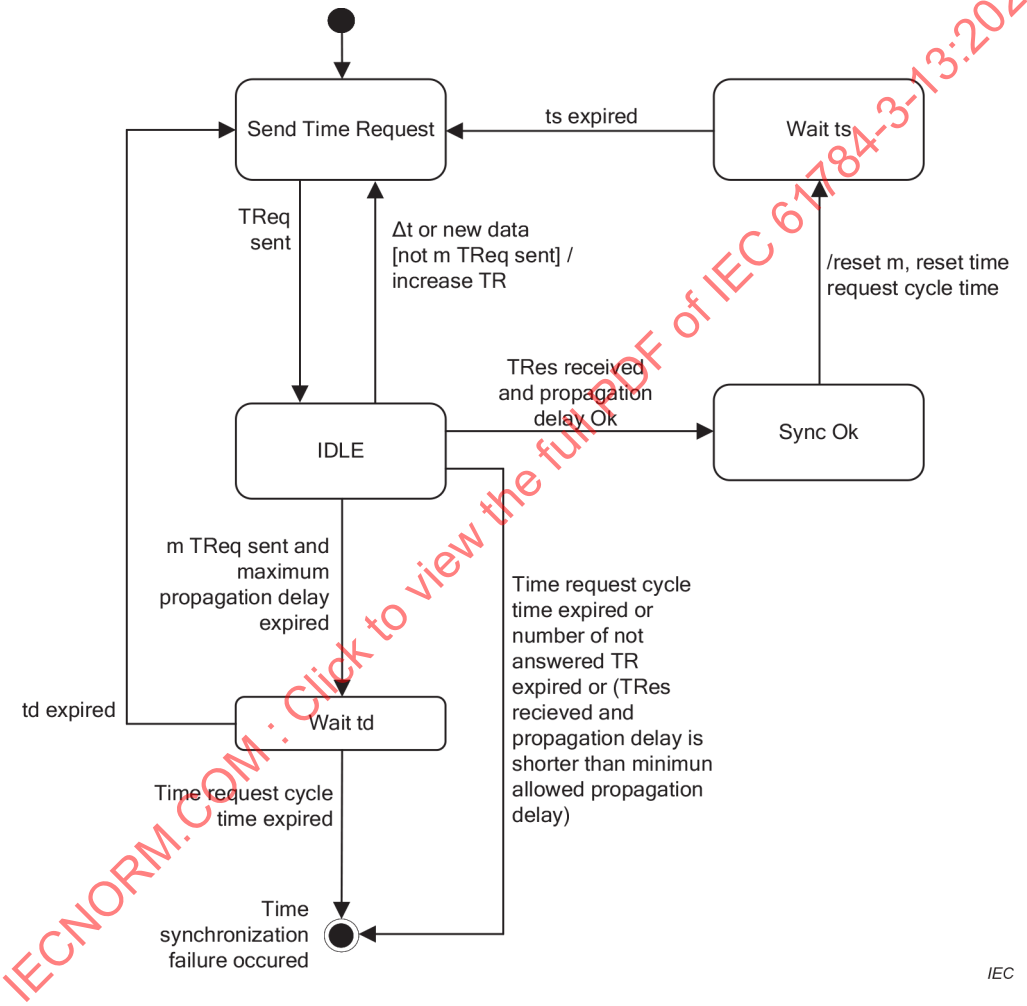
Table 231 – Time synchronization producer state description

State	Description
IDLE	The producer waits for a time request from any consumer
Process Time Request	After a time request from a consumer, the producer starts answering this request

NOTE The response to a Time Request is not a separate telegram. It is just part of the normal TxSPDO.

7.7.2.7 Time synchronization consumer

The time synchronization consumer state diagram is specified by Figure 65.



Key

See Table 232 and Table 233.

Figure 65 – State diagram time synchronization consumer

Table 232 specifies items used in Figure 65.

Table 232 – Time synchronization consumer item description

Item	Description	Min value	Max value	SOD
TReq	Time synchronization request	-	-	-
m	Number of consecutive TReq from consumer	1	63	Object index 1400h – 17FEh sub index 03h (see 7.5.4.16)
TR	Time Request Distinctive Number Field	0	63	-
Δt	TxSPDO refresh prescale [CT]	1	32 767	Object index 1400h – 17FEh sub index 02h (see 7.5.4.16)
td	Time delay between two TReq blocks [CT]	0	UNSIGNED32	Object index 1400h – 17FEh sub index 04h (see 7.5.4.16)
ts	Time delay for new synchronization [CT]. The delay between successful time synchronization and the next TReq	1	UNSIGNED32	Object index 1400h – 17FEh sub index 05h (see 7.5.4.16)
minpd	Minimum allowed propagation delay for time synchronization [CT]	1	65 535	Object index 1400h – 17FEh sub index 06h (see 7.5.4.16)
maxpd	Maximum allowed propagation delay for time synchronization [CT]	1	65 535	Object index 1400h – 17FEh sub index 07h (see 7.5.4.16)
tc	Time request cycle [CT]	1	UNSIGNED32	Object index 1400h – 17FEh sub index 0Bh (see 7.5.4.16)

Table 233 specifies the time synchronization consumer states.

Table 233 – Time synchronization consumer state description

State	Description
Send Time Request	Sending the time request to the corresponding producer
IDLE	Waiting for TRes
Wait td	Waiting until td expired. All received TRes are ignored
Wait ts	Waiting until ts expired. All received TRes are ignored
Sync Ok	Synchronization Ok. Reset m: reset time request cycle time. Reset number of not answered TR

7.7.2.8 Dynamic SCT calculation

A consumer may calculate the value for the SCT dynamically, to increase timing independence for the safety data transport, as well as increase availability by better utilizing the timing periods and SPDO transfer windows.

The time request/response cycle is used as measurement for the black channel to determine, if the underlying black channel allows for safe SPDO channel communication within the predefined maximum safety response time.

A telegram transport regarding time synchronization allways includes both the request from the consumer to the producer as well as the response from the producer to the consumer.

If the minimum telegram transit time is set to 0 or not set at all, it is assumed that telegram transport for the request between consumer and producer is imminent, and all measured times apply only to the response telegram transport between producer and consumer.

The same configuration object (see 7.5.4.16) is used for the dynamic SCT calculation as for the non-dynamic SCT calculation.

If an SN utilizing the dynamic SCT calculations is provided with a complete set of information, it shall calculate the safe reaction time value for the RxSPDO connection and process data validity by applying Formula (6).

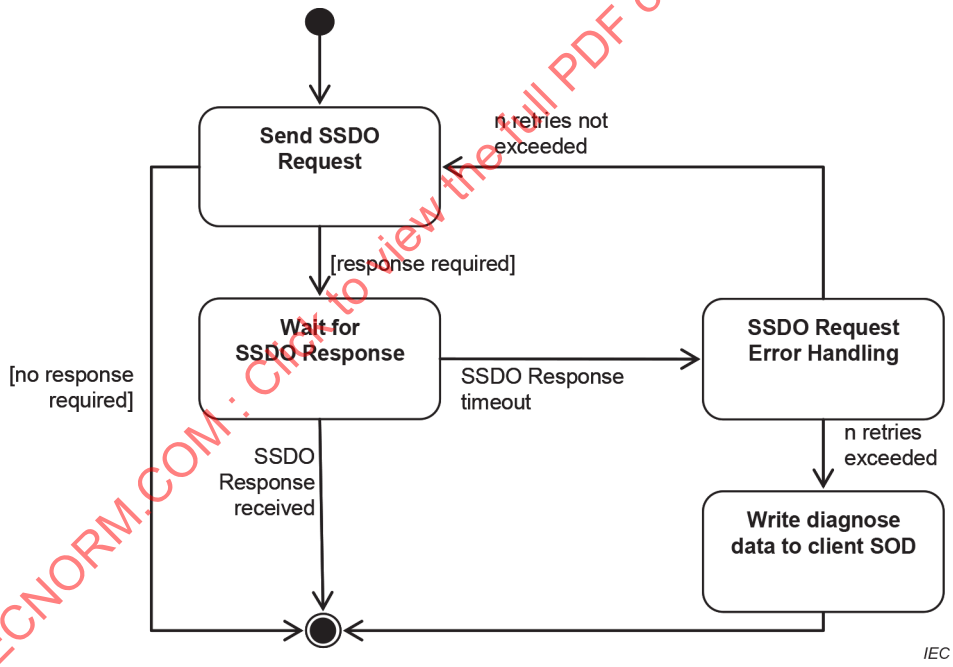
$$\text{SafeReactionTime} = \text{SCT\_U32} + \text{MaxTSyncPropagationDelay\_U16} + \text{BestCastTResDelay\_U16} \quad (6)$$

7.7.3 Safety Service Data Object (SSDO)

7.7.3.1 SSDO client

The client shall send an SSDO request telegram to the server. If a response is received or a response is not required, the client shall continue (more SSDO telegrams, etc.). If the response is timed out, the client shall retry to send the telegram n times. The number of retries as well as the timeout time for an SSDO Response telegram shall be set within the SOD of the client.

The SSDO client state diagram is specified by Figure 66.



Key

See Table 234 and Table 235.

Figure 66 – State diagram SSDO client

Table 234 specifies items used in Figure 66.

Table 234 – SSDO client item description

Item	Description	Min value	Max value	SOD
n	Maximum count of retries	0	UNSIGNED8	Object index 1201h sub index 02h (see 7.5.4.14)
	Timeout of SSDO Response [CT]	1	UNSIGNED32	Object index 1201h sub index 01h (see 7.5.4.16)

Table 235 specifies the SSDO client states.

Table 235 – SSDO client state description

State	Description
Send SSDO Request	Sending the SSDO telegram to the server
Wait for SSDO Response	Waiting for the response from the server
SSDO Request Error Handling	If the maximum number of retries was not reached, the telegram will be sent again otherwise a communication error occurs
Write diagnose data to client SOD	The diagnose data has to be available for the user. This is made through the SOD of the client

7.7.3.2 SSDO server

If the server receives an SSDO request telegram from the client, the telegram shall be processed according to the ID field. During processing, an SSDO response shall be generated and sent back to the client.

The SSDO server state diagram is specified by Figure 67.

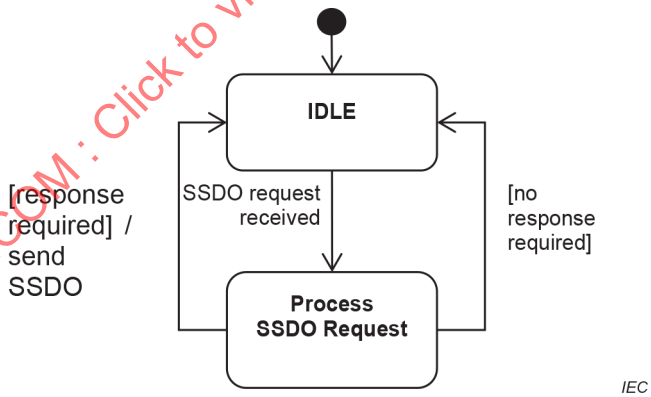


Figure 67 – State diagram SSDO server

Table 236 specifies the SSDO server states.

Table 236 – SSDO server state description

State	Description
IDLE	Waiting for SSDO Access Request telegrams
Process SSDO Request	Process the telegram according to the SSDO Access Command. Generate SSDO Response

7.7.4 SOD access

7.7.4.1 General

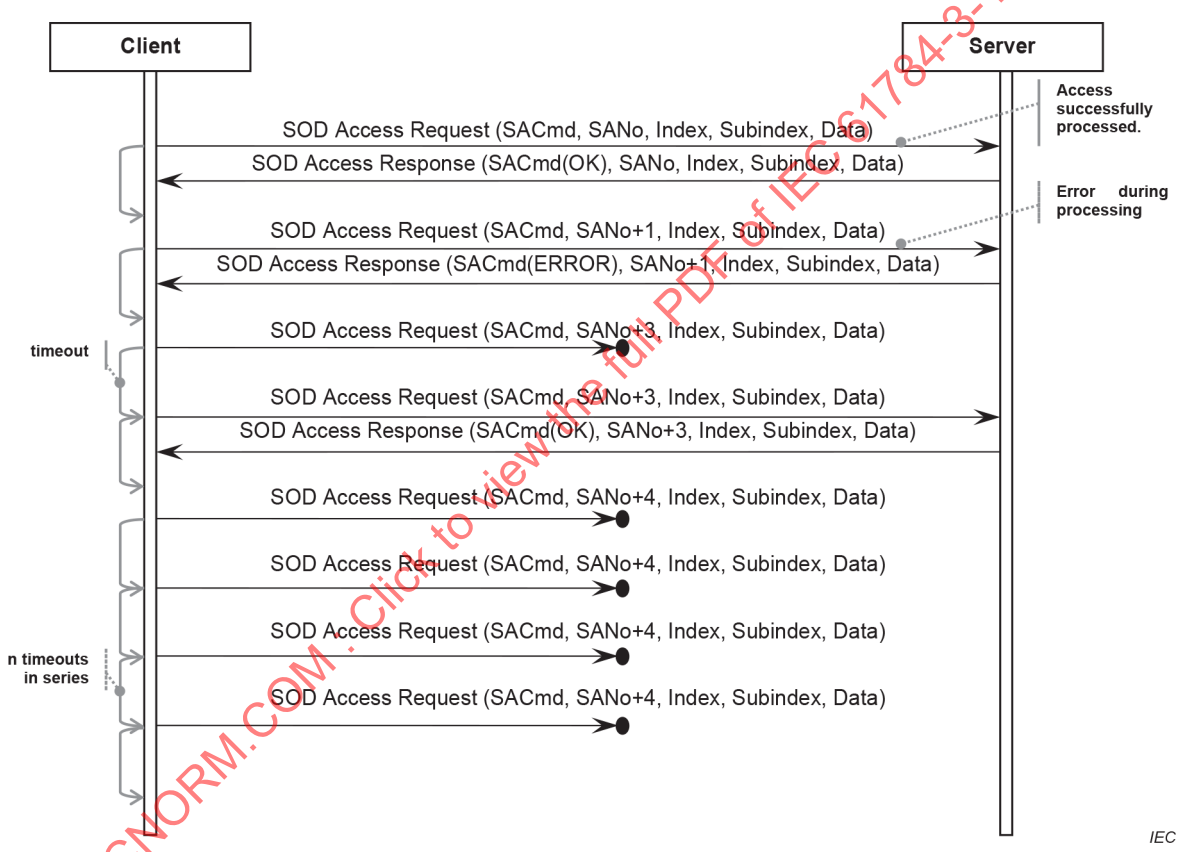
The SOD Access shall be based on the SSDO services. The SSDO service shall be used by the client to access the SOD (Safety Object Dictionary) within the server.

Write access shall be permitted in

- in Pre-Operational state for all parameters,
- in Operational state for user parameters within 2800h – 2FFFh index range only. Slim SSDO Services shall not be used to access these parameters.

7.7.4.2 SOD access (expedited)

The SOD access (expedited) sequence diagram is specified by Figure 68.



Key

See Table 237.

Figure 68 – Expedited SOD access

Table 237 specifies items used in Figure 68.

Table 237 – SOD access item description

Item	Description	Min value	Max value	SOD
n	Max. count of retries in series	0	UNSIGNED8	Object index 1201h sub index 02h (see 7.5.4.14)
Timeout	Timeout of SOD Access response	1	UNSIGNED32	Object index 1201h sub index 01h (see 7.5.4.14)

7.7.4.3 SOD download access with segmentation

7.7.4.3.1 Client state diagram

Segmented SOD Access shall be used to download large amounts of data to the SN (e.g. the entire SOD during parameterization). It shall consist of the initialization telegram, the segment download telegram and the end telegram. Each telegram requires a special response. The initialization telegram shall carry the size of the data to be downloaded as payload. The segment download telegrams shall carry data only. The end telegram shall carry the last data segment as payload. If one telegram fails, the entire download shall be repeated.

The segmented SOD download access client state diagram is specified by Figure 69.

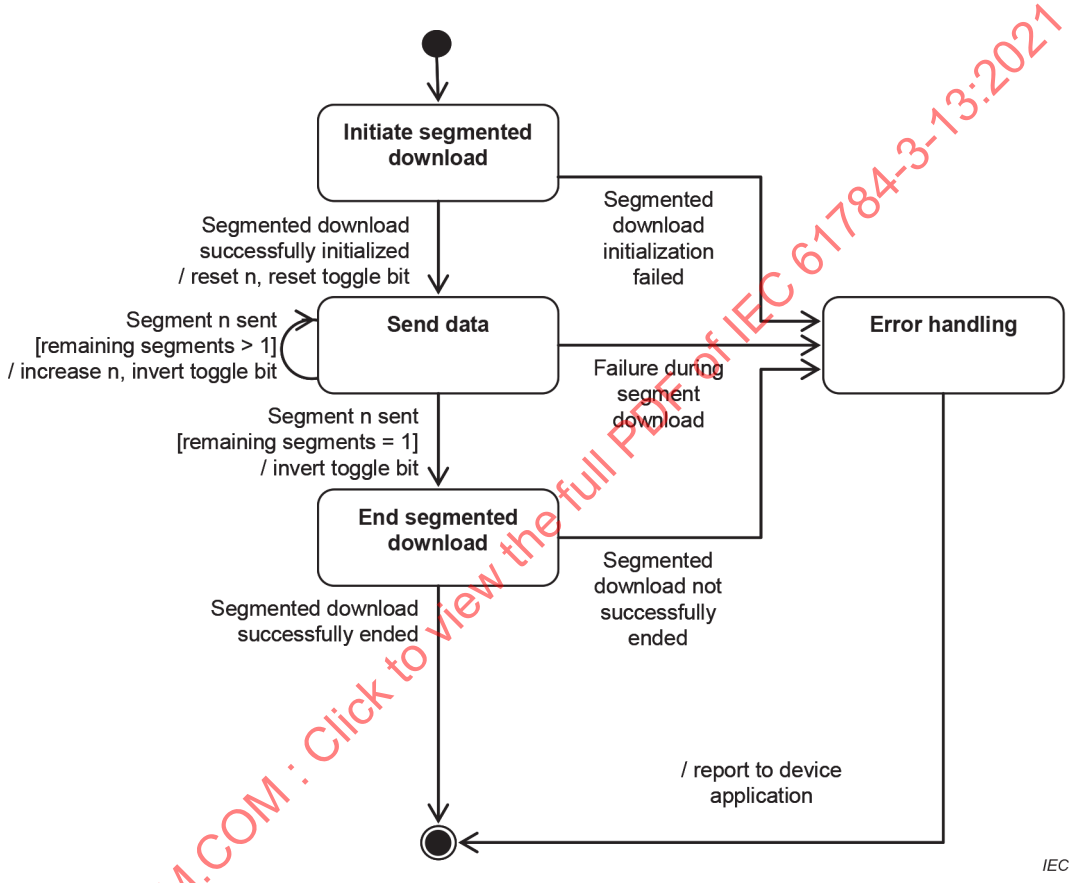
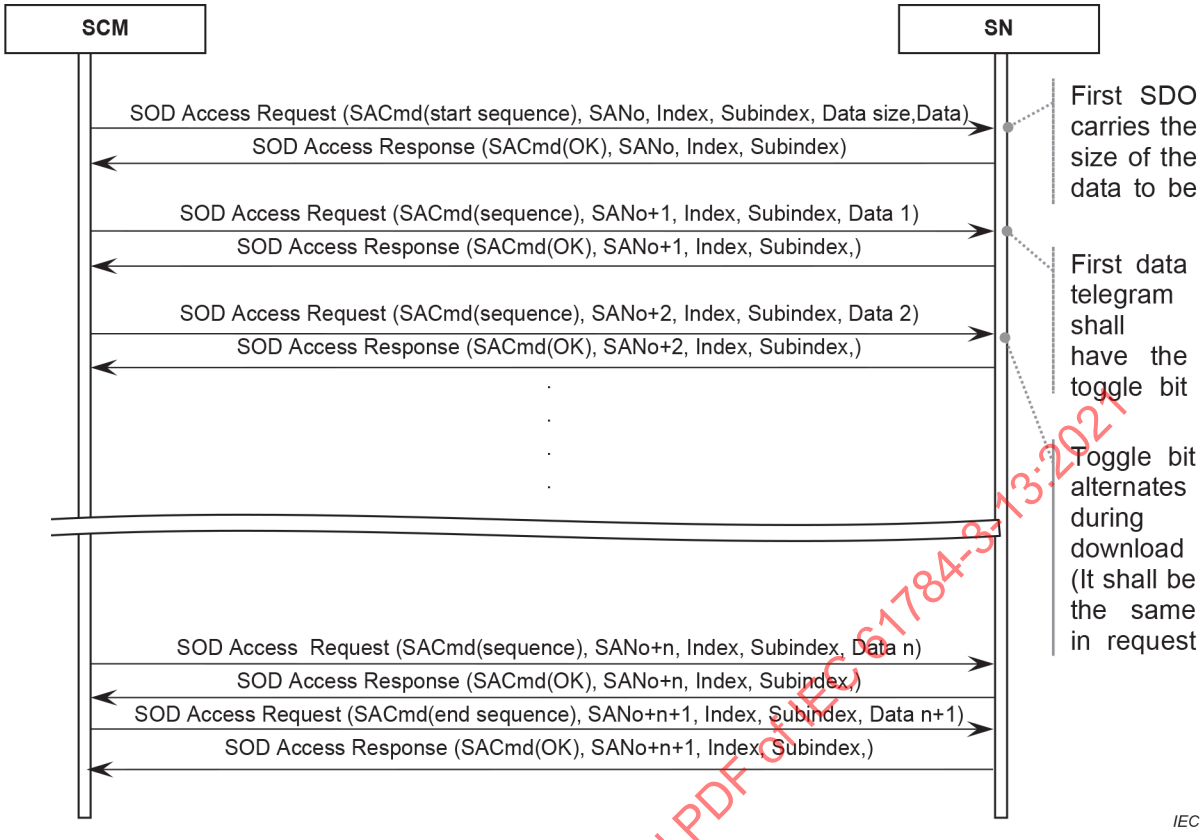


Figure 69 – State diagram segmented SOD download access client

The segmented SOD download access sequence diagram is specified by Figure 70.



Key

See Table 238.

Figure 70 – Segmented SOD download access

Table 238 specifies items used in Figure 70.

Table 238 – Segmented SOD access client item description

Item	Description	Min value	Max value	SOD
n	current segment number	-	-	-
Data size	4 octet value which identifies the size of data being downloaded with the segmented SOD Access	1	UNSIGNED32	-

Table 239 specifies the segmented SOD download access client states.

Table 239 – Segmented SOD download access client state description

State	Description
Initiate sequenced download	Starts the SOD Access with the initialization telegram
Send data	Starts the SOD Access with a segment download telegram
End sequenced download	Starts the SOD Access with the end download telegram
Error handling	Reports the communication error to the application

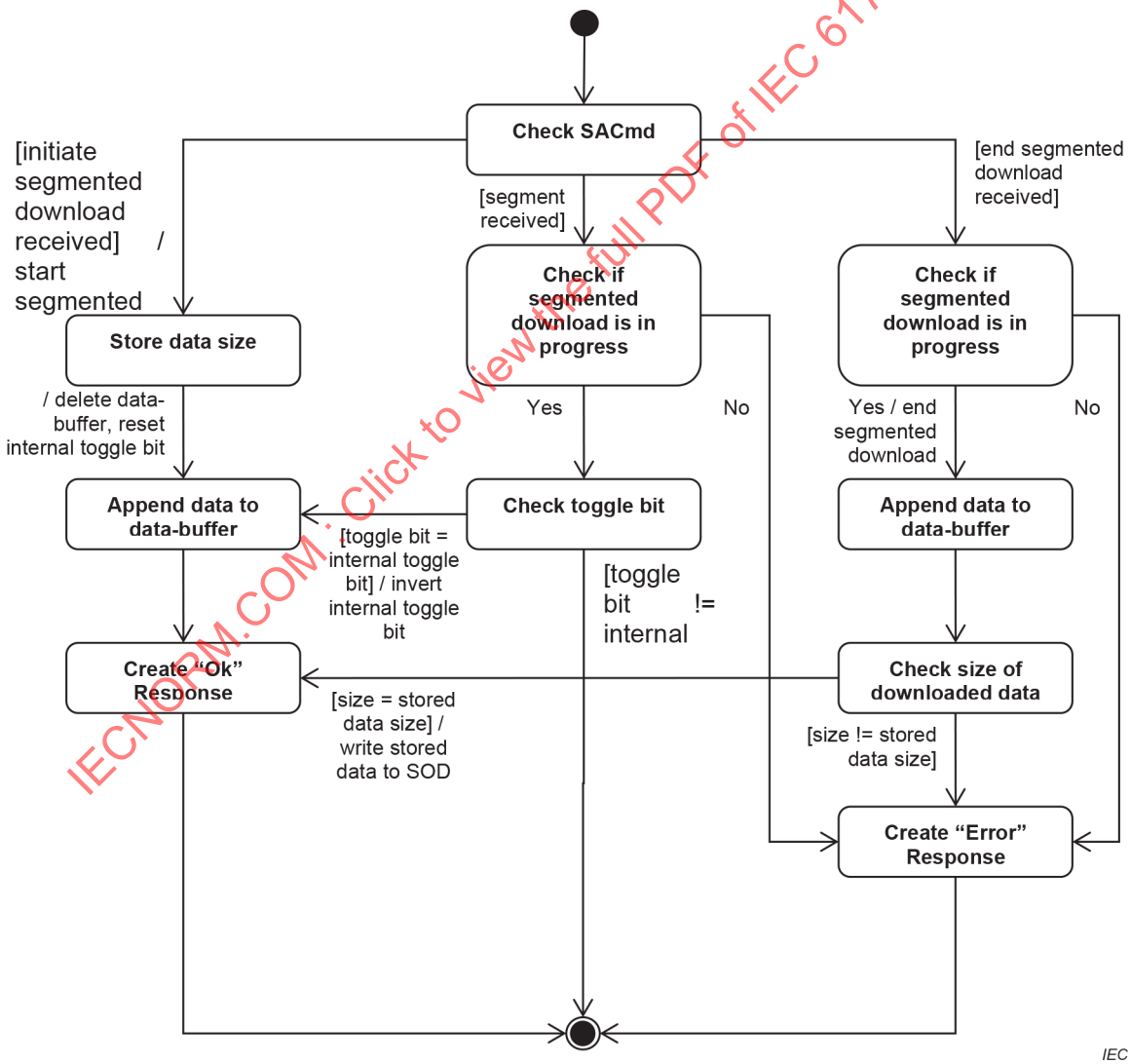
7.7.4.3.2 Server state diagram

The server which receives the data via the segmented download shall verify the correctness of the telegrams. If a download is initiated, the server shall store the data size which shall be received during the entire download.

When the download is complete, this data size shall be cross-checked with the size of the received data. If the received data is not equal to the expected data size, the server shall respond with an error telegram. If the server receives "segment" or "end segmented download" telegrams, it also shall check if the download was initiated. Otherwise, the data shall be ignored and an error response shall be generated.

During the segmented download, the toggle bit of the "segment" telegrams shall be inverted for each new telegram. If the server receives two or more consecutive telegrams with the same toggle bit, only the data of the first telegram shall be stored, all other telegrams shall be ignored.

The Segmented SOD Download Access Server state diagram is specified by Figure 71.



Key

See Table 240 and Table 241.

Figure 71 – State diagram segmented SOD download access server

Table 240 specifies items used in Figure 71.

Table 240 – Segmented SOD access server item description

Item	Description	Min value	Max value	SOD
toggle bit	The toggle bit is used to identify new segments	0	1	-
data size	4 octet value which identifies the size of data being downloaded with the segmented SOD Access	1	UNSIGNED32	-
size	Size of the received data	0	UNSIGNED32	-
SACmd	Safety Access Command	-	-	-

Table 241 specifies the segmented SOD access server states.

Table 241 – Segmented SOD access server state description

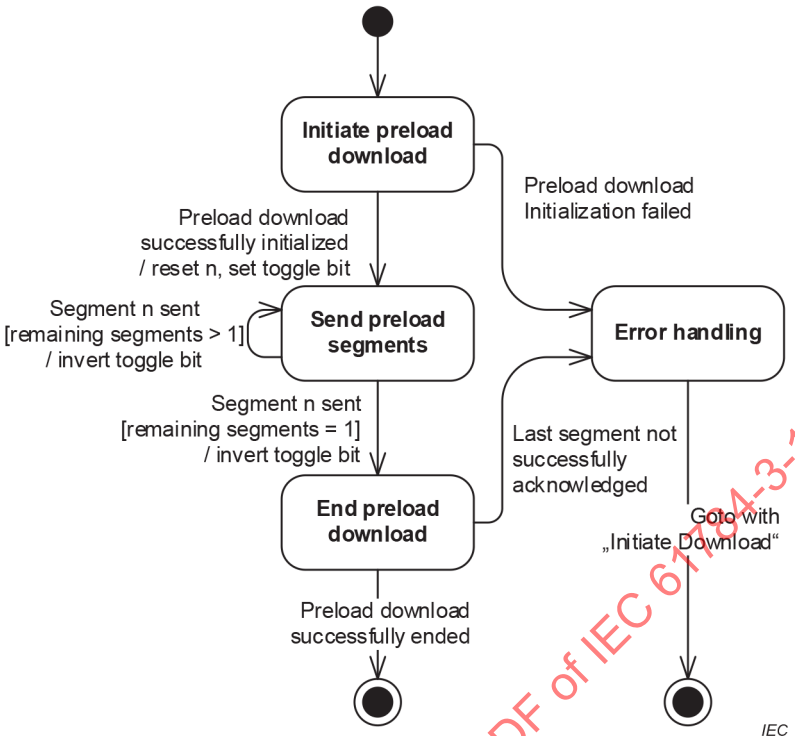
State	Description
Check SACmd	Checks which type of telegram was received
Store data size	Stores the expected data size for cross-check
Append data to data-buffer	Writes the received data into a temporary buffer
Create "Ok" Response	If the telegram was valid, the server sends a response to the client which indicates that the telegram was processed
Check if segmented download is in progress	If a "segment" or "end segmented download" telegram was received, the server has to check if the segmented download was initialized
Check toggle bit	Each "segment" telegram carries a toggle bit which has to be inverted for each new telegram
Check size of downloaded data	At the end of the segmented download, the server checks if the expected data size is equal to the received data size. If not, an error response is generated
Create "Error" response	Creates an error response which tells the client that the segmented download failed

7.7.4.4 SOD Download Access with Preload

7.7.4.4.1 Client state diagram

Preload SOD Access may be used to download a large amount of data to the SN (e.g. the entire SOD during parameterization) without the need of a immidiate confirmation for each segment. It shall consist of the initialization telegram, the preload download telegram and the end telegram. The initialization telegram shall carry the size of the data queue the server uses for storing telegrams before they are processed. If one telegram fails, the client shall reinitiate transmission with the next telegram which has not yet been acknowledged.

The SOD preload download access client state diagram is specified in Figure 72.

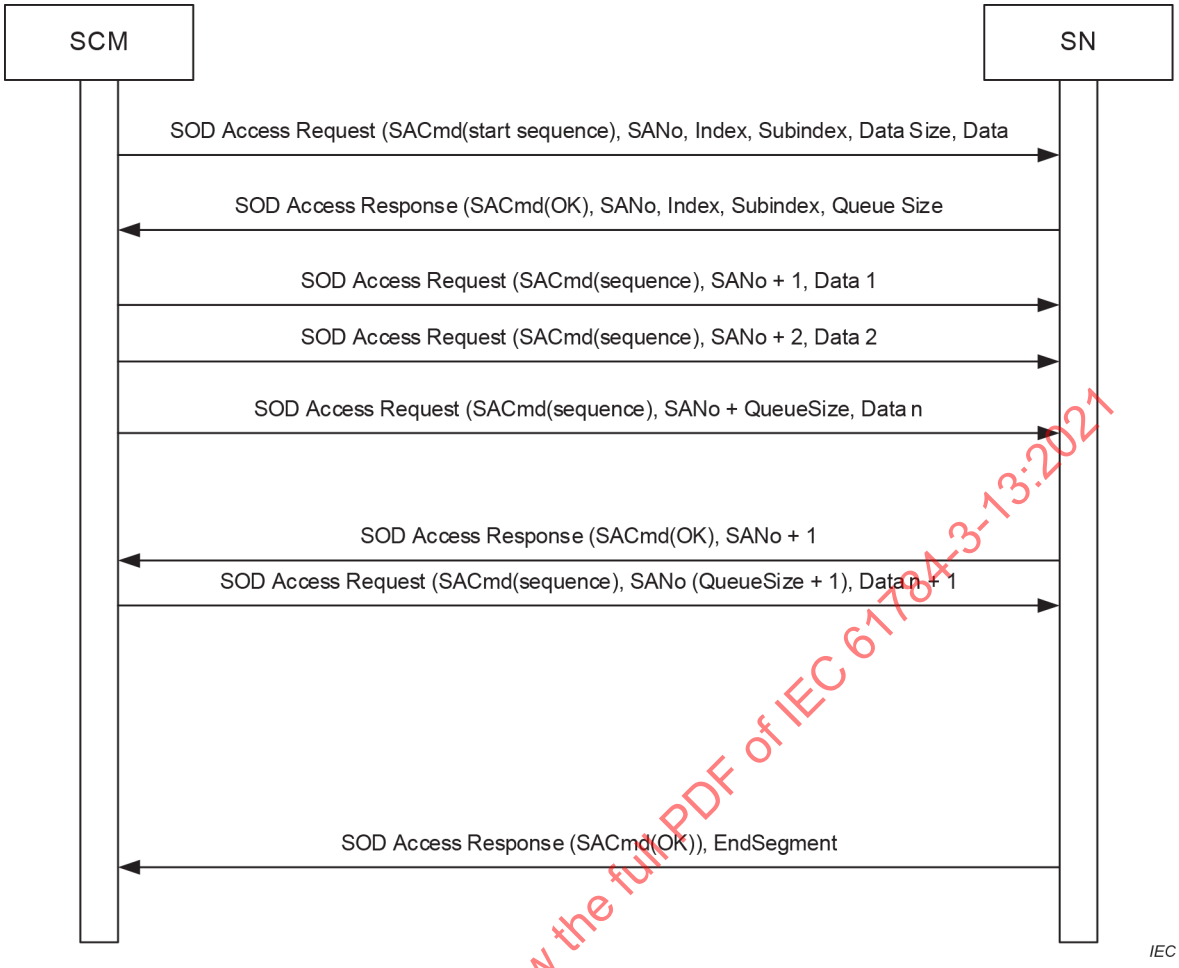


Key

See Table 242 and Table 243.

Figure 72 – State diagram SOD preload download access client

The SOD preload download access sequence diagram is specified in Figure 73.



Key

See Table 242.

Figure 73 – SOD preload download access

Table 242 specifies items used in Figure 73.

Table 242 – SOD preload download access client item description

Item	Description	Min value	Max value
n	Current segment number	–	–
Queue size	Size of queue on server for maximum number of queueable objects	0	15
Data size	4 octet value which identifies the size of data being downloaded with the segmented SOD access	1	UNSIGNED32

Table 243 specifies the SOD preload download access client states.

Table 243 – SOD preload download access client state description

State	Description
Initiate preload download	Starts the SOD access with the initialization telegram
Send preload download	Starts the SOD access with a preload download telegram
End preload download	Starts the SOD access with the end download telegram
Error handling	Reports the communication error to the application

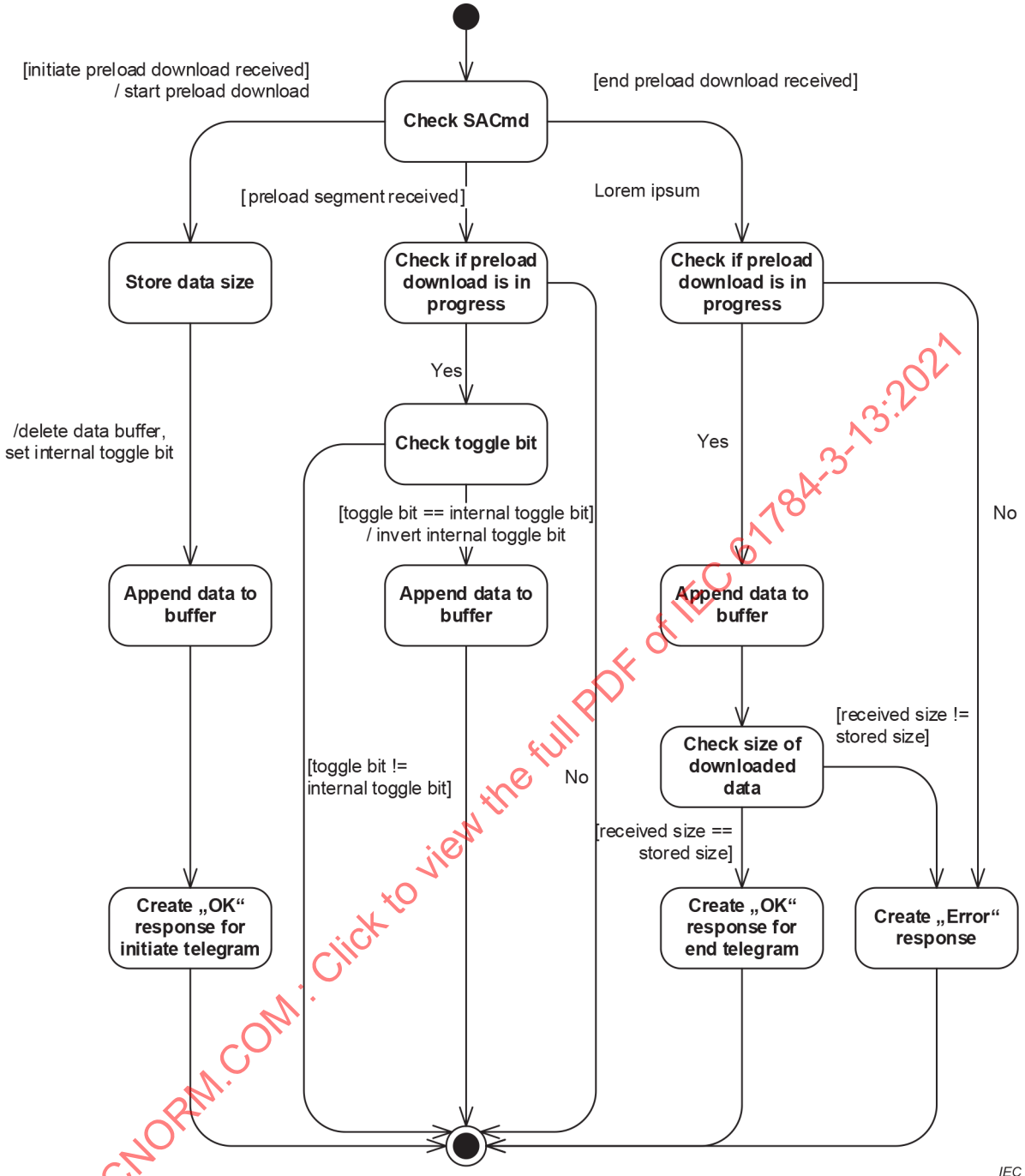
7.7.4.4.2 Server state diagram

The server which receives the data via the preload download shall verify the correctness of the telegrams. If a download is initiated, the server shall store the data size which shall be received during the entire download.

When the download is complete this data size shall be cross-checked with the size of the received data. If the received data is not equal to the expected data size, the server shall respond with an error telegram. If the server receives "segment" or "end segmented download" telegrams, it also shall check if the download was initiated. Otherwise the data shall be ignored and an error response shall be generated.

During the segmented download, the toggle bit of the "segment" telegrams shall be inverted for each new telegram. If the server receives two or more consecutive telegrams with the same toggle bit, only the data of the first telegram shall be stored, the other telegram shall be ignored.

The SOD preload download access server state diagram is specified in Figure 74.



Key

See Table 244 and Table 245.

Figure 74 – State diagram SOD preload download access server

Table 244 and Table 245 specify the SOD preload download access server items and states.

Table 244 – SOD preload download access server item description

Item	Description	Min value	Max value	SOD
toggle bit	The toggle bit is used to identify new segments	0	1	–
data size	4 octet value which identifies the size of data being downloaded with the segmented SOD Access	1	UNSIGNED32	–
size	Size of the received data	0	UNSIGNED32	–
SACmd	Safety Access Command	–	–	–

Table 245 – SOD preload download access server state description

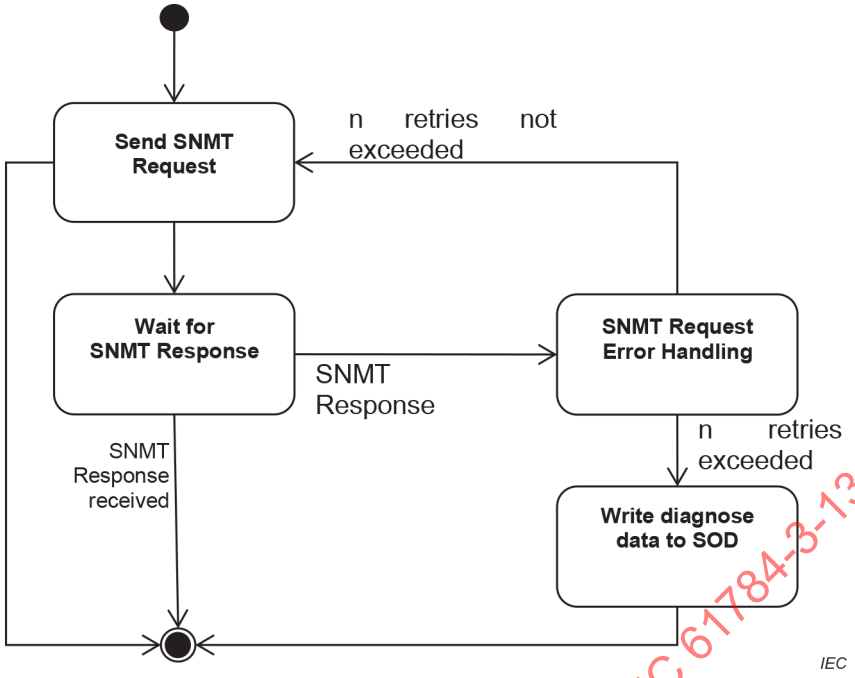
State	Description
Check SACmd	Checks which type of telegram was received
Store data size	Stores the expected data size for cross-check
Append data to data-buffer	Writes the received data into temporary buffer
Create "Ok" response for initiate telegram	If the telegram was valid, the server shall send a response to the client which indicates that the telegram was processed.
Create "Ok" response for end telegram	If the telegram was valid, the server shall send a response to the client which indicates that the telegram was processed.
Check if preload download is in progress	If a "preload" or "end preload download" was received, the server shall check if the preload download was initialized.
Check toggle bit	Each "preload" telegram carries a toggle bit which shall be inverted for each new telegram.
Check size of downloaded data	At the end of the preload download, the server checks the expected data size. If it is not equal to the received data size, an error response shall be generated.
Create "error" response	Creates an error response which tells the client that the preload download failed.

7.7.5 Safety Network Management Object (SNMT)

7.7.5.1 SNMT master

The master shall send an SNMT Request telegram to the slave. If a response is received or a response is not required, the client may continue (more SNMT telegrams, etc.). If the response times out, the client shall retry to send the telegram n times. The number of retries as well as the timeout time for an SNMT Response telegram shall be set within the SOD of the client.

The SNMT master state diagram is specified by Figure 75.



Key

See Table 246 and Table 247.

Figure 75 – State diagram SNMT master

Table 246 specifies items used in Figure 75.

Table 246 – SNMT master item description

Item	Description	Min value	Max value	SOD
n	Max. count of retries	0	UNSIGNED8	Object index 1202h sub index 02h (see 7.5.4.15)
	Timeout of SNMT Response [CT]	1	UNSIGNED32	Object index 1202h sub index 01h (see 7.5.4.15)

Table 247 specifies the SNMT master server states.

Table 247 – SNMT master state description

State	Description
Send SNMT Request	Sending the SNMT telegram to the slave
Wait for SNMT Response	Waiting for the response from the server
SNMT Request Error Handling	If the maximum number of retries was not reached the telegram will be sent again otherwise a communication error occurs
Write diagnose data to SOD	The diagnose data has to be available for the user. This is made through the SOD of the master

7.7.5.2 SNMT slave

The slave receives an SNMT request telegram from the master. The telegram shall be processed according to the ID field. During processing, an SNMT Response shall be generated and sent back to the master.

The SNMT slave state diagram is specified by Figure 76.

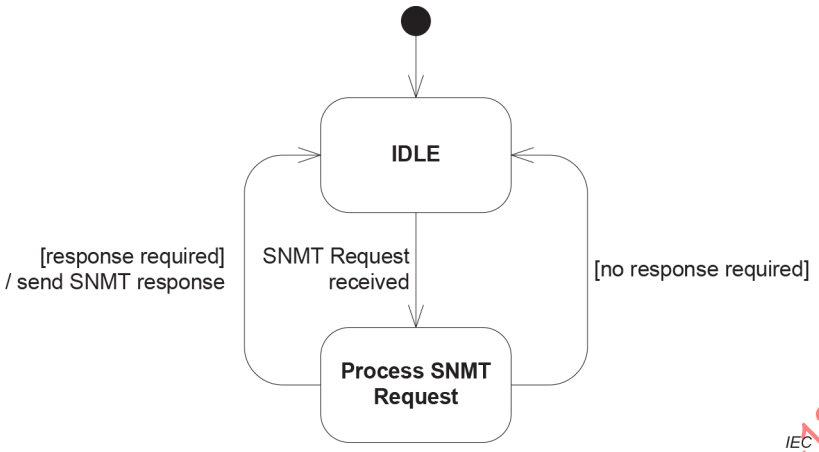


Figure 76 – State diagram SNMT slave

Table 248 specifies the SNMT slave states.

Table 248 – SNMT slave state description

State	Description
IDLE	Wait for SNMT Request telegrams
Process SNMT Request	Process the telegram according to the SNMT Request. Generates SNMT Response

7.7.6 SN power up

7.7.6.1 General

At power up, the SN shall start the self initialization. An SN may contain a flash memory where all data is stored. This data shall be loaded during the self initialization. After this, the SN shall enter the Pre-Operational state. In this state the SN may get parameters from the SCM. The SN also shall send cyclic messages to the SCM to inform the SCM about its status. When the SN switches from Pre-Operational to Operational, it shall store all parameters within the flash memory (this shall be an optional feature).

The SN power up state diagram is specified by Figure 77.

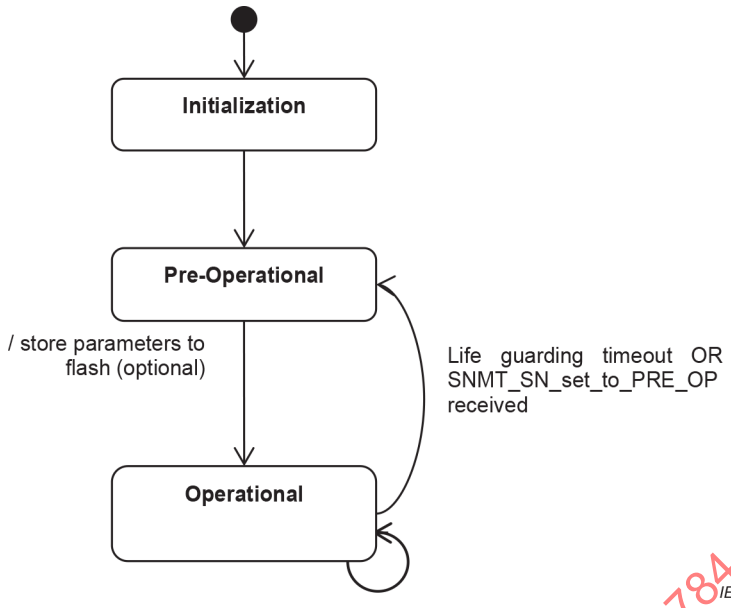


Figure 77 – State diagram SN power up

Table 249 specifies the SN power up states.

Table 249 – SN power up state description

State	Description
Initialization	Loading parameters from flash (optional)
Pre-Operational	Receiving parameters from SCM, UDID verification, etc. In this state all application data are set to the safe state (default of the object, see 7.5.2.10)
Operational	Sending and receiving SPDOs, time requests, etc. To keep the SN in Operational, it needs to receive a cyclic signal from the SCM. If the signal is timed out (timeout to be set within the SOD), the SN switches to Pre-Operational

The states Initialization, Pre-Operational and Operational are also referred to as communication states.

7.7.6.2 States and communication object relation

Table 250 specifies the relation between communication states and communication objects. Services on the listed communication objects shall only be executed if the device is in the appropriate communication state.

Table 250 – State and communication object relation

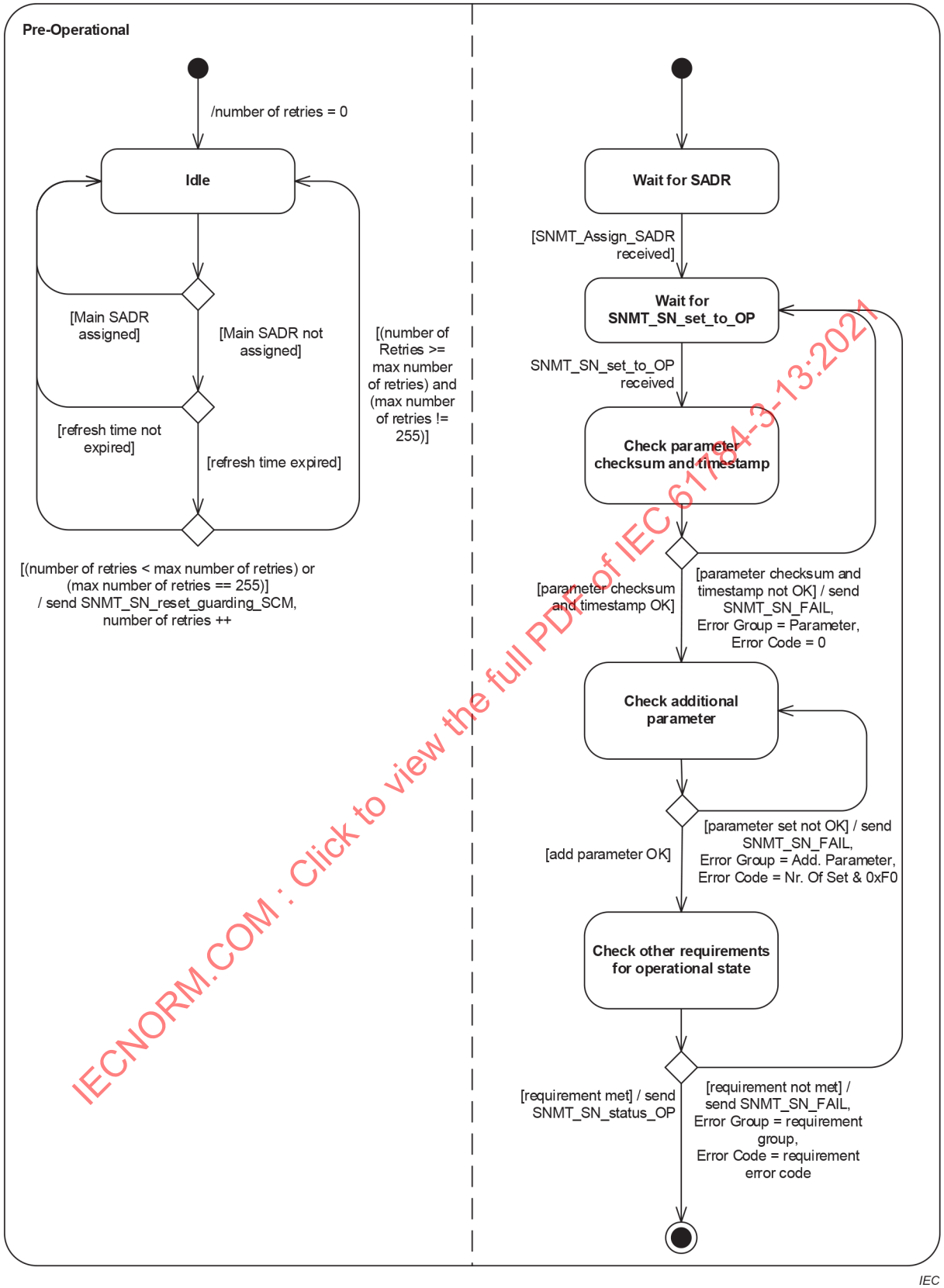
	Initialization	Pre-Operational	Operational
SPDO			X
SSDO		X	X
SNMT		X	X

7.7.6.3 Pre-Operational

7.7.6.3.1 General

When the SN enters the Pre-Operational state, the SNMT and SOD Access Rx services shall be started. In this state all application data shall be set to the safe state. The SNMT services are required for several purposes, e.g. receiving UDID request telegrams. With SOD Access Commands, the SN may get new parameters from the SCM. Before being able to switch to Operational, the SN at least shall receive its "Main SADR" and the "UDID of the SCM".

The SN Pre-Operational state diagram is specified by Figure 78.



Key

See Table 251 and Table 252.

Figure 78 – State diagram SN Pre-Operational

Table 251 specifies items used in Figure 78.

Table 251 – SN Pre-Operational state item description

Item	Description	Min value	Max value	SOD
Refresh Time	Refresh time of SNMT_SN_reset_guarding_SCM signal [µs]	1	UNSIGNED32	Object index 100Dh (see 7.5.4.7)
Max number of retries	Maximum number of retries for the SNMT_SN_reset_guarding_SCM service	–	UNSIGNED8	Object index 100Eh (see 7.5.4.8)

Table 252 specifies the SN Pre-Operational states.

Table 252 – SN Pre-Operational state description

State	Description
Idle	Receiving parameters from SCM. UDID verification. Etc.
Wait for SADR	The SN waits to get its Main SADR from the SCM
Wait for SNMT_SN_set_to_OP	The SN waits to be switched to Operational from the SCM
Check Parameter Checksum and Timestamp	Checks if the parameter checksum as well as the parameter timestamp match the local settings. In case the checksum calculation takes a long time, the SN may reply to the SCM using the service SNMT_SN_busy. The SCM then has to send the service SNMT_SN_set_to_OP again
Check additional parameters	Checks if the SN needs additional parameters. If this is the case, an SN shall respond to the SCM with an SNMT_SN_Fail and a corresponding error code. In case the check takes a long time, the SN may reply to the SCM using the service SNMT_SN_busy. The SCM then has to send the service SNMT_SN_set_to_OP again.
Check other requirements for Operational state	There might be several reasons for not being able to switch to Operational state. Within this state, the module firmware is requested if switching to Operational is allowed. If not the reason is reported to the SCM

7.7.6.3.2 Check Additional Parameters

An SN may request additional parameter sets during configuration.

NOTE The transfer method of additional parameters is similar to normal parameter download.

After the SCM and the SN have successfully verified the default parameter of the SN, the SCM shall trigger the SN to switch to operational using the signal SNMT_SN_set_to_OP.

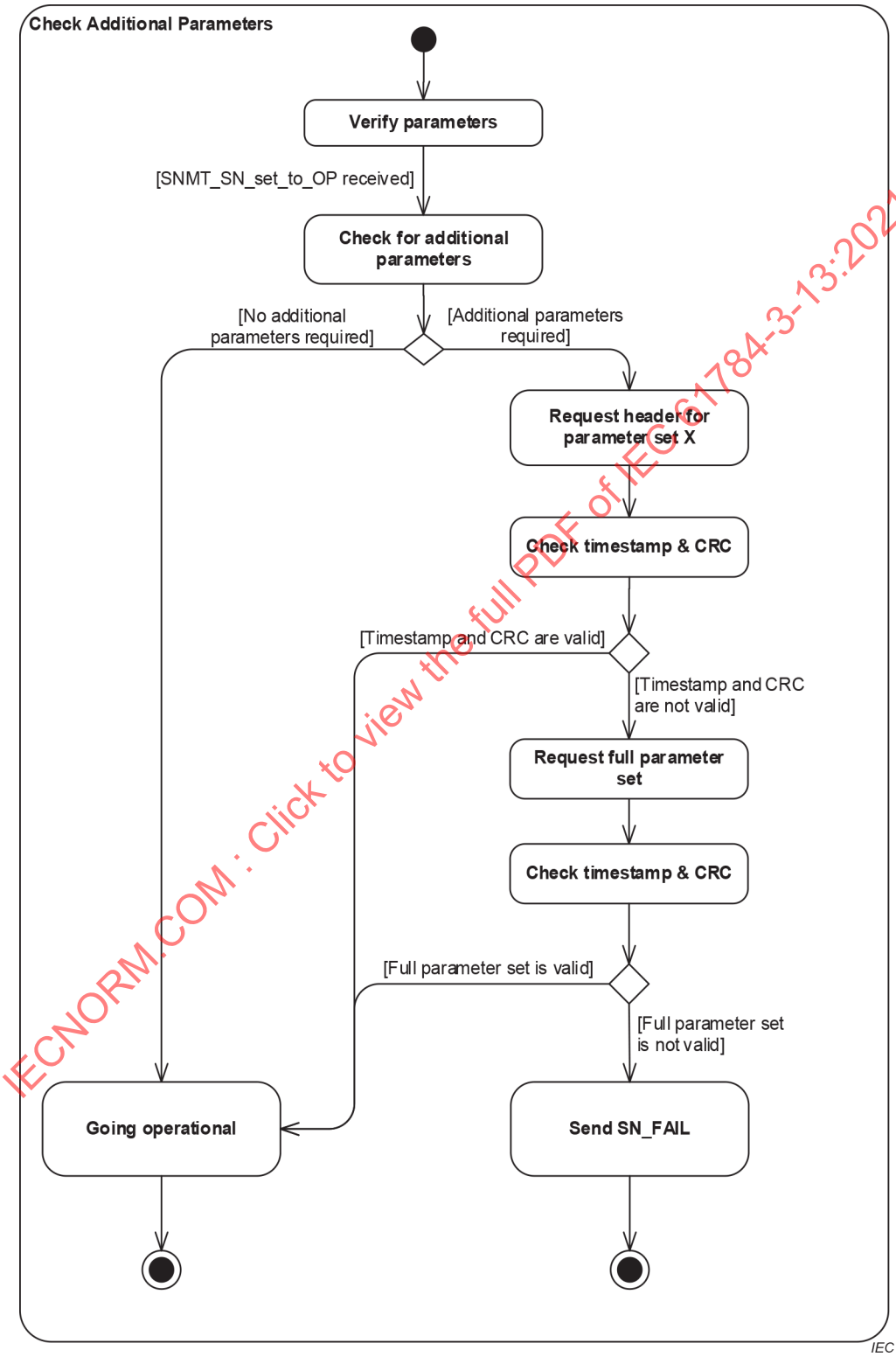
The SN chall check the need for one or more additional parameter sets to allow switching to operational. If additional parameter sets are required, the SN shall respond to the command for going operational by sending an SN_FAIL with an error group regarding additional parameters, and the number of the expected parameter set as error code. The SCM shall consult the corresponding set in the SOD entry E4XXh, where XX corresponds to the entry in C4XXh for the SN.

If the SCM does not have a corresponding entry in its SOD, it shall ask the SN to switch to operational again. If the SN receives a switch to operational, after it has requested an additional parameter set, but before it actually received such a set as download on 101Ah, it shall assume a configuration error on the side of the SCM and shall not switch to operational.

If the SCM has a corresponding parameter set in its E4XXh object structure, it shall send the parameter set header, as described in 7.5.4.11. The SN shall check, if received parameter set header is complete and if the timestamp and CRC match the stored timestamp and CRC on the device. The SCM's sole responsibility lies in the correct transmission using 0x101A. If the timestamp or CRC differ, the SN shall again respond by sending an SN_FAIL, this time asking for the complete set, which the SCM shall provide then.

If th SN requires further additional parameter sets the process may be repeated up to a maximum of 16 parameter sets.

The SN additional parameters state diagram is specified by Figure 79.



Key

See Table 253.

Figure 79 – State diagram SN additional paramters

Table 253 specifies the SN additional parameter states.

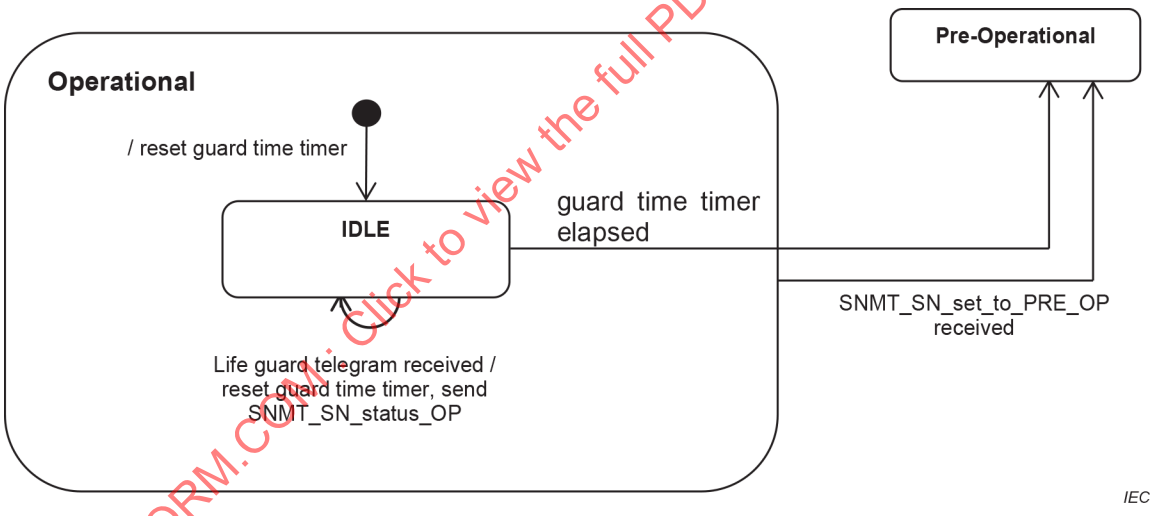
Table 253 – SN additional parameter state description

State	Description
Verifiy parameters	Standard parameters stored on the device are verified and the checksum is calculated
Check for additional parameters	If the device has received an indication or operates under the assumption, that additional parameters are needed, the information shall be processed solely on the device itself.
Request header for parameter set X	Request the header information for the provided parameter set, as stated in the error code
Check timestamp and CRC	Check the timestamp and CRC. The CRC algorithm is not specified
Request full parameter set	Request the full set, including again the set header

7.7.6.4 Operational

When the SN enters the Operational state, Life Guarding shall be started. The SN needs to receive cyclic telegrams from the SCM to remain in Operational state. If this telegram is timed out, the SN shall fall back into Pre-Operational state.

The SN Operational state diagram is specified by Figure 80.

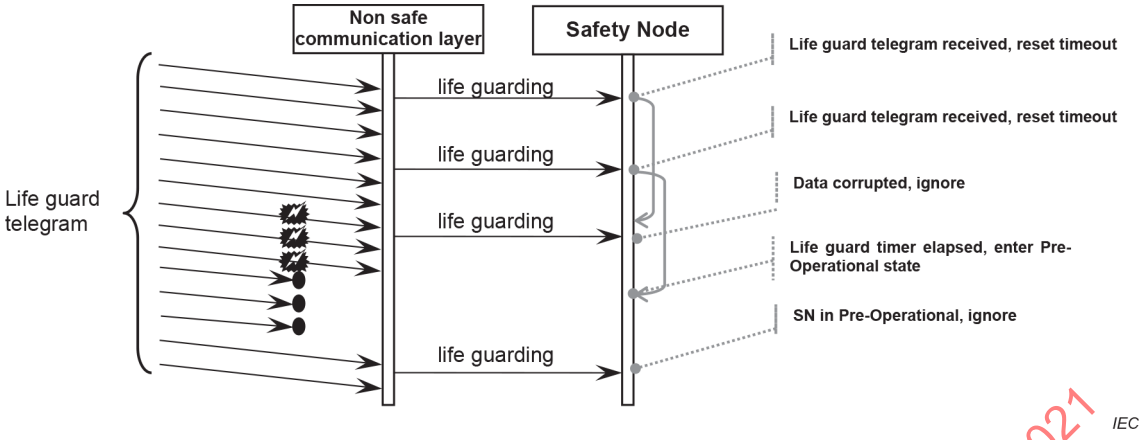


Key

See Table 254 and Table 255.

Figure 80 – State diagram SN Operational

The Life Guarding sequence diagram is specified by Figure 81.



Key

See Table 254.

Figure 81 – Life Guarding telegram

Table 254 specifies items used in Figure 80 and Figure 81.

Table 254 – SN Operational state item description

Item	Description	Min value	Max value	SOD
	Guard time timer	-	-	Object index 100Ch sub index 01h and 02h (see 7.5.4.6)

Table 255 specifies the SN Operational states.

Table 255 – SN Operational state description

State	Description
IDLE	Waits for Life Guarding telegram

7.7.7 SN power down

All safety outputs shall be set to safe state (0 V).

7.7.8 SN recovery after Restart / Error

If an error occurred or the SN was restarted, the SN shall enter the Pre-Operational state. In this state, the SN shall send a cyclic message (with no response) to the SCM. The SCM now shall check the SN again, verify all parameters, UDID, DVI and then set the SN to Operational again.

7.7.9 SCM power up

7.7.9.1 General

After power-on, the SCM loads its parameters from a non-volatile memory. Then it switches to Operational. The Operational super-state is described later on. The SCM can be switched into the Stopped state using an SNMT service. In Stopped, the SCM does not manage any safety nodes. If safety nodes are being configured with an external tool, the SCM shall be in state Stopped.

The SCM power up state diagram is specified by Figure 82.

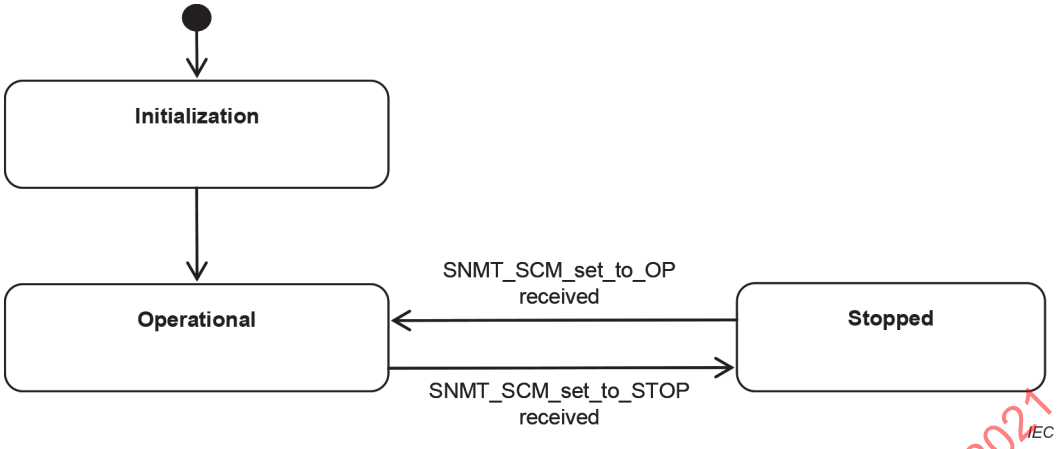


Figure 82 – State diagram SCM power up

Table 256 specifies the SCM power up states.

Table 256 – SCM power up state description

State	Description
Initialization	The SCM loads its parameters from a non-volatile memory
Operational	In Operational state all safety nodes are configured and started
Stopped	The SCM waits for the signal to switch back to Operational

7.7.9.2 States and communication object relation

Table 257 specifies the relationship between communication states and communication objects. Services on the listed communication objects shall only be executed if the device is in the appropriate communication state.

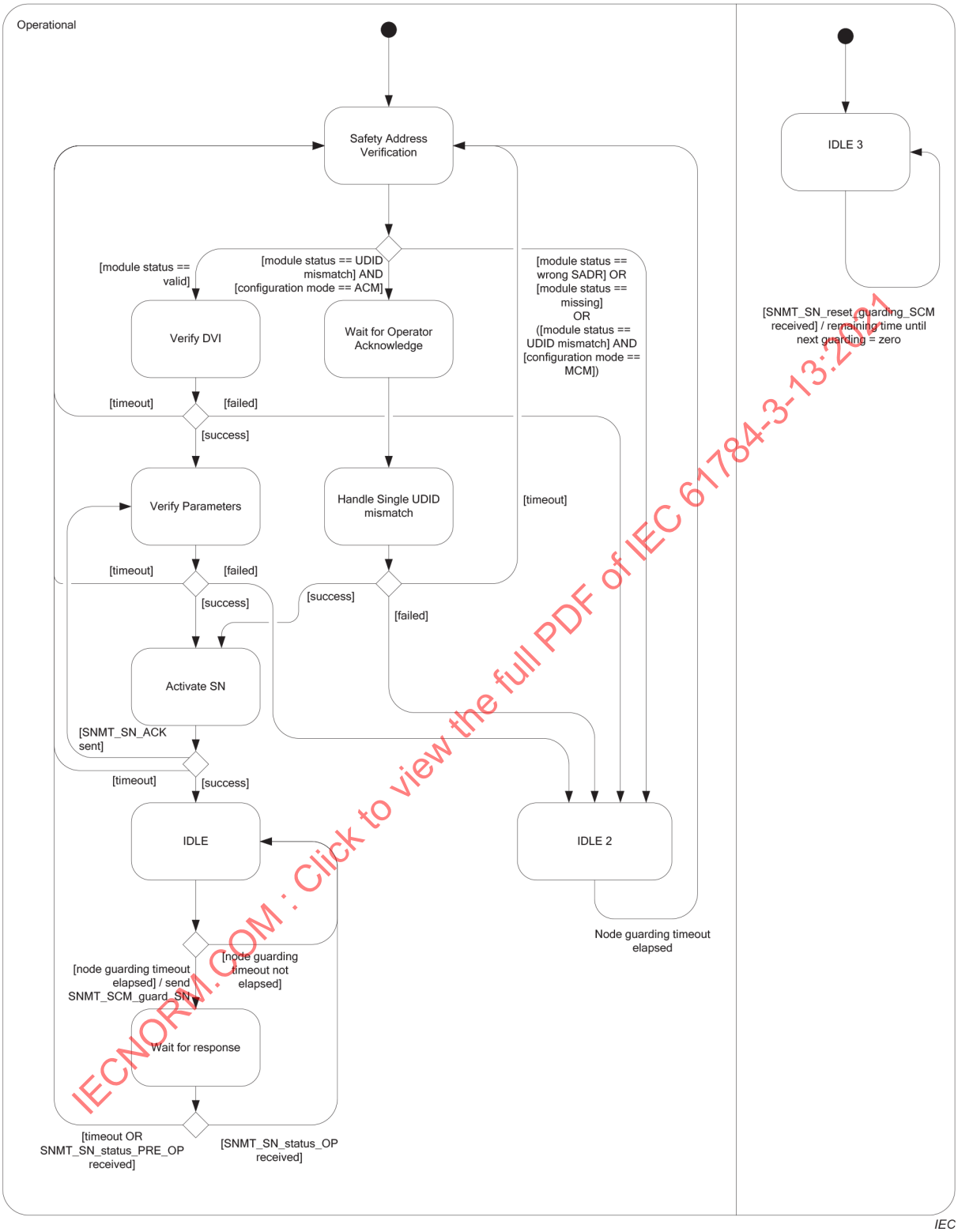
Table 257 – State and communication object relation

	Initialization	Stopped	Operational
SSDO		X	X

7.7.9.3 Operational

When the SCM enters state Operational, it shall start the UDID/SADR verification of all configured nodes. A UDID mismatch shall be handled separately. The parameters for all valid nodes shall be verified against the list of CRCs and Timestamps within the SOD (Object indices C400h – C7FEh). If a node is valid (correct UDID, SADR and DVI) and its parameters are also valid, it shall be set to Operational. Operational nodes get a cyclic message to keep them in Operational state.

The "SCM Operational" state diagram is performed for each SN in parallel and is specified by Figure 83.



Key

See Table 258 and Table 259.

Figure 83 – State diagram SCM Operational

Table 258 specifies items used in Figure 83.

Table 258 – SCM Operational state item description

Item	Description	Min value	Max value	SOD
	Guarding Timeout	1	UNSIGNED32	Object index 100Ch sub index 01h (see 7.5.4.6)

Table 259 specifies the SCM Operational states.

Table 259 – SCM Operational state description

State	Description
Safety Address Verification	The SCM verifies the UDID of the node. Also the correct SADR will be assigned to the node
Verify DVI	The DVI has to be checked to ensure that the new node meets the technical requirements
Verify Parameters	Parameter check and eventually re-parameterization of the SN
Activate SN	After successful parameterization, the SCM switches the SN to Operational
IDLE	Waits until next SNMT_SCM_guard _SN needs to be sent
Wait for response	Waits for the response of the SNMT_SCM_guard_SN service
Wait for Operator Acknowledge	If an unknown UDID was detected during start-up or operation, an operator-acknowledge is needed to accept and verify the new node. This state is only entered in Automatic Configuration Mode (ACM)
Handle Single UDID Mismatch	If a new SN was detected (replacement, commissioning, etc.), it is verified and checked to see if it meets the requirements or not
IDLE 2	If the node is not present, or does not meet the technical requirements (module was replaced with a wrong type), the node will not be handled for a node guarding interval. This ensures, that nodes, which are not present or nodes which were replaced with a different module type will get handled again, but do not produce too much bus activity
IDLE 3	Sets the remaining time until the next node guarding to zero in case an SNMT_SN_reset_guarding_SCM telegram is received. After a power up of a module, the module sends this type of telegram to the SCM. This ensures that the SCM recognizes (by the response to the guarding telegram) that the module is down and needs to be restarted

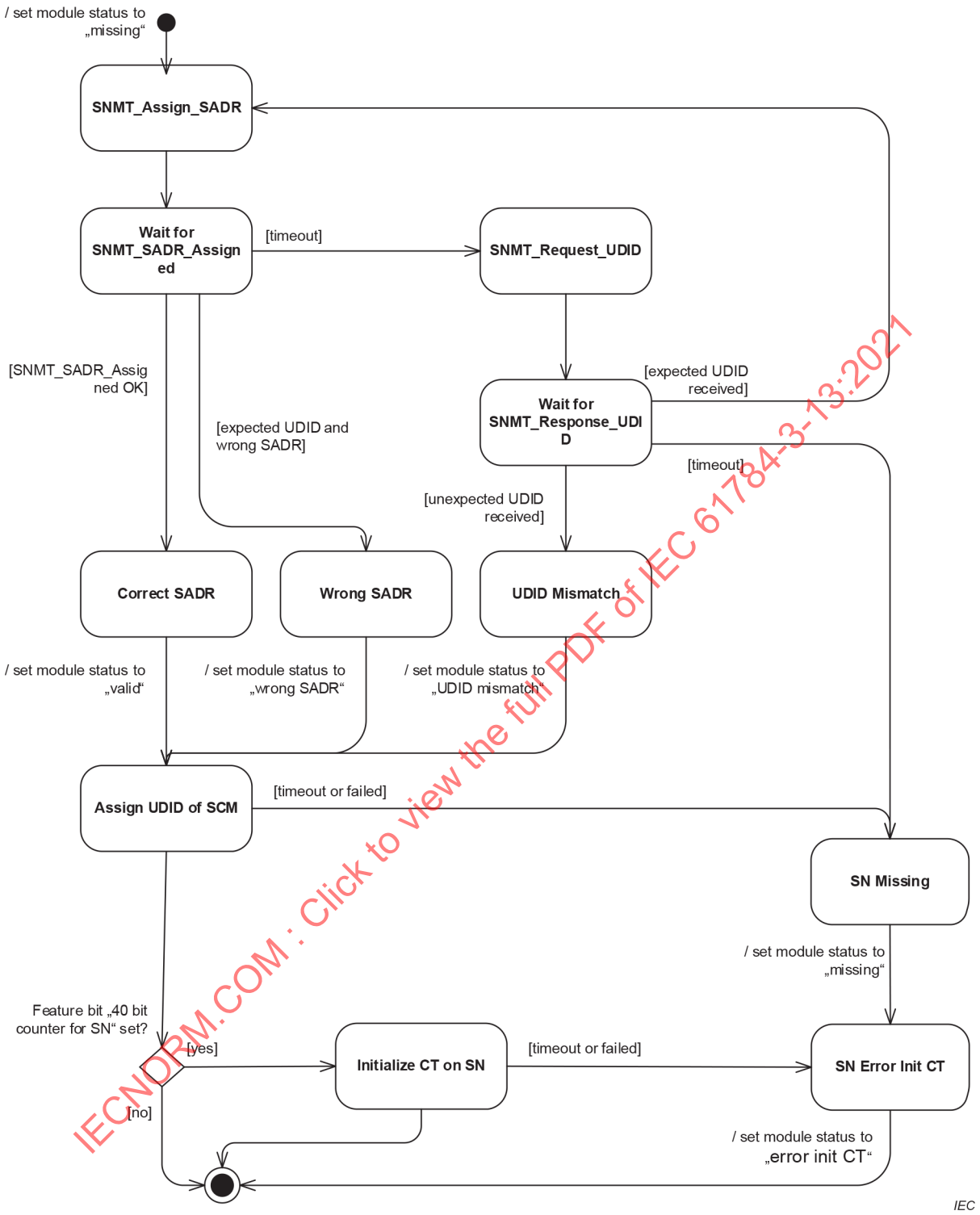
7.7.10 Address verification

The Address verification shall be used to verify the correctness of the safety network. All device configuration changes are detected due to the uniqueness of the UDID. The object indices CC01h-CFFFh shall correspond to the possible safety addresses 1d – 1023d.

In case of the 40 bit counter the necessary init value shall be written to the SN after the assignment of the SCM UDID but before any SSDO services are invoked.

This state machine shall apply in any configuration mode (ACM or MCM).

The "SCM address verification" state diagram is specified by Figure 84.



Key

See Table 260 and Table 261.

Figure 84 – State diagram SCM address verification

Table 260 specifies items used in Figure 84.

Table 260 – Address verification item description

Item	SOD
module status	Object index C400h – C7FEh sub index 05h (see 7.5.4.21)
expected UDID	Object index CC01h – CFFFh[a] sub index 1 – 254 (see 7.5.4.23)
assigned SADR	Object index CC01h – CFFFh[a] (see 7.5.4.23)
Feature bit "40 bit counter for SN"	Object index C400h – C7FEh[a] sub index 0Ch (see 7.5.4.21)
[a] The object indices CC01h – CFFFh shall correspond to the possible safety addresses 1d – 1023d.	

Table 261 specifies the SCM address verification states.

Table 261 – Address verification state description

State	Description
SNMT_Assign_SADR	Assigns the SADR to a configured node
Wait for SNMT_SADR_Assigned	The response contains the actual SADR. Some nodes might not be able to change the SADR because it is to be set using hardware switches
SNMT_Request_UDID	Requests the UDID of a configured node
Wait for SNMT_Response_UDID	The response from the node contains its UDID
Correct SADR	The SADR could be assigned. The node is valid
Wrong SADR	The UDID is known but the SADR could not be assigned. The node is invalid
UDID Mismatch	The node does not have the expected UDID. An operator-acknowledge is needed to accept and start the node
SN missing	The configured SN could not be reached. The node might be defective or not in the network. The node is missing
Assign UDID of SCM	Sends UDID of SCM to SN via the corresponding service
Initialize CT on SN	Send the initial CT value to the SN

7.7.11 Commissioning mode

FSCP 13/1 defines no explicit commissioning mode. The first time the SCM is started, all nodes shall be handled as nodes with a UDID mismatch.

7.7.12 Handle single UDID mismatch

7.7.12.1 General

Subclause 7.7.12 specifies the handling of nodes with UDID mismatch. If the non matching node (e.g. new node) meets the requirements (DVI, assignment of SADR, etc.), no re-configuration of the SCM shall be needed when replacing nodes.

The "SCM handle single UDID mismatch" state diagram is specified by Figure 85.

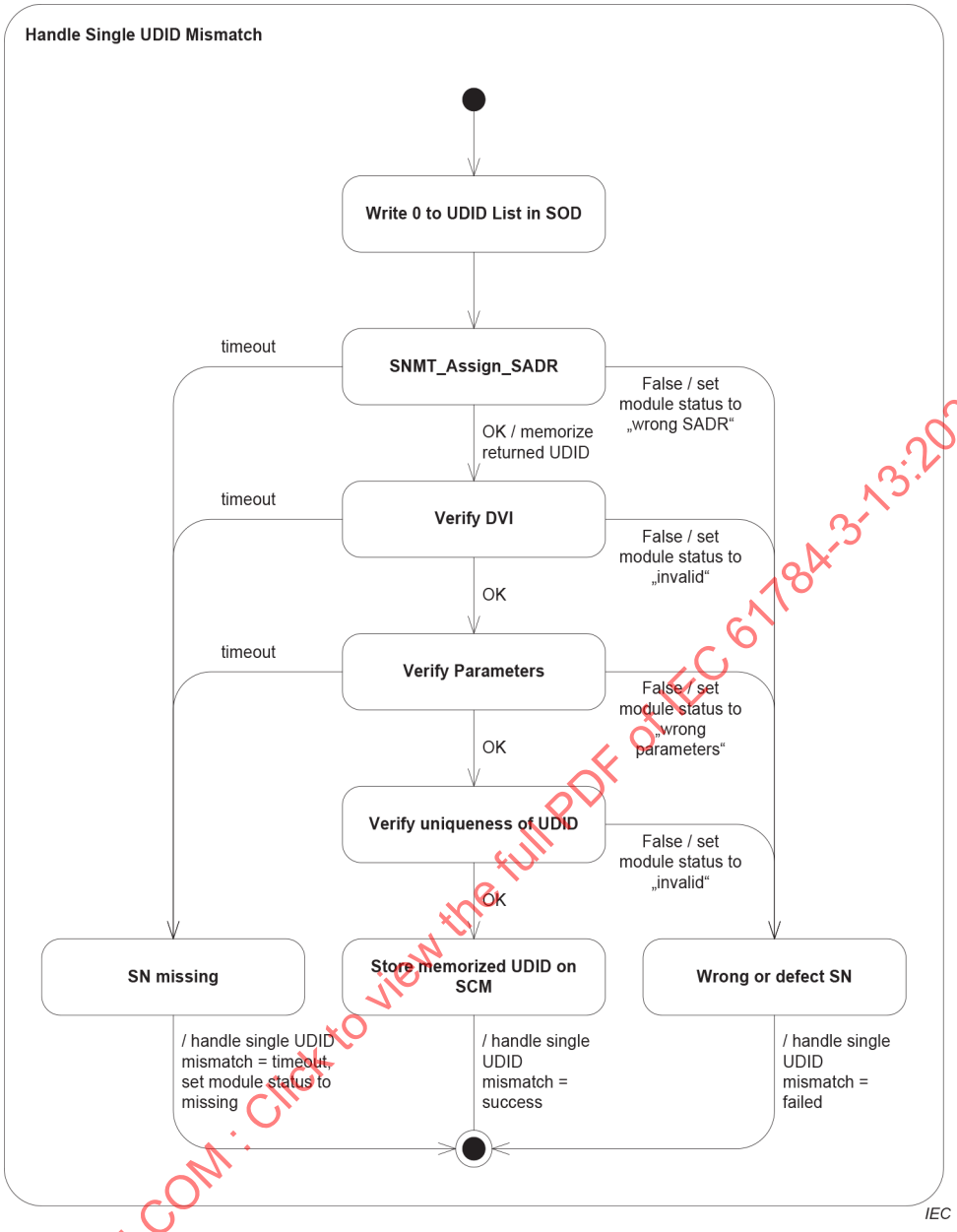


Figure 85 – State diagram SCM handle single UDID mismatch

Table 262 specifies the SCM handle single UDID mismatch states.

Table 262 – SCM handle single UDID mismatch state description

State	Description
Write 0 to UDID list in SOD	The SCM deletes the "old" stored UDID in the SOD (Object CC01h – CFFFh). 0 is no valid UDID
SNMT_Assign_SADR	The SCM writes the configured SADR to the module
Verify DVI	The DVI has to be checked to ensure that the new node meets the technical requirements
Verify Parameters	The parameters which are stored on the node are cross checked with the parameters stored on the SCM. The node will get new parameters if necessary/possible
Verify uniqueness of the UDID	The responded UDID is checked against the UDID list within the SOD (Object C001h-CFFFh)
Store memorized UDID on SCM	The new UDID is stored on the SCM to be able to recognize the node at the next start-up
SN missing	The SN did not answer one or more telegrams from the SCM. Therefore the SN will be handled like a missing node
Wrong or defect SN	The node does not meet the technical requirements specified in the SCM Possible reasons are: the DVI does not meet the requirements, it is not possible to change the SADR of the device (this can happen if the SADR is to be set with hardware switches), the parameters stored on the node are not equal to the parameters stored on the SCM but a download is not possible (this can happen if the SCM does not hold the parameters for a node and only knows the required checksum), the UDID is not unique within the SDN (this normally should never happen because the UDID is designed to be globally unique).

7.7.12.2 Verify parameters

The parameters (SOD) stored on the SN shall be compared with the parameters stored on the SCM.

It is not necessary to store all parameters for all nodes on the SCM. Alternatively the required parameter checksum shall be stored on the SCM. In this case, the SCM shall verify the correctness of the parameters on the node. If the parameters are not correct (the checksum does not match), the SCM will not be able to re-configure such nodes.

The "SCM verify parameters" state diagram is specified by Figure 86.

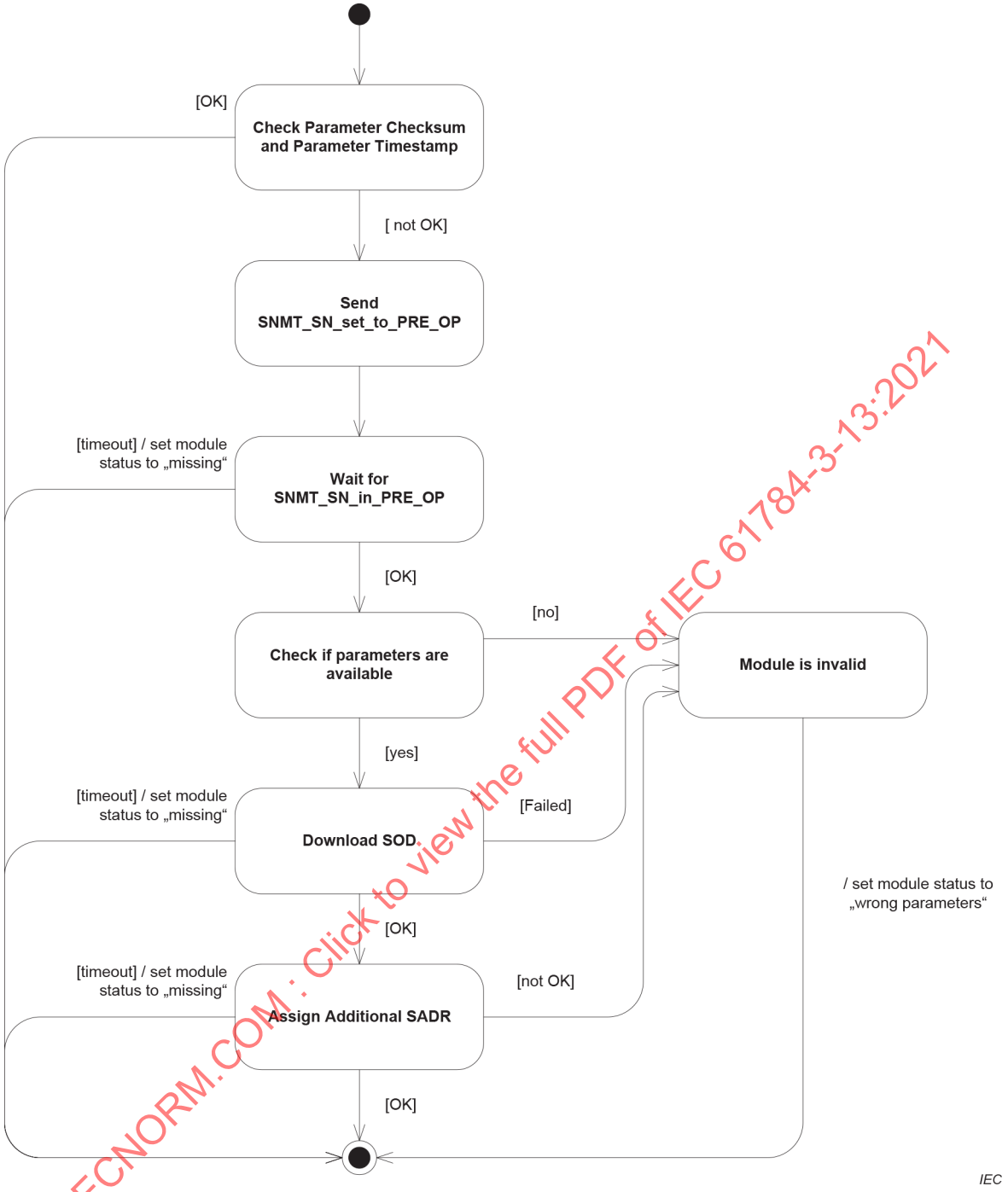


Figure 86 – State diagram SCM verify parameters

Table 263 specifies the SCM verify parameters states.

Table 263 – SCM verify parameters state description

State	Description
Check Parameter Checksum and Parameter Timestamp	The SCM checks if the current parameters for the node are the same which are stored on the SCM
Send SNMT_SN_set_to_PRE_OP	Resets the SN
Wait for SNMT_SN_in_PRE_OP	Waits for the reply of the reset service
Check if parameters are available	If the checksum on the node is different from the checksum on the SCM, the SCM checks if parameters for re-configuration of the node are available
Download SOD	The SCM reconfigures the node
Module is invalid	If the parameters on the node are not equal to the parameters stored on the SCM and reconfiguration of the module is not possible, the module is invalid and shall not be set to Operational
Assign additional SADRs	The SCM assigns the SADR for the optional TxSPDOs 2 – 1 023 of the corresponding SN

7.7.13 Activate SN

The SCM shall try to set the SN to Operational. The SN ether switches to Operational or an error occurred during switching to Operational. An error shall be reported with the SNMT_SN_FAIL telegram. The SCM receives this telegram and shall report it to the application. The vendor specific application shall decide how to cope with the situation. It shall send an acknowledgement telegram SNMT_SN_ACK if the error can get acknowledged or shall keep the SN in Pre-Operational if the problem cannot be solved.

The "activate SN" state diagram is specified by Figure 87.

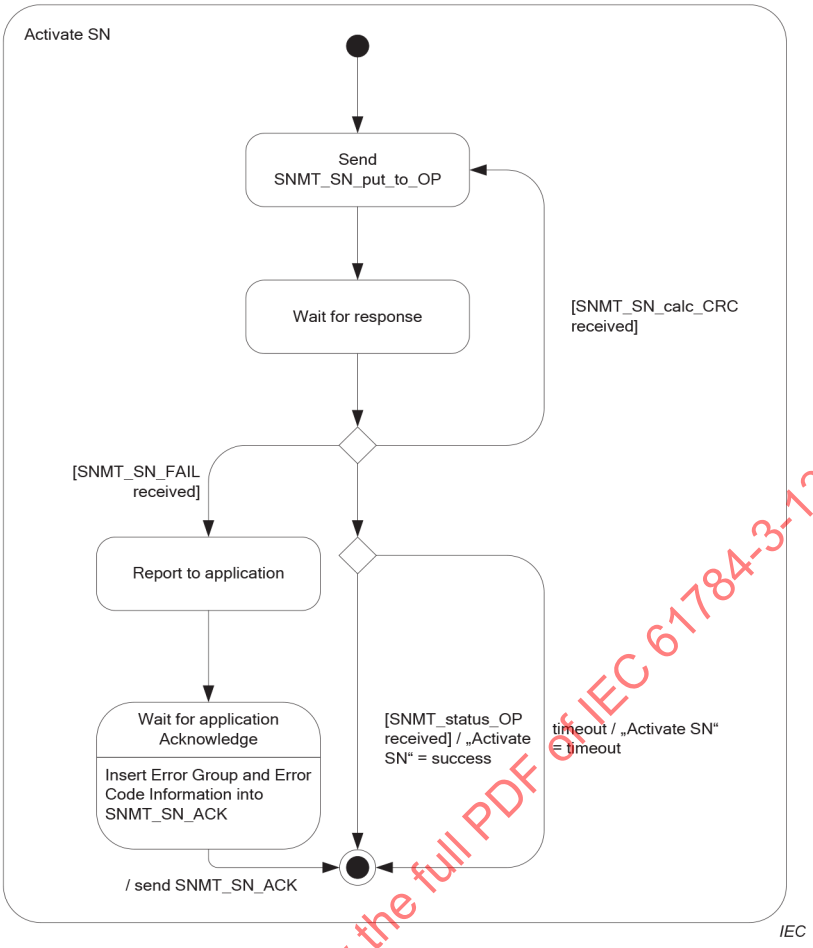


Figure 87 – State diagram activate SN

Table 264 specifies the activate SN states.

Table 264 – Activate SN state description

State	Description
Send SNMT_SN_put_to_OP	The SCM sends an OP request to the CN
Wait for response	The SCM waits for the response from the SN
Report to application	In case of an error the error is reported to the application
Wait for application Acknowledge	The application may set the error group and the error code information for the acknowledgement of the error

7.7.14 Device exchange

Exchanged devices shall send the SNMT_SN_in_PRE_OP message to the SCM. The SCM shall handle those exchanged nodes as UDID-mismatch nodes. After operator acknowledge, they shall be programmed and set to Operational.

8 Safety communication layer management

8.1 General

Clause 8 specifies the parameterization interface of SOD.

The parameterization of FSCP 13/1 devices shall be done using a software tool which was developed and assessed according to IEC 61508. Such a software tool shall always be part of a complete safety related automation system.

The software tool shall transfer a valid SOD configuration to the device by defined FSCP 13/1 services. Direct vendor specific access to the device shall not be supported.

8.2 Configuration of a Safety Domain

A single safety domain shall be configured by defining the SOD for an SCM. A software tool shall create a valid SOD configuration for such an SCM, containing every parameter and setting for each participant of this SD. This configuration shall be provided to the SCM in a secure way and prior to the power-up of the network.

During the power-up, the SCM shall query each individual SN within the SD and guarantee, that the correct and valid configuration is stored on each individual device

The SCM shall hold a parameter set for each SN within the SD. The parameter set can contain:

- General communication and timing parameters,
- SPDO communication parameters,
- SPDO mapping information,
- Device specific parameters,
- Channel mapping configuration.

Additionally the timestamp and the checksum of the parameter configuration shall be transmitted during power-up, enabling each device individually to verify, if the transmitted parameter set is valid.

If the device supports additional parameter sets, these shall be stored on the SCM before power-up as well, and shall be transported using openSAFETY services, as they are needed by the SN.

Only if all parameters are verified, and every device within a SD has verified, that it's configuration is valid and transferred successfully, the SD is fully configured and may enter the operational phase.

Parameters which where left at their default state, may be omitted from the configuration send to the SN, and the SN shall assume the default value is used for such entries.

8.3 Parameter check mechanism

The integrity of parameters within the SOD shall be protected by calculating a CRC checksum for the SOD entries. For each 4 k bit block of SOD data, a 32 bit checksum shall be calculated and stored in the DVI object entry of the SOD, see 7.5.4.9.

An SCM may either upload only the parameter timestamp from an SN or the timestamp together with the stored CRCs. Depending on this information the SCM shall decide, if a reconfiguration of the SN is necessary or not.

9 System requirements

9.1 Indicators and switches

There are no specific indicators and switches requirements for FSCP 13/1.

9.2 Installation guidelines

The installation guidelines of IEC 61918 and the CPF 13 specific amendments in IEC 61784-5-13 shall apply. Additionally, 6.1.3.3.2 and 6.1.3.3.3 shall apply.

9.3 Safety function response time

The safety function response time is the worst case elapsed time following an actuation of a safety sensor (for example switch, pressure transmitter, light curtain) connected to a fieldbus, before the corresponding safe state of its safety actuator(s) (for example relay, valve, drive) is achieved in the presence of errors or failures in executing the safety function.

The demand (actuation) on a safety function is caused either by an analog signal crossing a threshold or by a digital signal changing state.

Figure 88 shows an example of typical components making up a safety function response time.

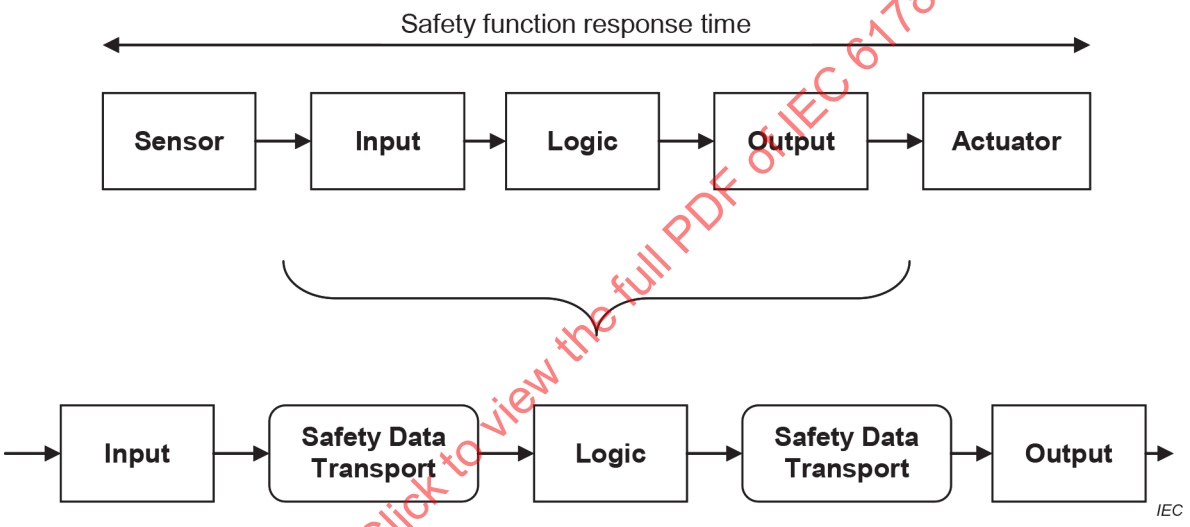


Figure 88 – Safety function response time

As shown in Figure 88, the safety function response time is calculated using Formula (7):

$$t_{SFRT} = t_{Sensor} + t_{Input} + 2 \times t_{NRT} + t_{Controller} + t_{Output} + t_{Actuator} \quad (7)$$

where

t_{SFRT} is the safety function response time;
t_{NRT} is the network reaction time, refer to Formula (8);
t_{Sensor} is the sensor reaction time;
t_{Input} is the input reaction time;
$t_{Controller}$ is the controller reaction time;
t_{Output} is the output reaction time;
$t_{Actuator}$ is the actuator reaction time.

All but the network reaction time are outside the scope of this document.

The network reaction time is calculated by Formula (8).

$$t_{NRT} = t_{SCT} + t_{maxSPD} + 2 \times t_{TRC} \times c_{clockTol} \quad (8)$$

where

t_{NRT} is the network reaction time;

t_{SCT} is the safety control time (see SCT_U32, Table 158);

t_{maxSPD} is the maximum SPDO propagation delay (see MaxSPDOPropagationDelay_U16, Table 165);

t_{TRC} is the time request cycle (see TimeRequestCycle_U32, Table 167);

$c_{clockTol}$ is the tolerance of the hardware clocks.

9.4 Duration of demands

The demand has to exist at least for the duration of the network reaction time (see 9.3) to guarantee that the functional safety communication system detects the demand.

9.5 Constraints for calculation of system characteristics

9.5.1 General

Several constraints shall be used for calculating safety related characteristics of systems using FSCP 13/1.

9.5.2 Number of sinks limit

FSCP 13/1 implementations are limited to a maximum of 1 023 producing and 1 023 consuming nodes. The number of producing nodes and the network cycle time have to be considered to not exceed the message rate limit (see 9.5.3).

The residual error rate values given in 9.5.6 are valid for a single point to point relationship. To calculate the residual error rate for a specific safety function the formulae (1) and (2) in IEC 61784-3:2021 shall be applied.

9.5.3 Message rate limit

The maximum rate of safety related messages shall be limited by the limit for the residual error rate for SIL 3 of 10^{-9} per hour shown in Formula (9):

$$mr = \frac{10^{-9}}{R_{SL} \cdot 3600} \quad (9)$$

where

mr is the maximum message rate in 1/s;

R_{SL} is the residual error rate (see 9.5.6).

9.5.4 Message payload limit

The message payload of one safety PDU shall not exceed 240 octets.

9.5.5 Bit error rate considerations

IEC 61784–3 demands a bit error probability of 10^{-2} to be used in probability calculations unless a better value can be proven.

For Ethernet based transport layer it is known that a data paket has a minimum size of 512 bits (64 octets). A bit error rate of 10^{-2} means that one out of 100 bits is corrupted. In case of an Ethernet based transport layer every paket is corrupted. So a base communication setup with this bit error rate would not be possible or in other words, a working Ethernet based transport layer garants a bit error rate of 10^{-3} or higher.

For FSCP 13/1 it is also known, that a stable communication within an industrial application requires a bit error rate, which allows statistically at least one error-free SPDU per connection. If the bit error rate is higher, statistically every SPDU within a connection is corrupted and no useful communication can be established.

The assumed bit error rate for a communication layer running FSCP 13/1 is

– transport layer specific if the quality of the transport layer is known, or
– 10^{-3} if a cable based Ethernet communication layer is used, or
– 1/(SPDU length) in all other cases.

NOTE For FSCP 13/1 the usage of a bit error probability of 10^{-3} for Ethernet based communication layers has been approved by TÜV.

9.5.6 Residual error rate

FSCP 13/1 fulfils the requirements of IEC 61508 / SIL 3 for reliability and safety. For SIL 3, an overall system residual error rate < 10^{-7} shall be achieved. The safety related network shall not use up more than 1 % of the system residual error rate, therefore the network residual error rate shall be below 10^{-9}.

Implementations according to FSCP 13/1 provide a network residual error rate below 10^{-9}. Table 265 shows the residual error rate R_{SL} for FSCP 13/1.

Table 265 – Residual error rate

	BER = 1/(SPDU length)	BER = 0,001
Payload[Octets]	R_{SL}	R_{SL}
1	$8{,}23 \times 10^{-15}$	$2{,}61 \times 10^{-22}$
2	$7{,}49 \times 10^{-15}$	$7{,}34 \times 10^{-22}$
3	$5{,}89 \times 10^{-15}$	$1{,}56 \times 10^{-21}$
4	$4{,}37 \times 10^{-15}$	$2{,}78 \times 10^{-21}$
5	$3{,}17 \times 10^{-15}$	$4{,}45 \times 10^{-21}$
6	$2{,}29 \times 10^{-15}$	$6{,}57 \times 10^{-21}$
7	$1{,}65 \times 10^{-15}$	$9{,}14 \times 10^{-21}$
8	$1{,}21 \times 10^{-15}$	$1{,}22 \times 10^{-20}$
9	$9{,}34 \times 10^{-21}$	$2{,}65 \times 10^{-25}$
10	$6{,}96 \times 10^{-21}$	$3{,}21 \times 10^{-25}$
11	$5{,}24 \times 10^{-21}$	$3{,}83 \times 10^{-25}$
12	$3{,}98 \times 10^{-21}$	$4{,}48 \times 10^{-25}$
13	$3{,}06 \times 10^{-21}$	$5{,}19 \times 10^{-25}$
14	$2{,}37 \times 10^{-21}$	$5{,}92 \times 10^{-25}$
15	$1{,}86 \times 10^{-21}$	$6{,}70 \times 10^{-25}$

	BER = 1/(SPDU length)	BER = 0,001
Payload[Octets]	R_{SL}	R_{SL}
16	$1,47 \times 10^{-21}$	$7,52 \times 10^{-25}$
17	$1,17 \times 10^{-21}$	$8,36 \times 10^{-25}$
18	$9,39 \times 10^{-22}$	$9,24 \times 10^{-25}$
19	$7,59 \times 10^{-22}$	$1,02 \times 10^{-24}$
20	$6,18 \times 10^{-22}$	$1,11 \times 10^{-24}$
21	$5,07 \times 10^{-22}$	$1,21 \times 10^{-24}$
22	$4,18 \times 10^{-22}$	$1,30 \times 10^{-24}$
23	$3,47 \times 10^{-22}$	$1,41 \times 10^{-24}$
24	$2,89 \times 10^{-22}$	$1,51 \times 10^{-24}$
25	$2,43 \times 10^{-22}$	$1,62 \times 10^{-24}$
26	$2,05 \times 10^{-22}$	$1,73 \times 10^{-24}$
27	$1,74 \times 10^{-22}$	$1,84 \times 10^{-24}$
28	$1,48 \times 10^{-22}$	$1,95 \times 10^{-24}$
29	$1,26 \times 10^{-22}$	$2,06 \times 10^{-24}$
30	$1,08 \times 10^{-22}$	$2,18 \times 10^{-24}$
31	$9,34 \times 10^{-23}$	$2,3 \times 10^{-24}$
32	$8,08 \times 10^{-23}$	$2,42 \times 10^{-24}$
40	$2,85 \times 10^{-23}$	$3,42 \times 10^{-24}$
44	$1,80 \times 10^{-23}$	$3,95 \times 10^{-24}$
48	$1,18 \times 10^{-23}$	$4,48 \times 10^{-24}$
52	$8,01 \times 10^{-24}$	$5,02 \times 10^{-24}$
56	$5,57 \times 10^{-24}$	$5,57 \times 10^{-24}$
60	$3,96 \times 10^{-24}$	$6,11 \times 10^{-24}$
64	$2,87 \times 10^{-24}$	$6,65 \times 10^{-24}$
96	$3,78 \times 10^{-25}$	$1,06 \times 10^{-23}$
128	$8,94 \times 10^{-26}$	$1,33 \times 10^{-23}$
160	$2,94 \times 10^{-26}$	$1,45 \times 10^{-23}$
192	$1,20 \times 10^{-26}$	$1,45 \times 10^{-23}$
240	$4,02 \times 10^{-27}$	$1,28 \times 10^{-23}$

9.6 Maintenance

9.6.1 Diagnostic information

Diagnostic information according to 7.5.4 shall be stored in the object 1001h, 1002h, 1003h by the SN. This information can be read by non safety related applications.

9.6.2 Replacement of safety related devices

9.6.2.1 SN replacement

After replacing the affected SNs, the maintenance technician has to decide.

– If only one SN is replaced, is the SCM able to handle the replacement automatically? Only a simple confirmation by the maintenance technician is requested.

– If more than one SN is replaced, the configuration (see 6.2.3.3) shall be repeated and verification of the safety functionality of the replaced SNs shall be done (see 6.2.3.4).

9.6.2.2 Replacement of SN running the SCM

If the configuration and the application with the corresponding parameters are available after the replacement, the SCM replacement has no influence on the safety functions.

If any data is lost, the steps according to 6.2.3 shall be repeated. In accordance with 6.2.3.4, a complete verification shall be done (as it is required during commissioning).

9.6.3 Modification

After modification, the steps according to 6.2.3 shall be repeated. In accordance with 6.2.3.4, a complete verification shall be done (as it is required during commissioning).

9.6.4 Machine part changing

All changeable machine parts shall be verified against the safety specification for the application at least one time according to the procedure listed in 6.2.3.4. After verification, the SCM knows the configuration for all changeable machine parts; this means that after changing machine parts, no further steps are required.

Open input situations due to missing producers after machine part changes shall be handled by the application.

9.6.5 Firmware update of safety related nodes

A firmware update for a safety related node is a vendor specific task. There is no service within FSCP 13/1 to support this task.

9.6.6 Machine check due to service interval

FSCP 13/1 requires no special check.

9.7 Safety manual

Vendors of devices which implement a safety communication layer compliant to this document shall supply a safety manual which follows the requirements in IEC 61508 which shall at least inform the user:

a) of constraints for calculation of system characteristics, see 9.5,

b) of their responsibilities in the proper parameterization of the device, and

c) about preconditions to be fulfilled for programming and parameterization modes, see 6.2.2.2.

10 Assessment

10.1 General

Assessment of conformity of an FSCP 13/1 implementation or device by a competent assessment body is highly recommended. Each node which is within an interoperable FSCP 13/1 communication network shall provide compatibility according to this document. Only the compatibility of devices guarantees the availability and safety of FSCP 13/1 networks.

Assessment flow of FSCP 13/1 devices is specified by Figure 89.

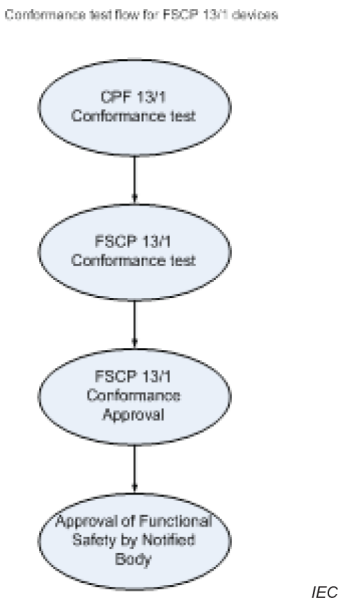


Figure 89 – Assessment flow of devices

Additional information can be found in Annex B.

10.2 CP 13/1 assessment

For FSCP 13/1 assessment, the underlying communication layer CP 13/1 for all non safety communications is required. For successful interactions between different safety nodes, all mandatory CP 13/1 services shall be implemented according to IEC 61784-3 CP 13/1 and IEC 61158-5-13.

10.3 FSCP 13/1 conformance test

The FSCP 13/1 conformance test ensures that an FSCP 13/1 device conforms to all safety services and measurements according to this document. The test should be done by a competent assessment body or by an assessed test tool. The result is a notice of assessment and a detailed test report which is needed for obtaining the functional safety assessment from a competent assessment body.

10.4 Approval of functional safety by competent assessment body

If machinery needs official approval (e.g. for IEC 61508 or other safety related standards), all FSCP 13/1 devices being used need this approval too. According to the standards, functional safety should be assessed by a competent assessment body.

Annex A
(informative)

Additional information for functional safety communication profiles of CPF 13

A.1 Hash function calculation

The following code example shows a possible way to calculate the CRC-8 and CRC-16 used by FSCP 13/1.


```
/*****************************************************************************
openSAFETY_CRC.c – Reference implementation for openSAFETY™ CRC-Calculations
This programm should be a reference implementation for the two openSAFETY™ CRC-
Calculations
It is possible to compare the implemented CRC’s with the mentioned reference values
from this programm.
This programm is not optimized for performance.

The programm depends on an 8 Bit/Byte machine.
unsigned int has to have 16 bits or more.

Summary:
1. Append zero bits to the data stream
2. Devide the complete data stream by using CRC arithmetic
3. The reminder is the checksum

POWERLINKsafety V1.0
CRC Referenz calculation
Review: B&R Industrial Automation GmbH
*****************************************************************************/

/* the polynomial is specified without its top bit, the top bit is contained
implicitly in the algorithm because 1 xor 1 is always 0 (register optimized) */
#define c_poly1 0x2F            /* x8+x5+x3+x2+x+1 = 0x12F */
#define c_poly2 0xD5            /* x8+x7+x6+x4+x2+1 = 0x1D5 */
#define c_poly3 0x5935          /* x16+x14+x12+x11+x8+x5+x4+x2+1  0x15935*/
#define c_poly4 0x755B          /* x16+x14+x13+x12++x10+x8+x6+x4+x3+x+1 0x1755B */

/* prototype of left shifting by one bit */
unsigned char
   uc_array_shift_left(unsigned char *, unsigned int);
/* prototype CRC8 */
unsigned char
   uc_openSAFETY_CRC8(unsigned char, unsigned char, unsigned char *);
/* prototype CRC16 */
unsigned int
   ui_openSAFETY_CRC16(unsigned int, unsigned int, unsigned char *);
/* test program */
/* the length of sub frame depands from length of payload */
#define payload 8                   /* length of payload in reference sub frame */
#define c_sub_length1 (payload + 4) /* length of reference sub frame 1 w/o CRC  */
#define c_sub_length2 (payload + 5) /* length of reference sub frame 2 w/o CRC  */

/*first part of openSAFETY™ short telegram    */
unsigned char uc_subframe_1[c_sub_length1 + 2];

/*second part of openSAFETY™ short telegram   */
unsigned char uc_subframe_2[c_sub_length2 + 2];

int main(void)
{
unsigned int result;
```

```
/* reference data stream for first sub frame */
                              /*MSB=bit 7 --- LSB bit 0              */
  uc_subframe_1[ 0] = 0x23;   /*  1. byte address 0-7                 */
  uc_subframe_1[ 1] = 0xC8;   /*  2. byte address 8-9, ID             */
  uc_subframe_1[ 2] = 0x08;   /*  3. byte LE                          */
  uc_subframe_1[ 3] = 0x34;   /*  4. byte CT 0-7                      */
  uc_subframe_1[ 4] = 0x11;   /*  5. byte DB 1                        */
  uc_subframe_1[ 5] = 0x22;   /*  6. byte DB 2                        */
  uc_subframe_1[ 6] = 0x33;   /*  7. byte DB 3                        */
  uc_subframe_1[ 7] = 0x44;   /*  8. byte DB 4                        */
  uc_subframe_1[ 8] = 0x55;   /*  9. byte DB 5                        */
  uc_subframe_1[ 9] = 0x66;   /* 10. byte DB 6                        */
  uc_subframe_1[10] = 0x77;   /* 11. byte DB 7                        */
  uc_subframe_1[11] = 0x88;   /* 12. byte DB 8                        */

/* reference data stream for second sub frame */
                              /*MSB=Bit 7 --- LSB BIT 0              */
  uc_subframe_2[ 0] = 0x22;   /*  1. byte address xor domain 0-7      */
  uc_subframe_2[ 1] = 0xC8;   /*  2. byte address xor domain 8-9, ID  */
  uc_subframe_2[ 2] = 0x12;   /*  3. byte CT 8-15                     */
  uc_subframe_2[ 3] = 0x56;   /*  4. byte TADR 0-7                    */
  uc_subframe_2[ 4] = 0x30;   /*  5. byte TADR 8-9, TR                */
  uc_subframe_2[ 5] = 0x11;   /*  6. byte DB 1                        */
  uc_subframe_2[ 6] = 0x22;   /*  7. byte DB 2                        */
  uc_subframe_2[ 7] = 0x33;   /*  8. byte DB 3                        */
  uc_subframe_2[ 8] = 0x44;   /*  9. byte DB 4                        */
  uc_subframe_2[ 9] = 0x55;   /* 10. byte DB 5                        */
  uc_subframe_2[10] = 0x66;   /* 11. byte DB 6                        */
  uc_subframe_2[11] = 0x77;   /* 12. byte DB 7                        */
  uc_subframe_2[12] = 0x88;   /* 13. byte DB 8                        */

/* calculation of CRCs with reference data stream */
result = uc_openSAFETY_CRC8(c_sub_length1, c_poly1, &uc_subframe_1[0]);
/* result = 0x3c */
uc_subframe_1[c_sub_length1]   = result;         /* 13. byte CRC-8 bit 0-7    */
result = uc_openSAFETY_CRC8(c_sub_length2, c_poly2, &uc_subframe_2[0]);
/* result = 0x4f */
uc_subframe_2[c_sub_length2]   = result;         /* 14. byte CRC-8 bit 0-7    */
result =  ui_openSAFETY_CRC16(c_sub_length1, c_poly3, &uc_subframe_1[0]);
/* result = 0x0374 */
  uc_subframe_1[c_sub_length1]   = result >> 8;    /* 13. byte CRC-16 bit 15-8  */
  uc_subframe_1[c_sub_length1+1] = result;         /* 14. byte CRC-16 bit 0-7   */

result =  ui_openSAFETY_CRC16(c_sub_length2, c_poly4, &uc_subframe_2[0]);
/* result = 07031 */
  uc_subframe_2[c_sub_length2]   = result >> 8;    /* 14. byte CRC-16 bit 15-8  */
  uc_subframe_2[c_sub_length2+1] = result;         /* 15. byte CRC-16 bit 0-7   */

return 0;
}


/*****************************************************************************
function uc_openSAFETY_CRC8()            version 1.0
date 2005-04-22
maxmuim datalength of 14 byte
return: unsinged char checksum
parameter:  1 byte frame length
1 byte polynominal
1 pointer start adress of frame array
*****************************************************************************/
unsigned char uc_openSAFETY_CRC8(unsigned char uc_sub_length,
unsigned char uc_poly,
unsigned char *uc_subframe)
{
  unsigned char uc_checksum;                  /* reminder as crc checksum     */
  unsigned char uc_i;                         /* counter variable             */
  unsigned int ui_stream_le;                  /* stream length                */
  unsigned char uc_data[15];                  /* include the complete stream
                                                  with appended zeros         */
```

```
                                        /*array is not optimized!       */

for (uc_i = 0; uc_i < uc_sub_length; uc_i++)
  {
  uc_data[uc_i]=uc_subframe[uc_i];          /*copy the frame in stream and */
  }
uc_data[uc_sub_length] = 0;                 /*append the zeros*/
ui_stream_le = uc_sub_length + 1;
/* CRC calculation by shifting and xor */
for (uc_i=0; uc_i < uc_sub_length * 8; uc_i++)
  {
  if (uc_array_shift_left( &uc_data[0], ui_stream_le))
      {
      /* XOR when bit falling out is not zero */
      uc_data[0] = uc_data[0] ^ uc_poly;
      }
  }

/* result*/
uc_checksum = uc_data[0];
return (uc_checksum);
}

/*****************************************************************************
function uc_openSAFETY_CRC16()          version 1.0
date 2005-04-22
maxmuim datalength of 261 byte
return: unsinged int checksum
parameter:  1 integer frame length
1 integer polynominal
1 pointer start adress of frame array
*****************************************************************************/

unsigned int ui_openSAFETY_CRC16(unsigned int ui_sub_length,
unsigned int ui_poly,
unsigned char *uc_subframe)
{
unsigned int ui_crc_sum=0;
unsigned int ui_i;
unsigned int ui_stream_le;                  /* stream length               */
unsigned char uc_data[263];
unsigned char uc_poly[2];


uc_poly[1]= ui_poly;        /*LSB*/
uc_poly[0]= ui_poly >>= 8;  /*MSB*/


for (ui_i = 0; ui_i < ui_sub_length; ui_i++)
  {
  uc_data[ui_i]=uc_subframe[ui_i];          /* copy the frame in stream and */
  }
uc_data[ui_sub_length] = 0;                 /* append the zeros */
uc_data[ui_sub_length+1] = 0;               /* append the zeros */
ui_stream_le = ui_sub_length + 2;
/* CRC calculation by shifting and xor */
for (ui_i=0; ui_i < ui_sub_length * 8; ui_i++)
  {
  if (uc_array_shift_left( &uc_data[0], ui_stream_le))
    {
    /* XOR when bit falling out is not zero */
    uc_data[0] = uc_data[0] ^ uc_poly[0];
    uc_data[1] = uc_data[1] ^ uc_poly[1];
    }
  }

/* result */
ui_crc_sum = uc_data[0];
ui_crc_sum <<= 8;
ui_crc_sum = ui_crc_sum | uc_data[1];
```

```
return(ui_crc_sum);
}


/*****************************************************************************
function uc_array_shift_left()          version 1.0
date 2005-04-22
maxmuim datalength of UINT_MAX
return: unsigned char value of bit shifted out at left
parameter: 1 pointer start adress of stream array
1 integer field length
*****************************************************************************/

unsigned char uc_array_shift_left(unsigned char *uc_info, unsigned int us_fields)
{
signed int si_i=0;              /* count variable */
unsigned char msb=0;            /* buffer */
unsigned char carry=0;          /* carry  */

/* shift entire array from right to left */
for (si_i = us_fields – 1; si_i >= 0; si_i--)
  {
  msb = uc_info[si_i] & 0x80;
  msb >>= 7;

  uc_info[si_i] <<= 1;                    /* shift left */
  uc_info[si_i] |= carry;                /* add carry */
  carry = msb;                           /* msb to next carry */
  }

return (carry);
}
```

A.2 Void

Void

Annex B
(informative)

Information for assessment of the functional safety communication profiles of CPF 13

According to IEC rules, this document does not make a statement on how to validate conformance. However, test and validation of compliance of FSCP 13/1 devices with IEC 61784-3-13 may be required by law.

Corresponding information relative to the test and compliance with this document can be retrieved from the local National Committees of the IEC or from the relevant fieldbus organization.

NOTE For IEC 61784-3-13, the relevant fieldbus organization is EPSG – Ethernet POWERLINK Standardization Group, see www.ethernet-powerlink.org.

Bibliography

[1] IEC 60050 (all parts), *International Electrotechnical Vocabulary (IEV)* available at <http://www.electropedia.org/>

NOTE See also the IEC Multilingual Dictionary – Electricity, Electronics and Telecommunications (available on CD-ROM and at <http://www.electropedia.org>)

[2] IEC 60050-191:1990[4], *International Electrotechnical Vocabulary (IEV) – Part 191: Dependability and quality of service*

[3] IEC 61000-1-2, *Electromagnetic compatibility (EMC) – Part 1-2: General – Methodology for the achievement of functional safety of electrical and electronic systems including equipment with regard to electromagnetic phenomena*

[4] IEC 61000-6-7, *Electromagnetic compatibility (EMC) – Part 6-7: Generic standards – Immunity requirements for equipment intended to perform functions in a safety-related system (functional safety) in industrial locations*

[5] IEC 61010-2-201, *Safety requirements for electrical equipment for measurement, control and laboratory use – Part 2-201: Particular requirements for control equipment*

[6] IEC 61131-2, *Industrial-process measurement and control – Programmable controllers – Part 2: Equipment requirements and tests*

[7] IEC 61131-6, *Programmable controllers – Part 6: Functional safety*

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

[9] IEC 61158-5 (all parts), *Industrial communication networks – Fieldbus specifications – Part 5: Application layer service definition*

[10] IEC 61326-3-1, *Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) – General industrial applications*

[11] IEC 61326-3-2, *Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 3-2: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) – Industrial applications with specified electromagnetic environment*

[12] IEC 61496 (all parts), *Safety of machinery – Electro-sensitive protective equipment*

[13] IEC 61508-1:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements*

[14] IEC 61508-4:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

[15] IEC 61508-5:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5: Examples of methods for the determination of safety integrity levels*

4 Withdrawn

[16] IEC 61511 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*

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

[18] IEC 61784-3 (all parts), *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses*

[19] IEC 61784-5 (all parts), *Industrial communication networks – Profiles – Part 5: Installation of fieldbuses – Installation profiles for CPF x*

[20] IEC 61800-5-2, *Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional*

[21] IEC 62061, *Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems*

[22] IEC 62280:2014, *Railway applications – Communication, signalling and processing systems – Safety related communication in transmission systems*

[23] IEC 62443 (all parts), *Industrial communication networks – Network and system security*

[24] ISO/IEC Guide 51:2014, *Safety aspects – Guidelines for their inclusion in standards*

[25] ISO/IEC 2382:2015, *Information technology – Vocabulary*

[26] ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

[27] ISO 10218-1, *Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*

[28] ISO 13849 (all parts), *Safety of machinery – Safety-related parts of control systems*

[29] ISO 13849-1:2015, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*

[30] ISO 13849-2, *Safety of machinery – Safety-related parts of control systems – Part 2: Validation*

[31] ANDREW S. TANENBAUM, DAVID J. WETHERALL, *Computer Networks*, 5th Edition, Prentice Hall, N.J., ISBN-10: 0132126958, ISBN-13: 978-0132126953

[32] W. WESLEY PETERSON, EDWARD J. WELDON, *Error-Correcting Codes*, 2nd Edition 1972, MIT-Press, ISBN 0-262-16-039-0

[33] GUY E. CASTAGNOLI, *On the Minimum Distance of Long Cyclic Codes and Cyclic Redundancy-Check Codes*, 1989, Dissertation No. 8979 of ETH Zurich, Switzerland

[34] GUY E. CASTAGNOLI, STEFAN BRÄUER, and MARTIN HERRMANN, *Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bits*, June 1993, IEEE Transactions On Communications, Volume 41, No. 6

[35] PHILIP KOOPMAN, TRIDIB CHAKRAVARTY, *Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks*

[36] HEINZ GALL, THOMAS STEFFENS, KLAUS KEMP, *Anwendung der Bussysteme in der Anlagensicherheit der Chemie-Industrie,* TÜV Anlagentechnik GmbH (available in German only)

[37] DIETMAR LOCHMANN, *Digitale Nachrichtentechnik, Signale, Codierungen, Übertragungs-systeme, Netze*, 2002, Verlag Technik, Berlin, ISBN 3-341-01321-0 (available in German only)

[38] Ethernet POWERLINK Standardization Group, EPSG DS 301 V1.4.0, *Ethernet POWERLINK Communication Profile Specification, Draft Standard, Version 1.4.0*

[39] Ethernet POWERLINK Standardization Group, *Ethernet POWERLINK V1.0 Profile Specification, Draft Standard Proposal, Version 1.07b*

[40] FH Campus Wien, University of Applied Sciences / VISSE, *openSAFETY: Studie zur CRC-Qualität und Restfehlerwahrscheinlichkeit, Version 0.5, Juni 2011* (available in German only)

[41] PHILIP KOOPMAN, *32-Bit Cyclic Redundance Codes for Internet Applications*

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

RÉSEAUX DE COMMUNICATION INDUSTRIELS – PROFILS –

Partie 3-13: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 13

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.

L'IEC 61784-3-13 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels. Il s'agit d'une Norme internationale.

Cette troisième édition annule et remplace la deuxième édition parue en 2016. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- suppression du téléchargement amont/aval de bloc;
- ajout de SSDO avec préchargement;

- ajout de jeux de fonctionnalités SPDO;
- ajout de SPDO avec un compteur 40 bits;
- ajout du SCT dynamique de SPDO;
- ajout de jeux de paramètres supplémentaires;
- corrections et améliorations rédactionnelles.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
65C/1083/FDIS	65C/1087/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

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

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Le présent document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

Une liste de toutes les parties de la série IEC 61784-3, publiées sous le titre général *Réseaux de communication industriels – Profils – Bus de terrain de sécurité fonctionnelle*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu du présent document 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 au document recherché. A cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

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.

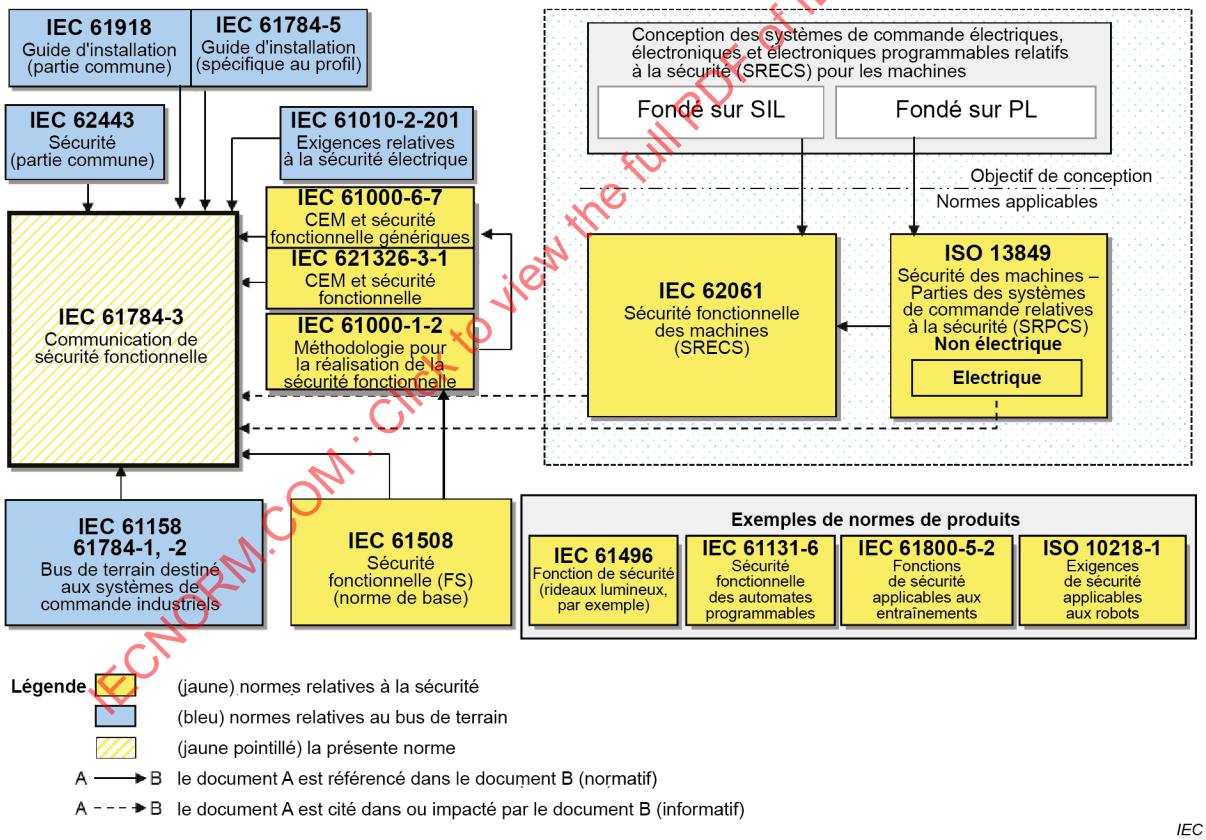
0 Introduction

0.1 Généralités

L'IEC 61158 (toutes les parties), relative aux bus de terrain, ainsi que ses normes associées IEC 61784-1 et IEC 61784-2, définissent un ensemble de protocoles de communication qui assurent la commande répartie d'applications automatisées. La technologie de bus de terrain est désormais reconnue et bien éprouvée. Les améliorations des bus de terrain se poursuivent; elles couvrent des applications pour des domaines comme les applications en temps réel relatives à la sécurité.

La série IEC 61784-3 (toutes les parties) explique les principes pertinents pour les communications de sécurité fonctionnelle en référence à l'IEC 61508 (toutes les parties) et spécifie plusieurs couches de communication de sécurité (profils et protocoles correspondants) qui reposent sur les profils de communication et les couches de protocole de l'IEC 61784-1, l'IEC 61784-2 et l'IEC 61158 (toutes les parties). Elle ne couvre pas les aspects relatifs à la sécurité électrique et à la sécurité intrinsèque. Elle ne couvre pas non plus les aspects relatifs à la sûreté et ne prévoit aucune exigence en matière de sûreté.

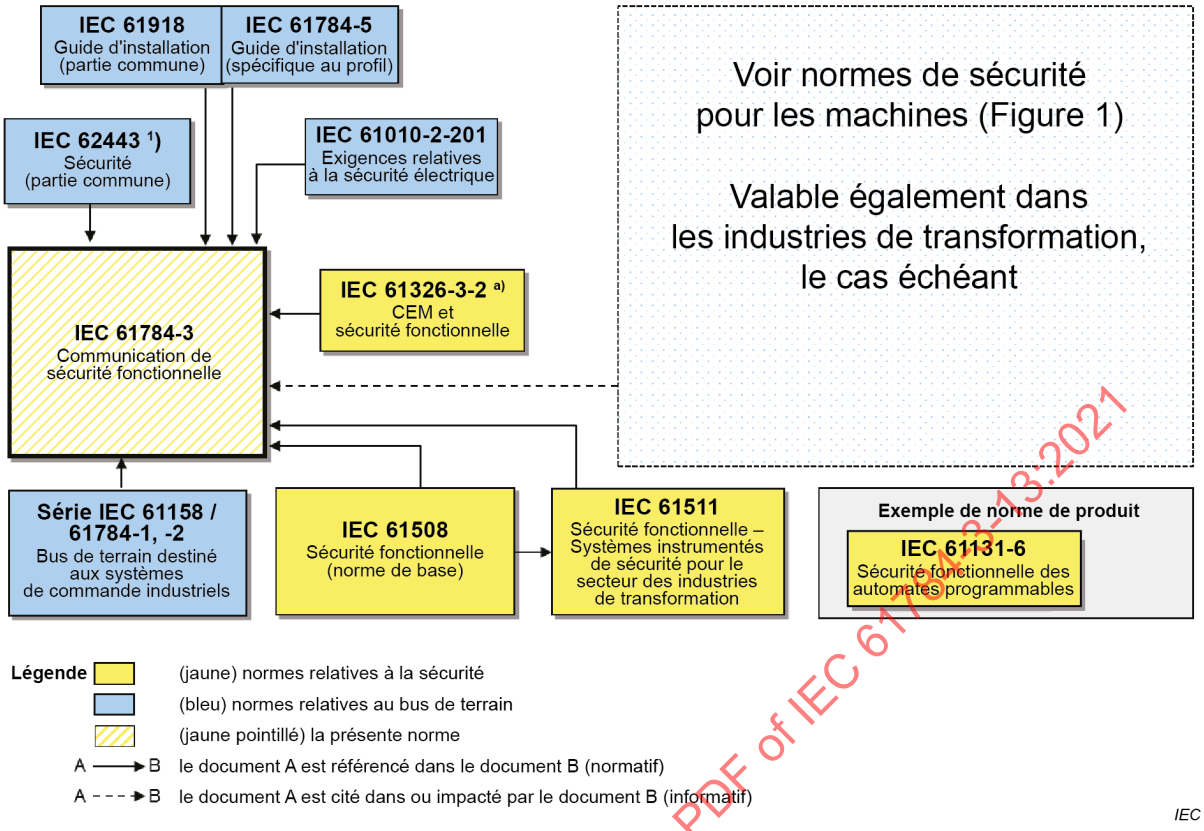
La Figure 1 représente les relations entre l'IEC 61784-3 (toutes les parties) et les normes pertinentes relatives à la sécurité et aux bus de terrain dans un environnement de machines.



NOTE L'IEC 62061 spécifie la relation entre PL (Catégorie) et SIL.

Figure 1 – Relations entre l'IEC 61784-3 et d'autres normes (machines)

La Figure 2 représente les relations entre l'IEC 61784-3 (toutes les parties) et les normes pertinentes relatives à la sécurité et aux bus de terrain dans un environnement de processus.



[a] Pour les environnements électromagnétiques spécifiés; sinon, l'IEC 61326-3-1 ou l'IEC 61000-6-7 s'applique.

Figure 2 – Relations entre l'IEC 61784-3 et d'autres normes (processus)

Les couches de communication de sécurité mises en œuvre dans le cadre de systèmes relatifs à la sécurité conformément à l'IEC 61508 (toutes les parties) assurent la confiance nécessaire à accorder à la transmission de messages (informations) entre plusieurs participants sur un bus de terrain dans un système relatif à la sécurité ou une fiabilité suffisante dans le comportement de sécurité en cas d'erreurs ou de défaillances du bus de terrain.

Les couches de communication de sécurité spécifiées dans l'IEC 61784-3 (toutes les parties) permettent de s'assurer qu'un bus de terrain peut être utilisé dans des applications qui nécessitent une sécurité fonctionnelle jusqu'au niveau d'intégrité de sécurité (SIL) spécifié par son profil de communication de sécurité fonctionnelle correspondant.

La revendication du SIL qui en résulte pour un système dépend de la mise en œuvre du profil de communication de sécurité fonctionnelle (FSCP) retenu au sein du système (la mise en œuvre du profil de communication de sécurité fonctionnelle dans un appareil normal ne suffit pas à le qualifier d'appareil de sécurité).

L'IEC 61784-3 (toutes les parties) décrit:

- les principes de base de la mise en œuvre des exigences de l'IEC 61508 (toutes les parties) pour les communications de données relatives à la sécurité, y compris les anomalies de transmission potentielles, les mesures correctives et des considérations relatives à l'intégrité des données;
- les profils de communication de sécurité fonctionnelle pour plusieurs familles de profils de communication dans l'IEC 61784-1 et l'IEC 61784-2, y compris les extensions de la couche de sécurité aux sections relatives au service et aux protocoles de communication de l'IEC 61158 (toutes les parties).

0.2 Déclaration de brevets

La Commission Electrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation de brevets intéressant les profils de communication de sécurité fonctionnelle pour la famille 13. L'IEC ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété.

Le détenteur de ces droits de propriété a donné l'assurance à l'IEC qu'il consent à négocier des licences avec des demandeurs du monde entier à des termes et conditions raisonnables et non discriminatoires. A ce propos, la déclaration du détenteur des droits de propriété est enregistrée à l'IEC. Des informations peuvent être obtenues dans la base de données des droits de propriété, disponible à l'adresse suivante: http://patents.iec.ch.

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux figurant dans la base de données des brevets. L'IEC ne saurait être tenue pour responsable de l'identification de ces droits de propriété en tout ou partie.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – PROFILS –

Partie 3-13: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 13

1 Domaine d'application

La présente partie de la série IEC 61784-3 (toutes les parties) spécifie une couche de communication de sécurité (services et protocole) qui repose sur la CPF 13 de l'IEC 61784-2 et le type 13 de l'IEC 61158. Elle identifie les principes applicables aux communications de sécurité fonctionnelle définies dans l'IEC 61784-3, qui correspondent à cette couche de communication de sécurité. Cette couche de communication de sécurité est destinée à être mise en œuvre uniquement sur les appareils de sécurité.

NOTE 1 Elle ne couvre pas les aspects relatifs à la sécurité électrique et à la sécurité intrinsèque. La sécurité électrique concerne les dangers tels que les chocs électriques. La sécurité intrinsèque concerne les dangers associés aux atmosphères explosibles.

Le présent document définit les mécanismes de transmission des messages relatifs à la sécurité entre les participants d'un réseau réparti, en utilisant la technologie de bus de terrain conformément aux exigences de la série IEC 61508 (toutes les parties)[1] concernant la sécurité fonctionnelle. Ces mécanismes peuvent être utilisés dans différentes applications industrielles, par exemple la commande de processus, l'usinage automatique et les machines.

Le présent document fournit des lignes directrices aux développeurs, ainsi qu'aux évaluateurs d'appareils et de systèmes conformes.

NOTE 2 La revendication du SIL qui en résulte pour un système dépend de la mise en œuvre du profil de communication de sécurité fonctionnelle retenu au sein du système (la mise en œuvre d'un profil de communication de sécurité fonctionnelle conforme au présent document dans un appareil normal ne suffit pas à le qualifier d'appareil de sécurité).

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences 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 programmes – Partie 3: Langages de programmation*

IEC 61158-3-13, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 3-13: Définition des services de la couche liaison de données – Eléments de type 13*

IEC 61158-4-13, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 4-13: Spécification du protocole de couche de liaison de données – Eléments de type 13*

IEC 61158-5-13, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-13: Définition des services de la couche application – Eléments de type 13*

[1] Dans les pages suivantes du présent document, "IEC 61508" remplace "IEC 61508 (toutes les parties)".

IEC 61158-6-13, *Réseaux de communication industriels – Spécification des bus de terrain – Partie 6-13: Spécification du protocole de la couche application – Eléments de type 13*

IEC 61508 (toutes les parties), *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité*

IEC 61784-2, *Réseaux de communication industriels – Profils – Partie 2: Profils de bus de terrain supplémentaires pour les réseaux en temps réel fondés sur l'ISO/IEC/IEEE 8802-3*

IEC 61784-3:2021, *Réseaux de communication industriels – Profils – Partie 3: Bus de terrain de sécurité fonctionnelle – Règles générales et définitions de profils*

IEC 61784-5-13, *Réseaux de communication industriels – Profils – Partie 5-13: Installation des bus de terrain – Profils d'installation pour le CPF 13*

IEC 61918, *Réseaux de communication industriels – Installation de réseaux de communication dans des locaux industriels*

ISO/IEC 19501, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2* (disponible en anglais seulement)

3 Termes, définitions, symboles, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 61784-3 ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

NOTE L'italique est utilisé dans les définitions pour mettre en évidence les termes définis en 3.1.

3.1.1 Termes et définitions communs

NOTE Ces termes et définitions communs sont issus de l'IEC 61784-3:2021.

3.1.1.1
probabilité d'erreurs sur les éléments binaires
Pe
probabilité de réception d'un bit donné avec la valeur incorrecte

3.1.1.2
canal noir
système de communication défini qui contient un ou plusieurs éléments sans preuve de conception ou de validation conformément à l'IEC 61508

Note 1 à l'article: Cette définition étend la signification habituelle du canal pour inclure le système qui le contient.

3.1.1.3
système de communication fermé
nombre fixe ou nombre maximal fixe d'éléments reliés par un *système de communication* dont les propriétés sont connues et fixées et où le *risque* d'accès non autorisé est considéré comme négligeable

[SOURCE: IEC 62280:2014, 3.1.6, modifié – "transmission" remplacé par "communication"]

3.1.1.4
canal de communication
connexion logique entre deux points limites d'un *système de communication*

3.1.1.5
système de communication
ensemble de matériels, de logiciels et de supports de propagation qui permet la transmission de *messages* (ISO/IEC 7498-1, couche d'application) d'une application à une autre

3.1.1.6
connexion
liaison logique entre deux objets d'application au sein du même appareil ou d'appareils différents

3.1.1.7
contrôle de redondance cyclique
CRC
<valeur> donnée redondante déduite, et enregistrée ou transmise simultanément, d'un bloc de données afin de détecter toute corruption des données

<méthode> procédure utilisée pour calculer les données redondantes

Note 1 à l'article: Les termes "code CRC" et "signature CRC", ainsi que les étiquettes comme CRC1, CRC2, peuvent également être utilisés dans le présent document pour se référer aux données redondantes.

Note 2 à l'article: Voir également [31], [32][2].

3.1.1.8
système de communication défini
canal défini
nombre fixe ou nombre maximal fixe d'éléments reliés par un système de communication à bus de terrain, dont les propriétés sont connues et fixées, par exemple les conditions d'installation, l'immunité électromagnétique, les éléments (actifs) de réseau industriel, et où le risque d'accès non autorisé est réduit à un niveau toléré conformément au modèle de cycle de vie de l'IEC 62443 (toutes les parties), en utilisant par exemple des zones et des conduits

3.1.1.9
DLPDU
DÉCONSEILLÉ: trame
unité de données de protocole de liaison de données

Note 1 à l'article: L'abréviation "DLPDU" est dérivée du terme anglais développé correspondant "data link protocol data unit".

2 Les chiffres entre crochets renvoient à la bibliographie.

3.1.1.10
erreur
écart ou discordance entre une valeur ou une condition calculée, observée ou mesurée et la valeur ou la condition vraie, prescrite ou théoriquement correcte

Note 1 à l'article: Les erreurs peuvent être causées par des erreurs de conception du matériel/logiciel et/ou des informations altérées du fait d'un brouillage électromagnétique et/ou autres effets.

Note 2 à l'article: Les erreurs ne produisent pas nécessairement une *défaillance* ou une *anomalie*.

[SOURCE: IEC 61508-4:2010, 3.6.11, modifié – notes ajoutées]

3.1.1.11
défaillance
cessation de l'aptitude d'une unité fonctionnelle à accomplir une fonction requise ou à fonctionner comme prévu

Note 1 à l'article: Une défaillance peut être causée par une *erreur* (problème de conception matérielle/logicielle ou rupture de message, par exemple).

[SOURCE: IEC 61508-4:2010, 3.6.4, modifié – notes et figures remplacées]

3.1.1.12
anomalie
condition anormale qui peut entraîner une réduction de capacité ou la perte de capacité d'une unité fonctionnelle à accomplir une fonction requise

Note 1 à l'article: L'IEC 60050-191:1990, 191-05-01, définit le terme "fault" (en français "panne") comme un état d'inaptitude à accomplir une fonction requise, en excluant l'inaptitude due à la maintenance préventive, à d'autres actions programmées ou à un manque de ressources extérieures.

[SOURCE: IEC 61508-4:2010, 3.6.1, modifié – référence à la figure supprimée]

3.1.1.13
bus de terrain
système de communication fondé sur le transfert de données en série et utilisé dans des applications d'automatisation industrielle ou de commande de processus

3.1.1.14
système de bus de terrain
système qui utilise un *bus de terrain* avec des appareils reliés

3.1.1.15
fonction de hachage
fonction (mathématique) de mapping des valeurs d'un ensemble (éventuellement) très grand de valeurs en une plage de valeurs (habituellement) plus petite

Note 1 à l'article: Les fonctions de hachage peuvent être utilisées pour déterminer l'altération des données.

Note 2 à l'article: Les fonctions de hachage communes incluent la parité, la somme de contrôle ou le CRC.

3.1.1.16
danger
source potentielle de dommage

Note 1 à l'article: Ce terme comprend le danger sur des personnes survenant dans un laps de temps très court (par exemple, feu et explosion), mais aussi le danger à long terme sur la santé d'une personne (par exemple, dégagement d'une substance toxique).

[SOURCE: IEC 61508-4:2010, 3.1.2, et Guide ISO/IEC 51:2014, définition 3.2]

3.1.1.17
maître
entité de communication capable d'initier et de programmer des activités de communication effectuées par d'autres stations qui peuvent être des maîtres ou des esclaves

3.1.1.18
message
<théorie de l'information et théorie des communications> suite ordonnée de caractères (généralement des octets) destinée à communiquer des informations

[SOURCE: ISO/IEC 2382:2015, 2123205, modifié – insertion de "(généralement des octets)", suppression des notes et de la source]

3.1.1.19
niveau de performance
PL
niveau discret d'aptitude de parties relatives à la sécurité à réaliser une *fonction de sécurité* dans des conditions prévisibles

Note 1 à l'article: L'abréviation "PL" est dérivée du terme anglais développé correspondant "performance level".

[SOURCE: ISO 13849-1:2015, 3.1.23]

3.1.1.20
redondance
existence de plusieurs moyens pour accomplir une fonction requise ou pour représenter des informations

[SOURCE: IEC 61508-4:2010, 3.4.6, modifié – exemple et notes supprimés]

3.1.1.21
taux d'erreurs résiduelles
taux statistique de défaut de détection d'*erreurs* par les *mesures de sécurité* SCL

3.1.1.22
risque
combinaison de la probabilité d'un dommage et de la gravité de ce dommage

Note 1 à l'article: Pour plus d'informations sur ce concept, voir Annexe A de l'IEC 61508-5:2010.

[SOURCE: IEC 61508-4:2010, 3.1.6, et Guide ISO/IEC 51:2014, définition 3.9, modifiée – note différente]

3.1.1.23
canal de communication de sécurité
canal de communication qui commence au sommet de la SCL de la source et qui se termine au sommet de la SCL du collecteur

Note 1 à l'article: Le canal peut être modélisé sous la forme de deux SCL reliées par un *canal noir*, un *système de communication défini* ou un *canal défini*.

3.1.1.24
couche de communication de sécurité
SCL
couche de communication située au-dessus de la FAL qui comprend toutes les mesures supplémentaires nécessaires qui permettent d'assurer la transmission de données en toute sécurité conformément aux exigences de l'IEC 61508

Note 1 à l'article: L'abréviation "SCL" est dérivée du terme anglais développé correspondant "safety communication layer".

3.1.1.25
données de sécurité
données transmises par un réseau de sécurité qui utilise un protocole de sécurité

Note 1 à l'article: La *couche de communication de sécurité* n'assure pas la sécurité des données proprement dites, mais uniquement la transmission en toute sécurité de ces dernières.

3.1.1.26
appareil de sécurité
appareil conçu conformément à l'IEC 61508 et qui met en œuvre le profil de communication de sécurité fonctionnelle

3.1.1.27
fonction de sécurité
fonction à réaliser par un système E/E/PE *relatif à la sécurité* ou par un dispositif externe de réduction de *risque*, prévue pour assurer ou maintenir un état de sécurité de l'EUC par rapport à un événement dangereux spécifique

[SOURCE: IEC 61508-4:2010, 3.5.1, modifié – références et exemples supprimés]

3.1.1.28
temps de réponse de la fonction de sécurité
temps écoulé dans le cas le plus défavorable à la suite de l'activation d'un capteur de sécurité relié à un *bus de terrain*, avant d'atteindre l'état de sécurité correspondant de ses actionneurs de sécurité, du fait d'*erreurs* ou de *défaillances* dans la *fonction de sécurité*

Note 1 à l'article: Ce concept, introduit dans l'IEC 61784-3:2021, 5.2.4, est traité par les profils de communication de sécurité fonctionnelle définis dans le présent document.

3.1.1.29
niveau d'intégrité de sécurité
SIL
niveau discret (parmi quatre possibles) correspondant à une gamme de valeurs d'intégrité de sécurité où le niveau 4 d'intégrité de sécurité possède le plus haut degré d'intégrité et le niveau 1 possède le plus bas

Note 1 à l'article: Les objectifs chiffrés de *défaillance* (voir l'IEC 61508-4:2010, 3.5.17) pour les quatre niveaux d'intégrité de sécurité sont indiqués dans les Tableaux 2 et 3 de l'IEC 61508-1:2010.

Note 2 à l'article: Les niveaux d'intégrité de sécurité sont utilisés pour spécifier les exigences concernant l'intégrité de sécurité des *fonctions de sécurité* à allouer aux *systèmes E/E/PE relatifs à la sécurité*.

Note 3 à l'article: Un niveau d'intégrité de sécurité (SIL) ne constitue pas une propriété d'un système, sous-système, élément ou composant. L'interprétation correcte de l'expression "*système relatif à la sécurité* à SILn" (où n est 1, 2, 3 ou 4) signifie que le système est potentiellement capable de prendre en charge les *fonctions de sécurité* avec un niveau d'intégrité de sécurité jusqu'à n.

Note 4 à l'article: L'abréviation "SIL" est dérivée du terme anglais développé correspondant "safety integrity level".

[SOURCE: IEC 61508-4:2010, 3.5.8]

3.1.1.30
mesure de sécurité
mesure qui permet de contrôler les *erreurs* de communication éventuelles, qui est conçue et mise en œuvre conformément aux exigences de l'IEC 61508

Note 1 à l'article: Dans la pratique, plusieurs mesures de sécurité sont combinées pour atteindre le *niveau d'intégrité de sécurité* exigé.

Note 2 à l'article: Les *erreurs* de communication et les mesures de sécurité associées sont décrites dans l'IEC 61784-3:2021, 5.3 et 5.4.

3.1.1.31
PDU de sécurité
SPDU
PDU transféré par le biais du *canal de communication de sécurité*

Note 1 à l'article: Le SPDU peut comporter plusieurs exemplaires des *données de sécurité* qui utilisent des structures de codage et des *fonctions de hachage* différentes, associées à des parties explicites de protections supplémentaires, par exemple une clé, un nombre de séquences ou un mécanisme de *datation*.

Note 2 à l'article: Les SCL redondantes peuvent fournir deux versions différentes du SPDU en vue de son insertion dans des champs séparés de la *trame de bus de terrain*.

Note 3 à l'article: L'abréviation "PDU" est dérivée du terme anglais développé correspondant "protocol data unit".

Note 4 à l'article: L'abréviation "SPDU" est dérivée du terme anglais développé correspondant "safety protocol data unit".

3.1.1.32
application relative à la sécurité
programmes conçus conformément à l'IEC 61508 pour satisfaire aux exigences SIL de l'application

3.1.1.33
système relatif à la sécurité
système qui exécute les *fonctions de sécurité* conformément à l'IEC 61508

3.1.1.34
esclave
entité de communication capable de recevoir des *messages* et de les envoyer en réponse à une autre entité de communication qui peut être *maître* ou esclave, mais pas d'initier des activités de communication

3.1.1.35
datation (horodatage)
information temporelle incluse dans un *message*

3.1.2 CPF 13: Termes et définitions supplémentaires

3.1.2.1
taux d'erreurs sur les bits
probabilité d'une interprétation erronée des bits due au bruit électrique

3.1.2.2
informations de fournisseur d'appareil
DVI
informations d'identification d'un *nœud de sécurité*

Note 1 à l'article: L'abréviation "DVI" est dérivée du terme anglais développé correspondant "device vendor information".

3.1.2.3
adresse de sécurité
SADR
adresse d'un *nœud de sécurité* dans un *domaine de sécurité*

Note 1 à l'article: L'abréviation "SADR" est dérivée du terme anglais développé correspondant "safety address".

3.1.2.4
gestionnaire de configuration de sécurité
SCM
service chargé de gérer les services relatifs à la sécurité

Note 1 à l'article: Les services de sécurité gérés sont, par exemple, la configuration, le paramétrage, etc.

Note 2 à l'article: L'abréviation "SCM" est dérivée du terme anglais développé correspondant "safety configuration manager".

3.1.2.5
nœud de sécurité
SN
nœud qui possède une *fonction de sécurité*

Note 1 à l'article: L'abréviation "SN" est dérivée du terme anglais développé correspondant "safety node".

3.1.2.6
Domaine de sécurité
SD
espace d'adresse qui comporte jusqu'à 1 023 *nœuds de sécurité*

Note 1 à l'article: L'abréviation "SD" est dérivée du terme anglais développé correspondant "safety domain".

3.1.2.7
Passerelle de domaine de sécurité
SDG
passerelle qui permet le partage de données entre différents *domaines de sécurité*

Note 1 à l'article: L'abréviation "SDG" est dérivée du terme anglais développé correspondant "safety domain gateway".

3.1.2.8
Numéro de domaine de sécurité
SDN
identifiant d'un *domaine de sécurité*

Note 1 à l'article: L'abréviation "SDN" est dérivée du terme anglais développé correspondant "safety domain number".

3.1.2.9
gestion de réseau de sécurité
SNMT
services qui permettent de gérer la couche de sécurité

Note 1 à l'article: L'abréviation "SNMT" est dérivée du terme anglais développé correspondant "safety network management".

3.1.2.10
dictionnaire d'objets de sécurité
SOD
référentiel de tous les objets de données accessibles au moyen des services du *bus de terrain* de sécurité fonctionnelle

Note 1 à l'article: L'abréviation "SOD" est dérivée du terme anglais développé correspondant "safety object dictionary".

3.1.2.11
objet de données de processus de sécurité
SPDO
objet qui permet l'échange de données de processus entre différents *nœuds de sécurité*

Note 1 à l'article: L'abréviation "SPDO" est dérivée du terme anglais développé correspondant "safety process data object".

3.1.2.12
objet de données de service de sécurité
SSDO
objet qui permet l'échange de données de service entre différents *nœuds de sécurité*

Note 1 à l'article: L'abréviation "SSDO" est dérivée du terme anglais développé correspondant "safety service data object".

3.1.2.13
initialisation
état de communication initial d'un *nœud de sécurité*

3.1.2.14
opérationnel
état de communication d'un *nœud de sécurité* dans lequel toutes les fonctions de sécurité fonctionnelle opèrent correctement

3.1.2.15
préopérationnel
état de communication d'un *nœud de sécurité* dans lequel toutes les sorties d'application sont à l'état de sécurité

3.1.2.16
identifiant unique d'appareil
UDID
identifiant unique et universel d'un appareil donné

Note 1 à l'article: L'abréviation "UDID" est dérivée du terme anglais développé correspondant "unique device identification".

3.2 Symboles et abréviations

3.2.1 Symboles et abréviations communs

CP (Communication Profile)	Profil de communication	[IEC 61784-1]
CPF (Communication Profile Family)	Famille de profils de communication	[IEC 61784-1]
CRC	Contrôle de redondance cyclique	
DLL (Data Link Layer)	Couche de liaison de données	[ISO/IEC 7498-1]
DLPDU (Data Link Protocol Data Unit)	Unité de données de protocole de liaison de données	
CEM	Compatibilité électromagnétique	
EUC (Equipment Under Control)	Equipement commandé	[IEC 61508-4:2010]
E/E/PE (Electrical/Electronic/Programmable Electronic)	Electrique/électronique/électronique programmable	[IEC 61508-4:2010]
FAL (Fieldbus Application Layer)	Couche d'application de bus de terrain	[IEC 61158-5 (toutes les parties)]
FS (Functional Safety)	Sécurité fonctionnelle	
FSCP (Functional Safety Communication Profile)	Profil de communication de sécurité fonctionnelle	
MTBF (Mean Time Between Failures)	Durée moyenne de bon fonctionnement	
MTTF (Mean Time To Failure)	Durée moyenne de fonctionnement avant défaillance	
PDU (Protocol Data Unit)	Unité de données de protocole	[ISO/IEC 7498-1]
PFH (Average frequency of dangerous failure [h-1] per hour)	Fréquence moyenne de défaillance dangereuse [h-1] par heure	[IEC 61508-4:2010]
PhL (Physical Layer)	Couche physique	[ISO/IEC 7498-1]
PL (Performance Level)	Niveau de performance	[ISO 13849-1]
PLC (Programmable Logic Controller)	Automate programmable	
SCL (Safety Communication Layer)	Couche de communication de sécurité	
SIL (Safety Integrity Level)	Niveau d'intégrité de sécurité	[IEC 61508-4:2010]
SPDU (Safety PDU)	PDU de sécurité	

3.2.2 CPF 13: Symboles et abréviations supplémentaires

ACM (Automatic Configuration Mode)	Mode de configuration automatique	
ADR (Address information)	Informations d'adresse (source ou destination)	
APDU (Application Protocol Data Unit)	Unité de données de protocole d'application	[ISO/IEC 7498-1]
CN (Controlled Node)	Nœud commandé	[IEC 61158-3-13]
CT (Consecutive Time field)	Champ temps consécutif	
DBx (Data Octet)	Octet de données	
DSADR (Destination SADR)	SADR de destination	
DVI (Device Vendor Information)	Informations de fournisseur d'appareil	
ID (PDU Identification)	Identifiant de PDU	
LE (Length field)	Champ longueur	
LSB (Least Significant Octet)	Octet de poids faible	
MAC (Medium Access Layer)	Couche d'accès au support	

MCM (Manual Configuration Mode)	Mode de configuration manuel	
MN (Managing Node)	Nœud gestionnaire	[IEC 61158-3-13]
MSB (Most Significant Octet)	Octet de poids fort	
PDO (Process Data Object)	Objet de données de processus	[IEC 61158-3-13]
RxSPDO (Receive Process Data Objects)	Objet de données de processus de réception	
SADR (Safety Address)	Adresse de sécurité	
SCM (Safety Configuration Manager)	Gestionnaire de configuration de sécurité	
SCT (Safety Control Time)	Temps de contrôle de sécurité	
SD (Safety Domain)	Domaine de sécurité	
SDG (Safety Domain Gateway)	Passerelle de domaine de sécurité	
SDN (Safety Domain Number)	Numéro de domaine de sécurité	
SDO (Service Data Object)	Objet de données de service	[IEC 61158-3-13]
SN (Safety Node)	Nœud de sécurité	
SNMT (Safety Network Management)	Gestion de réseau de sécurité	
SOD (Safety Object Dictionary)	Dictionnaire d'objets de sécurité	
SPDO (Safety Process Data Object)	Objet de données de processus de sécurité	
SPST (Safety Parameterization Software Tool)	Outil logiciel de paramétrage de sécurité	
SSDO (Safety Service Data Object)	Objet de données de service de sécurité	
SSADR (Source SADR)	SADR d'origine	
TADR (Additional SADR for timing information)	SADR supplémentaire pour informations temporelles	
TR (Time Request Distinctive Number)	Numéro distinctif de demande de temps	
TReq (Time Synchronization Request)	Demande de synchronisation temporelle	
TRes (Time Synchronization Response)	Réponse de synchronisation temporelle	
TÜV	Technischer Überwachungsverein	
TxSPDO (Safety Transmit Process Data Objects)	Objets de données de processus de transmission de sécurité	
UDID (Unique Device Identification)	Identifiant unique d'appareil	

3.3 Conventions

3.3.1 Valeurs hexadécimales

Certaines valeurs dans le présent document sont spécifiées en notation hexadécimale. Dans ce cas, la lettre "h" doit être ajoutée à la valeur proprement dite. Les caractères 1, 2, 3, 4, 5, 6, 8, 9, 0, a, b, c, d, e, f, A, B, C, D, E et F doivent être utilisés. Toutes les valeurs en notation hexadécimale doivent décrire des octets complets; par conséquent, une représentation par une valeur hexadécimale est toujours constituée d'un nombre pair de caractères.

EXEMPLE 1 12ABh, 05h, 1234ABCDh

Pour décrire des données dans lesquelles une suite d'octets doit être explicitement définie – afin d'être indépendante de la représentation exacte de la mémoire (gros-boutiste ou petit-boutiste) – une notation en séquence hexadécimale doit être utilisée. Chaque octet est représenté par une valeur hexadécimale à deux chiffres (sans le suffixe h); les représentations à un seul octet doivent être séparées par un tiret (-).

EXEMPLE 2 00-00-00-00-00-00, AC-3E

3.3.2 Valeurs binaires

Certaines valeurs dans le présent document sont spécifiées en notation binaire. Dans ce cas, la lettre "b" doit être ajoutée à la valeur proprement dite. Les caractères 1 et 0 doivent être utilisés.

EXEMPLE 11001010b

3.3.3 Chiffres en caractères génériques

Si des parties de valeurs constantes sont à ignorer (don't care) ou qu'elles ont une signification supplémentaire qui ne relève pas du texte descriptif, celles-ci doivent être représentées par un x à la place du chiffre correspondant.

EXEMPLE 12xBh, 1x0xx010b, 123x45

3.3.4 Diagrammes

Le cas échéant, dans l'ensemble du présent document, les notations graphiques conformes à l'ISO/IEC 19501 sont utilisées.

4 Présentation générale de FSCP 13/1 (openSAFETY™)

4.1 Profil de communication de sécurité fonctionnelle 13/1

La famille de profils de communication 13 (souvent appelée Ethernet POWERLINK™[3]) définit les profils de communication qui reposent sur l'IEC 61158-3-13, l'IEC 61158-4-13, l'IEC 61158-5-13 et l'IEC 61158-6-13.

Le profil de base CP 13/1 est défini dans l'IEC 61784-2. Le profil de communication de sécurité fonctionnelle FSCP 13/1 CPF 13 (openSAFETY™) repose sur les profils de base de la famille CPF 13 spécifiés dans l'IEC 61784-2 et sur les spécifications de la couche de communication de sécurité définies dans le présent document.

4.2 Aperçu technique

Le FSCP 13/1 définit un protocole de sécurité ouvert pour des communications Ethernet jusqu'à la microseconde.

[3] Ethernet POWERLINK™ et openSAFETY™ sont des appellations commerciales de l'organisme à but non lucratif Ethernet POWERLINK™ Standardization Group (EPSG). Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande le détenteur de la marque ou de l'un de ses produits. La conformité au présent document n'exige pas d'utiliser les appellations commerciales Ethernet POWERLINK™ ou openSAFETY™. L'utilisation des appellations commerciales Ethernet POWERLINK™ ou openSAFETY™ exige l'autorisation d'Ethernet POWERLINK Standardization Group (EPSG) et la conformité aux conditions d'utilisation (essais et validation).

Les services et le protocole FSCP 13/1 spécifient une communication de données de sécurité entre nœuds de sécurité (SN). La technologie de protocole FSCP 13/1 est conçue pour assurer une fonction de sécurité de niveau SIL 3 conformément à l'IEC 61508. La configuration du réseau est réalisée au moyen du gestionnaire de configuration de sécurité (SCM). Des services de gestion de réseau de sécurité (SNMT) sont utilisés pour le diagnostic au démarrage et dans l'environnement d'exécution des réseaux FSCP 13/1. Le dictionnaire d'objets de sécurité (SOD) est une base de données fournie par chaque SN qui contient des données de configuration et des données utilisateur. Les objets de données de service de sécurité (SSDO) sont utilisés pour l'échange de données spontanées. Les objets de données de processus de sécurité (SPDO) fournissent des services d'échange de données synchrones.

La communication de données synchrones entre SN est modélisée conformément au modèle éditeur-abonné (voir Figure 3), tandis que la communication des données spontanées utilise le modèle client-serveur (voir Figure 4).

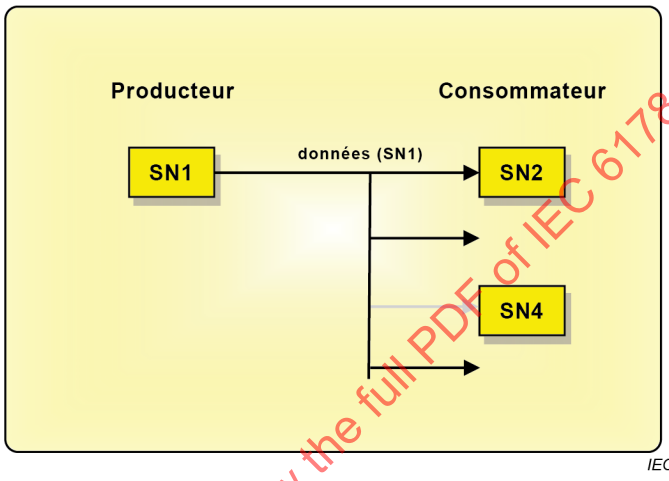


Figure 3 – Exemple de relation producteur/consommateur

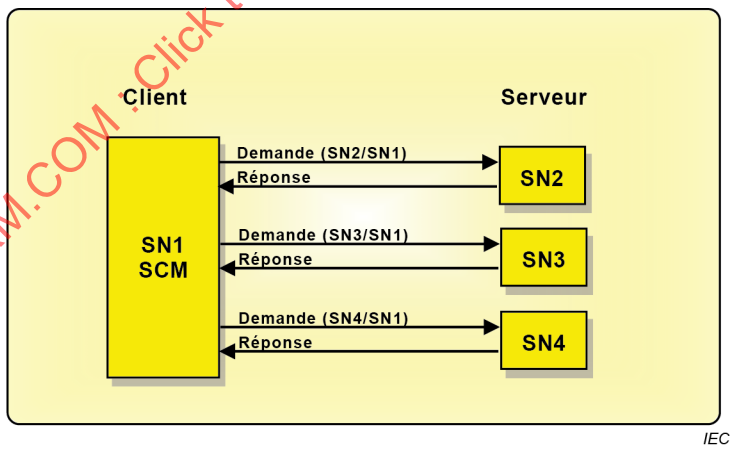


Figure 4 – Exemple de relation client/serveur

5 Généralités

5.1 Documents externes de spécifications applicables au profil

Les documents [38] et [39] peuvent aider à comprendre les concepts du FSCP 13/1.

5.2 Exigences fonctionnelles de sécurité

Les exigences suivantes ont été utilisées pour l'élaboration du FSCP 13/1 et doivent s'appliquer à la conception des appareils qui mettent en œuvre le protocole FSCP 13/1:

- le protocole FSCP 13/1 doit prendre en charge le niveau d'intégrité de sécurité 3 (SIL 3) conformément à l'IEC 61508;
- les exigences fondamentales pour le développement du FSCP 13/1 sont spécifiées dans l'IEC 61784-3;
- le FSCP 13/1 utilise des services définis dans l'IEC 61158-5-13;
- le protocole FSCP 13/1 repose sur une approche "canal noir"; aucune dépendance de sécurité ne s'applique au profil de communication CPF 13;
- les mises en œuvre du FSCP 13/1 doivent être conformes à l'IEC 61508;
- sauf spécification contraire dans le présent document, les exigences de la CPF 13 pour la sécurité doivent demeurer inchangées;
- les communications de sécurité et les communications normales doivent être indépendantes; Cependant, les appareils normalisés et les appareils FSCP 13/1 doivent pouvoir utiliser le même canal de communication;
- il doit être possible d'utiliser des appareils d'infrastructure de réseaux non relatifs à la sécurité pour construire un réseau FSCP 13/1;
- un domaine de sécurité donné doit prendre en charge jusqu'à 1 023 appareils de sécurité;
- jusqu'à 1 023 domaines de sécurité doivent pouvoir être pris en charge;
- la largeur de bande utilisée par les messages de sécurité doit être aussi faible que possible; les PDU de sécurité doivent être optimisés pour les appareils qui utilisent moins de données (par exemple des E/S numériques); et
- sauf spécification contraire dans le présent document, l'état de sécurité de toute valeur doit être zéro (0).

5.3 Mesures de sécurité

Le profil FSCP 13/1 doit utiliser les mesures de sécurité indiquées dans le Tableau 1 pour la détection d'erreurs de communication dans des données cycliques transmises dans des SPDO (voir 7.2). Les mesures de sécurité doivent être traitées et surveillées au sein de chaque appareil FSCP 13/1.

Tableau 1 – Erreurs de communication et mesures de détection (cycliques)

Erreurs de communication	Mesures de sécurité							
	Numéro de séquence	Horodatage	Délai d'attente	Authentification de connexion	Message de réaction	Assurance d'intégrité des données	Redondance avec contrevérification	Différents systèmes d'assurance d'intégrité des données
Corruption						X	X	
Répétition non prévue		X						
Séquence incorrecte		X						
Perte			X[a]					
Retard inacceptable		X	X					
Insertion			X[b]					
Déguisement			X			X	X	X
Adressage				X				
NOTE Les erreurs "perte" et "insertion" ne sont pas détectées par un horodatage. Une perte de données est détectée dans le délai maximal de réaction par un temporisateur de surveillance (watchdog). Si un message a été envoyé plusieurs fois (insertion), l'horodatage reste inchangé et l'appareil de réception ignore ce message.								
[a] Des PDU manquants doivent être détectés dans le délai maximal de réaction. [b] Un seul message doit être accepté au cours d'un créneau temporel défini.								

Le profil FSCP 13/1 doit utiliser les mesures de sécurité indiquées dans le Tableau 2 pour la détection d'erreurs de communication dans des données acycliques transmises dans des SSDO (voir 7.3) ou dans des SNMT (voir 7.4). Les mesures de sécurité doivent être traitées et surveillées au sein de chaque appareil FSCP 13/1.

Tableau 2 – Erreurs de communication et mesures de détection (acycliques)

Erreurs de communication	Mesures de sécurité							
	Numéro de séquence	Horodatage	Délai d'attente	Authentification de connexion	Message de réaction [e]	Assurance d'intégrité des données	Redondance avec contrevérification	Différents systèmes d'assurance d'intégrité des données
Corruption					X	X	X	
Répétition non prévue	X							
Séquence incorrecte	X							
Perte	X				X			
Retard inacceptable			X					
Insertion	X				X			
Déguisement						X	X	X
Adressage				X				
NOTE Les erreurs "perte" et "insertion" ne sont pas détectées par un horodatage. Une perte de données est détectée dans le délai maximal de réaction par un temporisateur de surveillance. Si un message a été envoyé plusieurs fois (insertion), l'horodatage reste inchangé et l'appareil de réception ignore ce message.								

Pour éviter des erreurs systématiques et stochastiques, les lignes directrices suivantes doivent être envisagées lors de l'élaboration et de la mise en œuvre du FSCP 13/1 (voir 7.1):

– Le producteur des données de sécurité doit:
  - générer un horodatage;

    NOTE Tout nouveau SPDU contient un nouvel horodatage.
  - générer des CRC; et
  - dupliquer le message de sécurité en fonction du PDU de sécurité utilisé.
– Le consommateur de données de message de sécurité doit:
  - vérifier la partie 1 du PDU avec le CRC;
  - vérifier la partie 2 du PDU avec le CRC;
  - comparer les zones des données des parties 1 et 2 du PDU bit par bit;
  - vérifier l'adresse recue;
  - réinitialiser le temporisateur de surveillance (compteur) interne si un nouveau PDU valide a été reçu;
  - si un PDU comporte le même horodatage que le dernier PDU reçu ou qu'un ancien PDU, le PDU doit être ignore;
  - si aucun PDU valide n'est reçu dans un délai défini (temps de contrôle de sécurité, voir SCT_U32, Tableau 158), l'appareil doit passer à l'état de sécurité; et
  - si le temporisateur de surveillance (compteur écoulé) présente un dépassement (absence de réinitialisation dans le délai maximal), l'appareil doit passer à l'état de sécurité.

5.4 Structure de la couche de communication de sécurité

Le protocole FSCP 13/1 est placé en couche supérieure sur un canal de communication normalisé, qui doit être considéré comme un canal noir. La Figure 5 représente la relation entre la couche de communication de sécurité et le canal de communication normalisé.

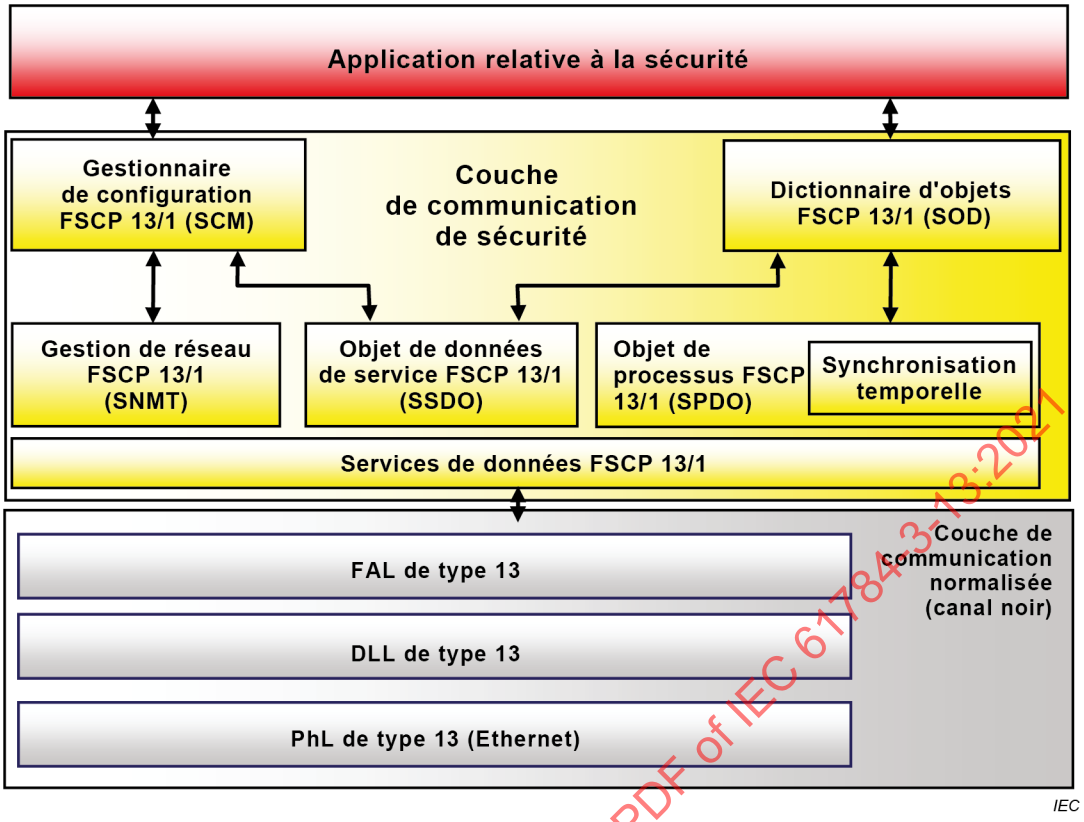


Figure 5 – Structure de la couche de communication

Les données de sécurité sont produites par l'entité d'application relative à la sécurité. Ces données sont encapsulées dans un message de sécurité par les couches de communication de sécurité et envoyées sur le canal de communication normalisé. La couche de communication de sécurité dans l'appareil de réception décode le message de sécurité et vérifie l'intégrité ainsi que l'actualité des données. Les données sont présentées à l'entité d'application relative à la sécurité. Si aucune donnée n'est reçue dans un créneau temporel défini, l'entité d'application est informée de cet événement, qui peut être utilisé pour mettre toutes les sorties à l'état de sécurité.

La couche de communication de sécurité assure:

– le transport des données de sécurité (intégrité et surveillance temporelle);
– l'amorçage du réseau;
– le diagnostic de l'environnement d'exécution;
– la mise en service;
– le téléchargement des programmes et micrologiciels;
– le paramétrage; et
– le remplacement de l'appareil.

Il est possible d'accéder à toutes ces fonctions et de les contrôler à partir du canal de communication de sécurité (voir Figure 6).

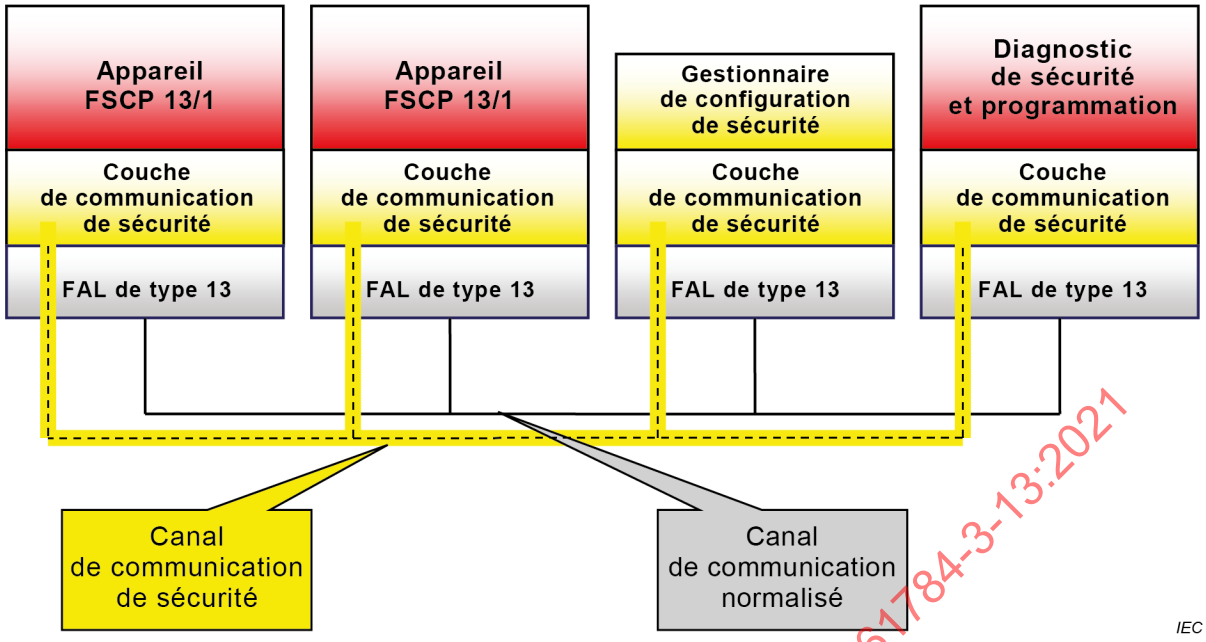


Figure 6 – Canal de communication de sécurité

5.5 Relations avec la FAL (et DLL, PhL)

5.5.1 Généralités

Le FSCP 13/1 doit utiliser les services définis dans l'IEC 61158-5-13.

5.5.2 Types de données

Les profils définis dans le présent document prennent en charge tous les types de données CPF 13 définis dans l'IEC 61158-5-13.

6 Services de la couche de communication de sécurité

6.1 Modélisation

6.1.1 Modèle de référence

6.1.1.1 Généralités

Une mise en œuvre conforme FSCP 13/1 doit fournir les objets d'application suivants:

- la gestion de réseau de sécurité (SNMT);
- le gestionnaire de configuration de sécurité (SCM);
- les objets de données de service de sécurité (SSDO);
- le dictionnaire d'objets de sécurité (SOD); et
- les objets de données de processus de sécurité (SPDO).

La Figure 5 représente les relations entre ces objets d'application.

L'utilisation d'une couche de communication sous-jacente doit être obligatoire.

6.1.1.2 Gestion de réseau de sécurité (SNMT)

Le SNMT doit fournir les services exigés pour que le gestionnaire de configuration de sécurité (SCM) configure et vérifie les nœuds de sécurité au sein du réseau.

6.1.1.3 Objets de données de service de sécurité (SSDO)

Le SSDO doit fournir les services exigés pour l'accès au dictionnaire d'objets de sécurité (SOD) et la prise en charge du SCM.

6.1.1.4 Objets de données de processus de sécurité (SPDO)

Le SPDO doit fournir les services exigés pour l'échange de données de processus de sécurité entre certaines entrées dans le SOD des nœuds communicants. Les services de synchronisation temporelle, qui font partie des SPDO, doivent vérifier les performances du réseau entre nœuds communicants.

6.1.1.5 Dictionnaire d'objets de sécurité (SOD)

Le SOD doit décrire le format ainsi que les services d'accès des données spécifiques à l'appareil.

6.1.1.6 Gestionnaire de configuration de sécurité (SCM)

Le SCM doit fournir les services exigés pour le démarrage, le paramétrage ainsi que les autres mesures ultérieures qui permettent d'assurer un fonctionnement sûr pendant toute la durée du cycle de vie de l'application.

6.1.2 Modèle de communication

Le FSCP 13/1 doit utiliser les services FAL définis dans l'IEC 61158-5-13 pour l'échange de données entre les nœuds de sécurité (SN); le FSCP 13/1 définit un modèle de communication producteur/consommateur construit à l'aide de ces services. Les nœuds FSCP 13/1 doivent être logiquement séparés en nœuds producteurs et consommateurs.

Les nœuds producteurs doivent envoyer des données identifiées par une adresse de sécurité spécifique (SADR). Tout SN dans le domaine de sécurité peut recevoir ces données. Cela est réalisé au moyen du mécanisme décrit en 7.5. Dans le modèle de communication, la SADR permet d'identifier de manière unique un SN ainsi qu'un PDU de sécurité envoyé par un nœud spécifique.

La Figure 7 représente une communication type producteur/consommateur. Le nœud producteur SN1 envoie des données. Les données sont identifiées de manière unique par les SADR des producteurs. Chaque nœud dans le domaine de sécurité est capable de recevoir ces données.

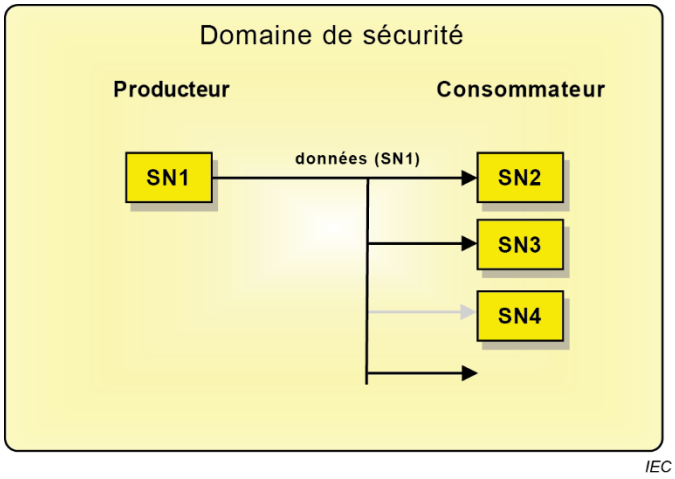


Figure 7 – Communication type producteur/consommateur

La Figure 8 représente une version étendue du modèle de communication producteur/consommateur qui permet au producteur de générer des données séparées pour des consommateurs individuels. A cette fin, le producteur doit utiliser des SADR. Le gestionnaire de configuration de sécurité (SCM) doit s'assurer que les SADR utilisées sont toujours uniques au sein du domaine de sécurité.

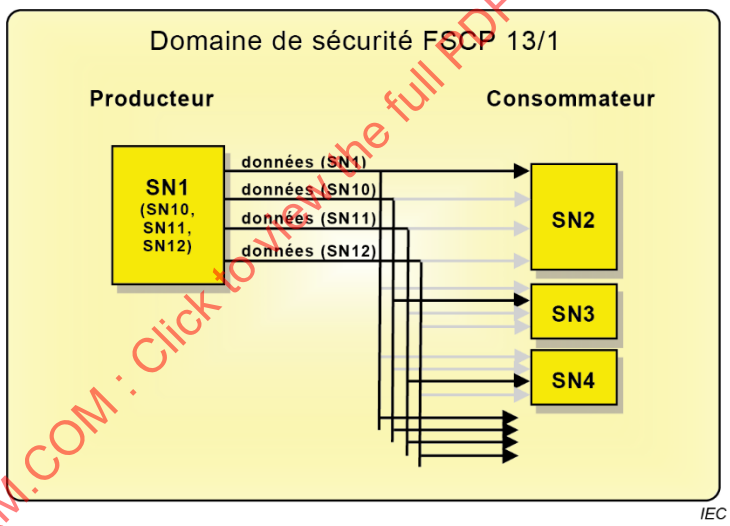


Figure 8 – Communication étendue producteur/consommateur

La Figure 9 représente la communication client/serveur qui doit être utilisée pour les services de configuration et la gestion du réseau. Le SCM (ou un outil de configuration dédié, voir Article 8) doit jouer le rôle de client. Les SN doivent se limiter au rôle de serveur.

A l'aide du modèle de communication client/serveur, le client (SCM ou outil) envoie une demande au serveur (SN) qui est identifié dans le champ ADR (voir 7.1.2) avec l'adresse du serveur et qui contient également l'adresse du client dans le champ TADR (voir 7.1.8). Le serveur envoie une réponse à l'adresse indiquée dans le champ ADR (voir 7.1.2) avec l'adresse du serveur et qui contient également l'adresse du client dans le champ TADR (voir 7.1.8).

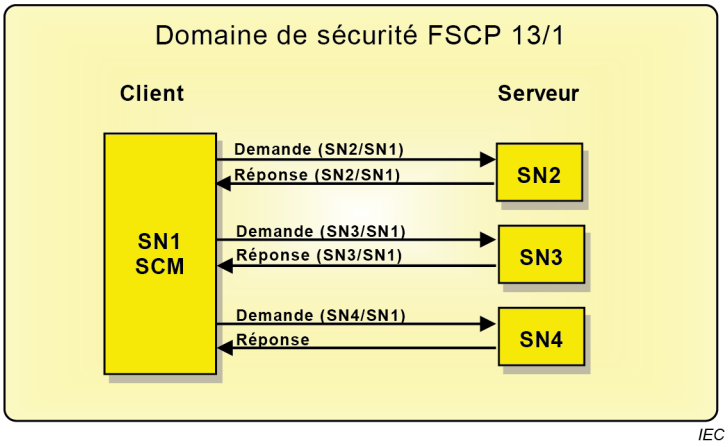


Figure 9 – Communication client/serveur

6.1.3 Rôles des appareils et topologie

6.1.3.1 Généralités

Le Tableau 3 définit les rôles des appareils FSCP 13/1.

Tableau 3 – Rôles des appareils

Rôle de l'appareil	Description
Nœud de sécurité (SN)	Appareil conforme FSCP 13/1
Gestionnaire de configuration de sécurité (SCM)	Service centralisé de gestion d'un domaine de sécurité (SD) [a]
Passerelle de domaine de sécurité (SDG)	Appareil FSCP 13/1 qui permet l'échange de données entre différents SD
[a] Un domaine de sécurité définit un espace d'adresse isolé jusqu'à 1 023 SN.	

Un appareil FSCP 13/1 doit mettre en œuvre le rôle de l'appareil d'un SN et peut prendre le rôle de SCM et/ou de SDG.

La Figure 10 donne une présentation générale des relations possibles entre rôles des appareils et domaines de sécurité. SD1 est un exemple de domaine de sécurité simple. La communication de sécurité des appareils FSCP 13/1 dans le SD1 est limitée aux SN dans SD1.

L'association de SD2 avec SD3 permet d'établir une communication FSCP 13/1 entre des domaines de sécurité différents. Pour ce faire, une SDG spéciale capable de communiquer avec différents domaines de sécurité est utilisée. La communication entre les domaines est traitée en interne par l'appareil. La capacité SDG est indépendante de la capacité SCM d'un appareil. La SDG doit posséder une SADR dans SD2 et une SADR dans SD3.

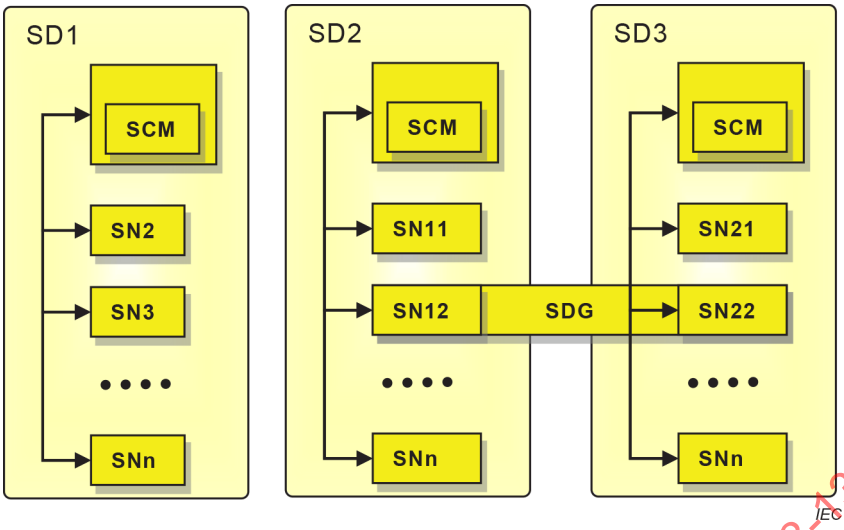


Figure 10 – Présentation générale de la topologie

6.1.3.2 Nœud de sécurité (SN)

Un SN est un nœud au sein d'un réseau conforme au présent document. Un SN doit être identifié par:

- une adresse logique unique au sein d'un SD donné, appelée adresse de sécurité (SADR). La plage de numérotation de la SADR doit être comprise entre 1 et 1 023 (0 est réservé et ne doit pas être utilisé). La SADR doit être définie lors de l'étape de planification de l'application. La présence de doublons SADR dans un SD donné doit être détectée par le SCM lors de la vérification du réseau au moment du démarrage du SCM ou du démarrage de tout nœud dans le domaine de sécurité;
- une adresse physique globalement unique dans tous les appareils conformes au FSCP 13/1, appelée Identifiant unique d'appareil (UDID). La présence de doublons UDID dans un SD donné doit être détectée par le SCM.

Pour fonctionner, un SN peut nécessiter d'autres données, telles que les paramètres de sécurité ou les données d'application. Il dépend de la capacité de l'appareil dans lequel sont enregistrées ces données.

- Le micrologiciel spécifique à l'appareil, l'UDID et les informations de fournisseur de l'appareil (DVI) (voir 7.5.4.9) doivent au minimum être stockés de manière permanente dans l'appareil. Il ne doit y avoir dans le protocole FSCP 13/1 aucun moyen de modifier ces données par le biais du réseau.
- Les SN simples, sans mémoire permanente, doivent recevoir le SDN, la SADR et tout autre paramètre envoyé par le SCM après démarrage.

  NOTE 1 Le SCM enregistre toutes les données spécifiques à ce SN.

- Les SN complexes à mémoire permanente ou à commutateur physique qui permettent le réglage du SDN et de la SADR doivent conserver leurs données en permanence sur l'appareil.

  NOTE 2 Le SCM enregistre uniquement les données qui ne sont pas stockées de manière permanente dans le SN; pour toutes les autres données, le SCM ne procède qu'à une vérification.

6.1.3.3 Domaine de sécurité (SD)

6.1.3.3.1 Généralités

Un domaine de sécurité doit constituer un espace d'adresse qui contient jusqu'à 1 023 SADR (1 à 1 023). Seuls les SN qui partagent le même SD doivent pouvoir communiquer directement entre eux. Les SN qui résident sur des SD différents doivent se limiter à une communication par le biais d'une passerelle de domaine de sécurité.

Un SD est identifié par son numéro de domaine de sécurité (SDN) compris dans la plage de 1 à 1 023. 0 est réservé et ne doit pas être utilisé. Le SDN est défini manuellement. Lors de la planification de l'agencement de l'application, des SDN uniques doivent être utilisés.

6.1.3.3.2 Protection des domaines de sécurité

Le danger d'une attaque extérieure du FSCP 13/1 dans les données de processus du PDU de sécurité (SPDO) est faible. Il existe cependant un risque potentiel pour les services de configuration. Lorsqu'Internet est utilisé conjointement avec le FSCP 13/1, la sûreté du système doit être assurée. Le choix d'un système de chiffrement approprié incombe à la planification du déploiement et ne relève pas du domaine d'application du présent document. Des mesures de sûreté adaptées correspondant à l'état de l'art doivent être appliquées.

La Figure 11 donne un exemple de protection des domaines de sécurité FSCP 13/1.

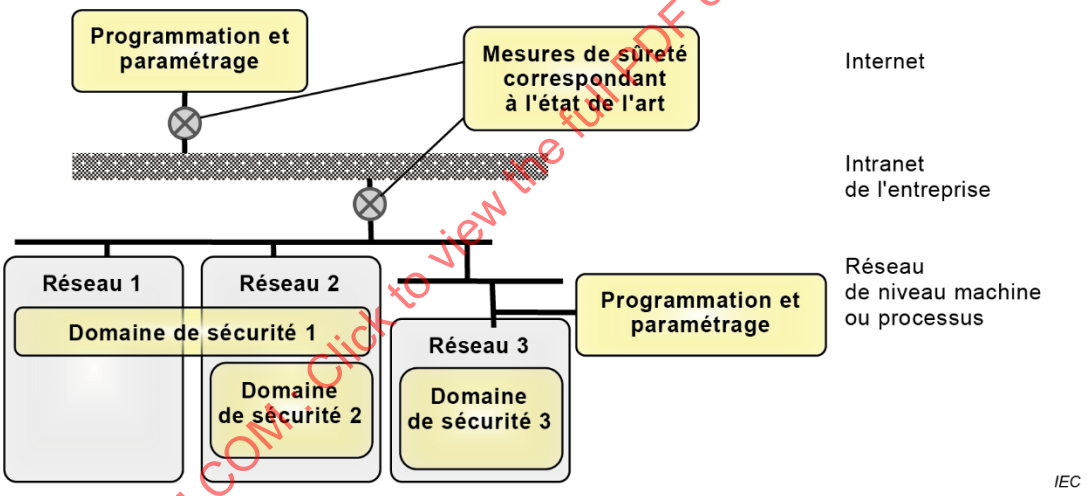


Figure 11 – Protection des domaines de sécurité (exemple)

6.1.3.3.3 Séparation des domaines de sécurité

L'interaction entre les domaines de sécurité ainsi que le danger de confusion durant la configuration et le paramétrage sont évités par l'utilisation de numéros de domaines de sécurité uniques.

Si la taille de l'Intranet de l'entreprise et l'organisation de la gestion du réseau ne permettent pas l'administration de numéros de domaines de sécurité uniques, il convient de diviser le réseau en plusieurs domaines de réseaux gérables et de les séparer au moyen de méthodes établies telles que des pare-feux.

Afin d'éviter des situations cruciales pour la sécurité, les données de SPDO et de PDO de SSDO doivent en outre être codées de manière à assurer leur unicité au sein d'un domaine de sécurité donné (voir 7.1.10). Cela permet ainsi d'identifier et de rejeter les PDO de sécurité qui sont transmis par erreur dans le mauvais domaine.

La Figure 12 donne un exemple de séparation des domaines de sécurité.

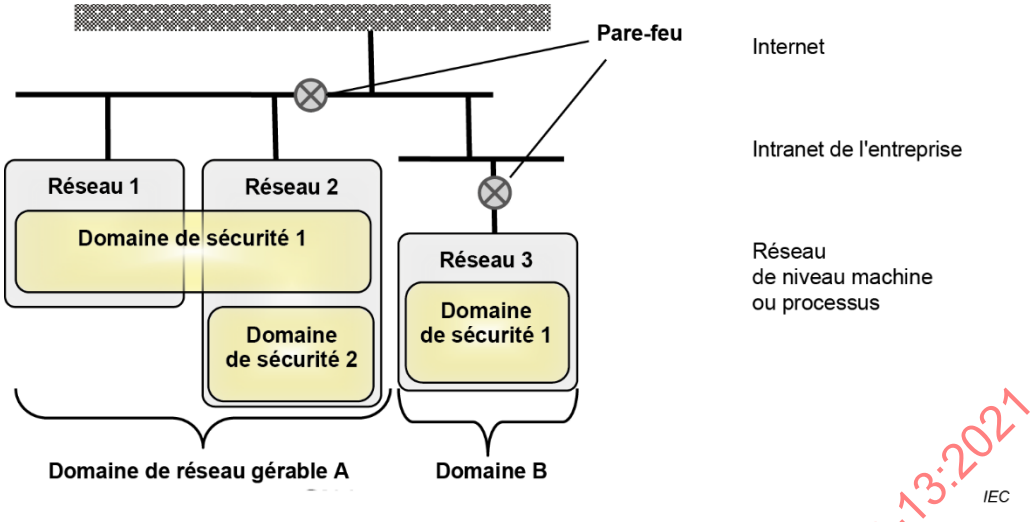


Figure 12 – Séparation des domaines de sécurité (exemple)

6.1.3.4 Passerelle de domaine de sécurité (SDG)

Dans le protocole FSCP 13/1, il doit exister un rôle d'appareil spécial qui est capable de communiquer avec différents domaines de sécurité (SD). La communication entre domaines est traitée en interne par la SDG. Dans un SD donné, la SDG agit comme un SN normal.

Une SDG assure:

– l'acheminement des PDU de SSDO entre les domaines de sécurité si un SN souhaite envoyer un SSDO à un SN situé dans un autre domaine de sécurité; ce SSDO est envoyé à la SDG qui l'achemine au nœud de destination et il peut être nécessaire de passer par plusieurs SDG pour atteindre le SN de destination;

– l'acheminement des données SPDO entre les domaines de sécurité; la SDG peut disposer de moyens de lecture des données contenues dans les SPDO d'un domaine de sécurité donné et envoyer ces données dans un autre domaine de sécurité en les plaçant dans le SPDO de la SDG.

La configuration de la SDG est spécifique au fournisseur.

Le domaine d'application du présent document est la spécification des mécanismes de communication de sécurité nécessaires à l'échange de données au sein d'un domaine de sécurité unique. Les communications entre différents domaines de sécurité (par le biais de la passerelle de domaine de sécurité) ne relèvent pas du domaine d'application du présent document.

6.1.3.5 Gestionnaire de configuration (SCM)

Le gestionnaire de configuration de sécurité (SCM) doit être au centre des responsabilités de gestion FSCP 13/1. Dans un domaine de sécurité, il ne doit être admis qu'un seul SN qui exécute le service SCM.

Le SCM doit assurer:

– l'enregistrement permanent des paramètres spécifiques aux SN qui ne sont pas stockés dans le SN (voir 6.1.3.2);

– l'attribution unique et la vérification de la SADR des SN au sein d'un SD donné;

– la vérification de l'unicité de l'UDID au sein du SD:

– la vérification des paramètres et DVI attendus concernant les SN; et

– l'envoi d'un signal de sauvegarde périodique à tous les SN au sein du SD.

NOTE Avec ce signal de sauvegarde périodique émis par le SCM, le SN détecte certaines situations de défaillance comme lorsqu'aucun SCM ne gère le SN.

6.2 Modèle de cycle de vie

6.2.1 Généralités

Comme défini dans l'IEC 61508, le protocole FSCP 13/1 prend en charge l'application relative à la sécurité au cours de plusieurs phases de son cycle de vie. La communication de sécurité n'est qu'une partie de l'ensemble du concept de sécurité applicable aux machines. Les autres parties de sécurité de la machine doivent également être prises en compte. Le 6.2 concerne uniquement la communication relative à la sécurité dans le cadre du protocole FSCP 13/1.

6.2.2 Concept, planification et mise en œuvre

6.2.2.1 Agencement de l'application

Lors de la conception de l'agencement de l'application, les caractéristiques du FSCP 13/1 doivent être prises en compte. Les possibilités et restrictions suivantes doivent être respectées:

- domaine de sécurité, attribution de SDN;
- séparation et protection des domaines de sécurité;
- adresses de sécurité;
- gestionnaire de configuration de sécurité; et
- communication entre les domaines.

6.2.2.2 Programmation et paramétrage

6.2.2.2.1 Généralités

La programmation et le paramétrage doivent être effectués au moyen d'outils de sécurité. Les fichiers et paramètres de l'application doivent être stockés dans l'appareil SCM uniquement ou directement dans l'appareil correspondant en fonction de ses capacités.

La programmation et le paramétrage de l'application de sécurité doivent comprendre les tâches suivantes:

- l'accès à des SN dans un SD donné est référencé par la SADR uniquement;

  NOTE Cela permet d'installer plusieurs applications identiques avec des SDN différents sans modifier l'application.
- la communication avec d'autres SD n'est possible que par l'intermédiaire d'une SDG;
- localiser le service SCM;
- répertorier les DVI des appareils;
- déployer l'application de sécurité; et
- spécifier les paramètres de sécurité (voir 7.5).

Le SCM est le nœud central pour la configuration et la vérification des appareils de sécurité. Le FSCP 13/1 utilise un seul flux de données pour configurer un appareil. Cela est nécessaire pour satisfaire à l'ensemble des exigences qui résultent de la vaste gamme d'applications compatibles avec FSCP 13/1. Les modes de configuration sont:

- le mode de configuration automatique (ACM); et
- le mode de configuration manuel (MCM).

6.2.2.2.2 Mode de configuration automatique (ACM)

Lorsqu'il utilise le mode de configuration automatique (ACM), le système regroupe automatiquement les UDID des SN connectés. L'ACM doit être utilisé uniquement si l'ensemble des conditions suivantes sont remplies:

– tous les appareils de sécurité peuvent être connectés au moment de la configuration;
– un appareil (UDID unique) pour chaque SN qui possède une SADR spécifique;
– si un appareil de sécurité est remplacé, le SCM vérifie que le nouvel appareil est du même type que celui qui a été remplacé; si cette condition est remplie, le nouvel UDID est recueilli et l'opérateur doit confirmer le remplacement.

6.2.2.2.3 Mode de configuration manuel (MCM)

Le mode de configuration manuel (MCM) s'applique à tous les types d'applications; cependant, ce mode nécessite une configuration manuelle des appareils de sécurité. De ce fait, ce mode est principalement destiné aux applications qui présentent les fonctionnalités suivantes:

– tous les appareils de sécurité connectables ne peuvent pas être connectés en une seule fois;

EXEMPLE 1 Bras robotisé avec plusieurs outils de sécurité.

– plusieurs appareils (plusieurs UDID) pour un SN qui possède une SADR spécifique;

EXEMPLE 2 Soit une application de machine modulaire avec plusieurs chaînes de machines dans une installation donnée. Le module A (correspondant à la SADR) de la machine peut être utilisé pour toutes les chaînes de machines. Pour éviter les problèmes de goulots d'étranglement, il y a plusieurs équipements (c'est-à-dire plusieurs UDID) pour le module A (=SADR) et ces équipements (UDID) sont tous applicables sur une chaîne de machines donnée.

– le remplacement du module doit être réalisé automatiquement sans intervention manuelle.

6.2.3 Mise en service

6.2.3.1 Généralités

La mise en service peut être divisée en trois étapes principales:

a) installation de l'application (voir 6.2.3.2);
b) réglage de la configuration en fonction du mode de configuration (voir 6.2.3.3);
c) vérification de la fonctionnalité de sécurité (voir 6.2.3.4).

6.2.3.2 Installation

Pour installer les applications et les paramètres sur les appareils FSCP 13/1, l'une des méthodes suivantes doit être utilisée:

– un outil de programmation conforme FSCP 13/1 qui utilise les services FSCP 13/1 spécifiés (voir 7.3); ou
– un protocole propriétaire de transmission de sécurité pour l'échange de données du système de programmation vers un appareil de sécurité par le biais du réseau ou une connexion de communication directe (une liaison série, par exemple).

Au cours de la phase d'installation du cycle de vie, la capacité de l'appareil doit déterminer quelles données doivent être téléchargées et à quel emplacement:

– si l'appareil lui-même détient les données, celles-ci doivent être téléchargées vers l'appareil;
– si l'appareil est incapable de détenir les données, celles-ci doivent être stockées sur le SCM. Le SCM doit assurer la mise à jour correcte de tous les appareils à la mise sous tension en utilisant les services de configuration FSCP 13/1.

Après installation des applications et des paramètres, le SDN doit être mis à jour si nécessaire (voir 6.1.3.3).

6.2.3.3 Réglage de la configuration

6.2.3.3.1 Réglage de la configuration en ACM

Le réglage de la configuration en ACM doit comprendre les quatre étapes suivantes:

a) mise sous tension de tous les appareils de sécurité;

b) vérification que la couche de communication fonctionne pour tous les SN;

c) démarrage du SCM en rassemblant les UDID (sans intervention de l'opérateur);

d) le personnel chargé de la mise en service doit effectuer la reconnaissance de chaque appareil.

6.2.3.3.2 Réglage de la configuration en MCM

Le réglage de la configuration en MCM doit comprendre les trois étapes suivantes:

a) le personnel chargé de la mise en service doit saisir manuellement la liste des SADR possibles ainsi que les UDID correspondants; pour cela, un outil de configuration de sécurité doit être utilisé;

b) le SCM doit accepter toutes les combinaisons SADR/UDID configurées;

c) le SCM ne doit pas accepter de discordances et ne doit pas corriger les éventuelles discordances comme cela a été effectué en ACM. En d'autres termes, toutes les discordances doivent être corrigées par le personnel chargé de la mise en service en saisissant la valeur corrigée au moyen de l'outil de configuration de sécurité.

6.2.3.4 Vérification

La vérification doit comprendre les cinq étapes suivantes:

a) le personnel chargé de la mise en service doit vérifier la fonctionnalité de la sécurité par rapport à la spécification de sécurité de l'application. En cas de modification ou de remplacement des appareils sur le trajet au sein de la couche de communication, le personnel chargé de la mise en service doit passer à l'étape décrite en 6.2.3.3;

b) si la vérification est effectuée au cours de la mise en service, toutes les combinaisons de modules échangeables doivent être vérifiées;

NOTE Cela est notamment important dans le cas de machines modulaires complexes (en combinaison avec le MCM).

c) si la vérification est effectuée dans le cadre d'une maintenance, seule la fonctionnalité de sécurité affectée à la tâche de maintenance doit être vérifiée;

d) lorsque la vérification est achevée, il convient d'effectuer une sauvegarde spécifique à l'installation de la configuration actuelle;

e) en cas de remplacement d'un SCM, un téléchargement des fichiers d'application et des paramètres, ainsi que la configuration doivent être effectués; dans le cas contraire, les étapes de mise en service (installation, configuration et vérification) doivent être répétées.

6.2.4 Conditions de fonctionnement

6.2.4.1 Transfert des données de sécurité

Pendant le fonctionnement, le transfert des données de sécurité est géré par le protocole FSCP 13/1. Le réseau sous-jacent assure le transport des données entre les nœuds, avec une qualité de temporisation définie. La perte de synchronisation ou de la qualité du transport dans la couche de communication sous-jacente est détectée par la couche FSCP 13/1 (voir 7.7.2) et entraîne une perte de la disponibilité et non une perte de sécurité (approche "canal noir").

EXEMPLE Voir Figure 13.

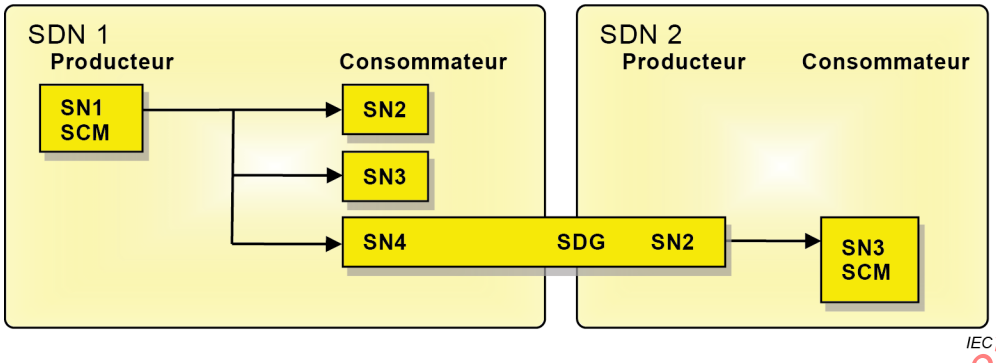


Figure 13 – Exemple de flux de données

6.2.4.2 Synchronisation et validation temporelles

Pour éviter l'absence ou les retards de données intempestifs, une vérification des performances du réseau doit être effectuée. Chaque consommateur doit envoyer périodiquement une requête aux producteurs connectés afin de leur demander leur temps relatif. En comparant les temps relatifs, le temps écoulé et les informations temporelles des transferts, le consommateur doit vérifier l'exactitude des données reçues.

La vérification des performances du réseau fait partie du transfert de données. En premier lieu, un consommateur envoie un PDU de "Demande de temps" au producteur connecté. Après réception de ce PDU reçu, le producteur envoie ses données avec un code supplémentaire qui permet au consommateur d'identifier ce message comme étant une "Réponse de temps". En mesurant le délai écoulé entre la "demande de temps" et la "réponse de temps", le consommateur peut vérifier si le réseau est suffisamment rapide pour répondre aux exigences de l'application, en fonction du temps de réaction le plus défavorable.

Une description complète est fournie en 7.7.2.

6.2.4.3 Sauvegarde

Le service Sauvegarde (voir 7.5.4.6) doit permettre au FSCP 13/1 de s'assurer que seuls les SN configurés sont opérationnels sur un réseau FSCP 13/1. Un SN attend des signaux de sauvegarde fréquents pour rester à l'état opérationnel. Dans le cas contraire, il doit passer à l'état préopérationnel et donc ne plus représenter de risque pour le réseau de sécurité.

La fréquence du signal de sauvegarde doit être spécifique à l'application et est généralement déterminée par la durée nécessaire à la vérification des fonctions de sécurité de l'application.

L'utilisation de mesurages organisationnels, comme la mise hors tension de l'ensemble des SN en cas de reconfiguration du SCM, peut contribuer à éliminer tout risque dans une telle situation. En conséquence, la fréquence du signal de sauvegarde peut être réglée à une valeur très importante (plusieurs jours) dans de tels cas.

6.2.4.4 Démarrage après mise sous tension ou réinitialisation

Après mise sous tension, le SCM (voir 7.7.9) doit effectuer une vérification du réseau. S'il n'y a aucune modification sur le réseau, aucune tâche particulière n'est exigée.

6.2.4.5 Récupération après une défaillance du réseau

Après une panne du réseau, tous les SN concernés passent à l'état préopérationnel. Etant donné que les SN consommateurs correspondants n'obtiennent plus de données, ces nœuds doivent passer à l'état de sécurité.

Après récupération du réseau, tous les SN concernés doivent envoyer un message SNMT_SN_in_PRE_OP au SCM. Le SCM doit ensuite démarrer l'initialisation des SN.

6.2.5 Conditions de maintenance

Voir 9.6.

6.3 Couche de communication non relative à la sécurité

6.3.1 Généralités

Le 6.3 répertorie les exigences applicables à la couche de communication non relative à la sécurité pour le transport des PDU de sécurité. Les protocoles de communication conformes à la CPF 13 (voir IEC 61784-2, IEC 611583-3-13, IEC 61158-4-13, IEC 61158-5-13 et IEC 61158-6-13) satisfont à l'ensemble de ces exigences.

6.3.2 Exigences pour le transport des données

6.3.2.1 Généralités

La couche de communication non relative à la sécurité assure le transport des données entre les SN. La coexistence de nœuds de sécurité et de nœuds non relatifs à la sécurité est admise.

6.3.2.2 Déguisement

Pour éviter toute possibilité de déguisement de données telles que des PDU de sécurité, les nœuds non relatifs à la sécurité ne doivent pas comporter un codage capable de générer des PDU de sécurité selon l'Article 7.

La couche de communication non relative à la sécurité peut avoir un accès en lecture seule pour les données dans un PDU de sécurité. Cela doit être réalisé en permettant l'accès aux données dans les PDU de sécurité sans utiliser les mesures d'intégrité de données spécifiques aux PDU de sécurité, telles que le CRC et la répétition de données de la partie 1 du PDU dans la partie 2 du PDU.

6.3.2.3 Modèle de communication

Du point de vue de la CPF 13, la Figure 14 décrit le modèle de communication de sécurité commun des CN pour 0 à n SN.

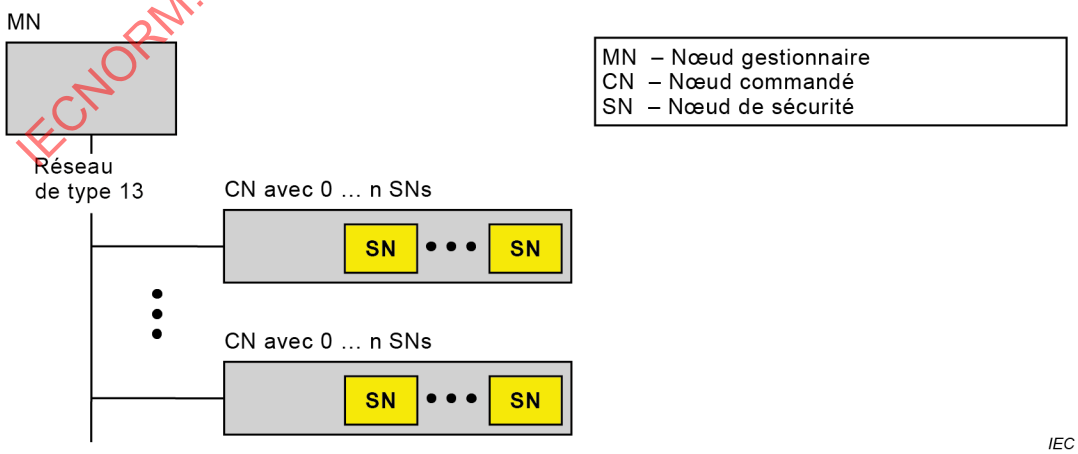


Figure 14 – Modèle de communication

Les fonctionnalités MN et CN sont telles que définies dans l'IEC 61158-3-13; elles ne sont impliquées dans aucune fonction de sécurité.

La fonction de sécurité est mise en œuvre par une entité au sein du CN, appelée nœud de sécurité (SN). Différentes situations peuvent se présenter:

– les CN ne comportent pas de SN;
– les CN comportent un seul SN;
– les CN comportent plusieurs appareils de sécurité représentés comme un seul SN. La sécurité du bus du module et des modules supplémentaires doit être assurée par le fournisseur;
– les CN comportent plusieurs SN; ou
– CN représenté comme un système modulaire et plusieurs SN. La sécurité du bus de connexion du module est assurée par le FSCP 13/1.

6.3.2.4 Transport des SPDO

La couche de communication non relative à la sécurité doit assurer le transport des données SPDO entre les SN. Les services définis dans l'IEC 61158-5-13 doivent être utilisés.

NOTE 1 La charge utile des SPDO est utilisée pour les données de processus cycliques, comme l'état d'une entrée ou d'une sortie. Les SPDO sont retransmis de manière cyclique au cours d'une certaine période. L'envoi d'un SPDO est uniquement déclenché par un élément temporel et non lié à une modification des données proprement dites.

Les données SPDO doivent être transmises comme des PDU de sécurité dans des PDU de couche d'application de type 13 (APDU). Les APDU Isoc2 conformes à l'IEC 61158-6-13 doivent être utilisés. Un SPDO est généré de manière cyclique par un SN, appelé producteur; ce SPDO peut avoir un ou plusieurs SN cibles, appelés consommateurs. Pour le transport d'un SPDO de son producteur à ses consommateurs, des mécanismes de type 13 doivent être utilisés.

Le principe de base du transport de SPDO est représenté à la Figure 15.

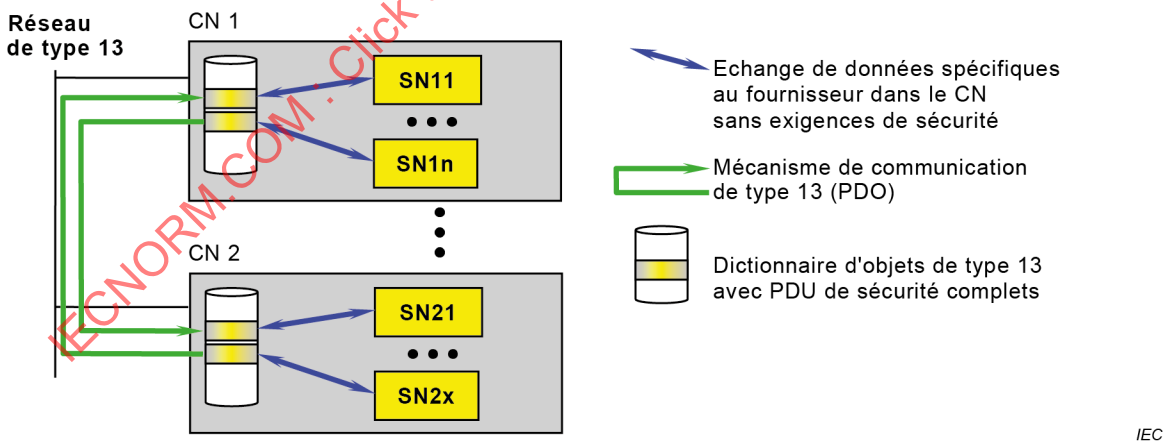


Figure 15 – Transport des SPDO

Le trajet de transmission entre le SN producteur et le SN consommateur est défini de la manière suivante:

– le SN producteur construit un SPDU;
– dans le CN auquel le SN producteur est connecté (CN producteur), une entrée de dictionnaire d'objets de domaine type de données doit être définie pour incorporer le SPDU;

- le déplacement de données SPDU vers l'entrée du dictionnaire d'objets incombe au CN producteur;
- l'entrée de dictionnaire d'objets doit alors être transmise dans un PDU de couche d'application de type 13 (APDU), qui est transmis de manière cyclique en utilisant l'ASE d'objet de données de processus (ASE de PDO) comme spécifié dans l'IEC 61158-5-13;
- le dictionnaire d'objets du CN de réception (CN consommateur) doit définir un objet de domaine type de données qui correspond à la définition de l'entrée du dictionnaire d'objets du producteur. Le CN consommateur doit recevoir l'APDU et copier le PDU de sécurité qu'il contient vers l'objet domaine mentionné auparavant; et
- au cours de la dernière étape, le CN consommateur assure le déplacement des données SPDU de l'entrée du dictionnaire d'objets vers le SN consommateur.

Les mécanismes impliqués dans la transmission du SPDO entre le dictionnaire d'objets de CN et les SN (représentés par des flèches bleues sur la Figure 15) ne relèvent pas du domaine d'application du présent document et peuvent être spécifiques au fournisseur.

NOTE 2 Du point de vue des services et du protocole de communication de type 13, il n'est pas nécessaire d'établir une distinction entre la communication de SPDO et d'autres données. Il n'y a aucun traitement particulier des données de sécurité par des mécanismes de type 13. Cela correspond à l'approche "canal noir" appliquée.

NOTE 3 Aucune connaissance concernant le trajet de communication de type 13 n'est nécessaire aux SN impliqués dans la communication.

6.3.2.5 Transport des SSDO

La couche de communication non relative à la sécurité doit assurer le transport des données SSDO entre les SN. Les services définis dans l'IEC 61158-5-13 doivent être utilisés.

La Figure 16 représente le principe de base du transport des SSDO.

NOTE 1 La charge utile des SSDO est utilisée pour les communications de données spontanées, telles que la configuration et le paramétrage.

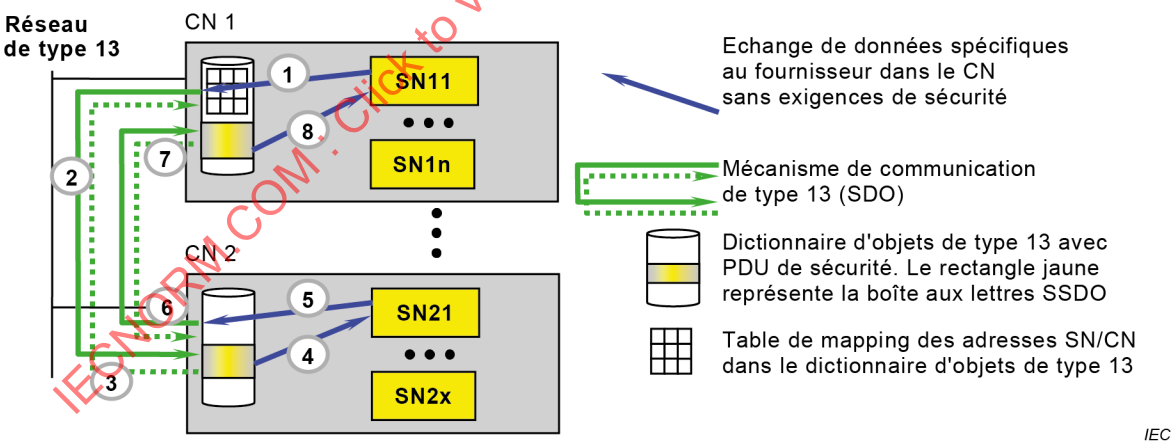


Figure 16 – Transport des SSDO

Les données SSDO doivent être transmises comme des PDU de sécurité dans des PDU de couche d'application de type 13 (APDU). Les APDU Asyn2 avec SDO de type pduBody conformes à l'IEC 61158-6-13 doivent être utilisés.

Un SSDO est généré de manière spontanée par un SN (le SN d'origine) et est transmis à exactement un SN cible. Pour transporter un SSDO de son origine à sa destination, des mécanismes de type 13 doivent être utilisés.

L'échange de SSDO entre les SN est réalisé au moyen de services et de protocoles de type 13 associés à un mécanisme de messagerie (les chiffres entre parenthèses renvoient à la Figure 16):

(1) le SN d'origine génère une demande SSDO à un SN de destination spécifique et construit un SPDU correspondant;

(2) le CN qui dessert le SN d'origine crée un accès en écriture de SDO vers la boîte aux lettres du CN qui dessert le SN de destination; le SPDU de l'étape (1) est écrit dans le dictionnaire d'objets du CN cible; cette opération nécessite une table de mapping entre les adresses SN et CN dans le CN qui dessert le SN d'origine afin d'obtenir l'ID du nœud de type 13 du CN qui dessert le SN de destination (CN cible); il incombe à la configuration système de type 13 de fournir la table de mapping des adresses, la boîte aux lettres doit être une entrée de dictionnaire d'objets de type 13 du domaine type de données qu'il convient de définir pour intégrer un SPDU;

(3) l'accès en écriture SDO est confirmé par le protocole SDO et la connexion est maintenue ouverte;

(4) le CN de destination redirige les données (SPDU) dans la boîte aux lettres vers le SN de destination; pour cela, le CN de destination doit lire les informations d'adresse contenues dans le SPDU;

(5) le SN de destination génère une réponse SSDO et construit un SPDU correspondant;

(6) le CN de destination crée un accès en écriture SDO vers la boîte aux lettres du CN d'origine et écrit le SPDU de réponse dans le dictionnaire d'objets du CN d'origine;

(7) l'accès en écriture de SDO est confirmé, la connexion est fermée; et

(8) le CN d'origine redirige les données (SPDU) dans la boîte aux lettres vers le SN de destination; pour cela, le CN d'origine doit lire les informations d'adresse contenues dans le SPDU.

Les mécanismes impliqués dans la transmission du SPDU entre le dictionnaire d'objets de CN et les SN (représentés par des flèches bleues sur la Figure 16) ne relèvent pas du domaine d'application du présent document et peuvent être spécifiques au fournisseur.

NOTE 2 Du point de vue des services et du protocole de communication de type 13, il n'est pas nécessaire d'établir une distinction entre la communication de SSDO et d'autres données. Il n'y a aucun traitement particulier des données de sécurité par des mécanismes de type 13. Cela correspond à l'approche "canal noir" appliquée.

NOTE 3 Aucune connaissance concernant le trajet de communication de type 13 n'est nécessaire aux SN impliqués dans la communication.

6.3.2.6 Représentation des données de diagnostic

La couche de communication non relative à la sécurité doit prendre en charge des services qui fournissent des données de diagnostic spécifiques au FSCP 13/1 selon l'Article 7.

La Figure 17 représente le principe de base de la représentation des données de diagnostic.

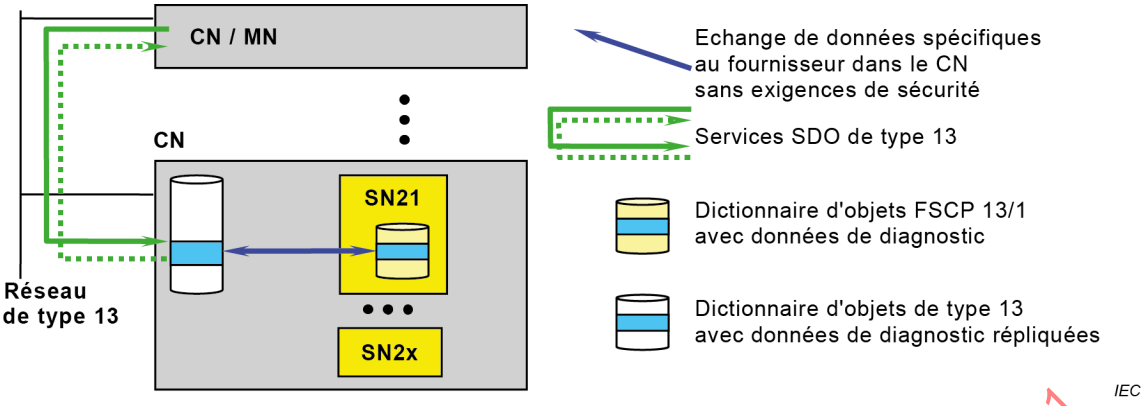


Figure 17 – Représentation des données de diagnostic

Les données de diagnostic dans un SN sont stockées dans le SOD (voir Article 7).

NOTE En général, ces données de diagnostic sont demandées à des nœuds particuliers (une unité de commande centralisée, une interface homme-machine, des équipements de service, par exemple) pour traiter les données de diagnostic et les communiquer à l'utilisateur final.

Le CN doit répliquer les données de diagnostic du SOD dans le dictionnaire d'objets de type 13 du CN et il est alors possible d'y accéder au moyen de services SDO de type 13.

6.3.3 Protection et séparation des domaines

Selon 6.1.3.3.2 et 6.1.3.3.3, la couche de communication doit prendre en charge des fonctions de sûreté spécifiques afin de protéger le domaine contre les attaques extérieures et éviter les interférences entre des domaines séparés.

7 Protocole de couche de communication de sécurité

7.1 Format PDU de sécurité

7.1.1 Structure des PDU de sécurité

7.1.1.1 Généralités

Un PDU de sécurité est transféré dans la partie "données" d'un APDU de type 13. Un APDU de type 13 peut contenir zéro ou plusieurs PDU de sécurité, qui reflètent chacun une relation producteur/consommateur entre SN (voir Figure 18). Les données de sécurité et non relatives à la sécurité peuvent être mélangées dans un APDU de type 13.

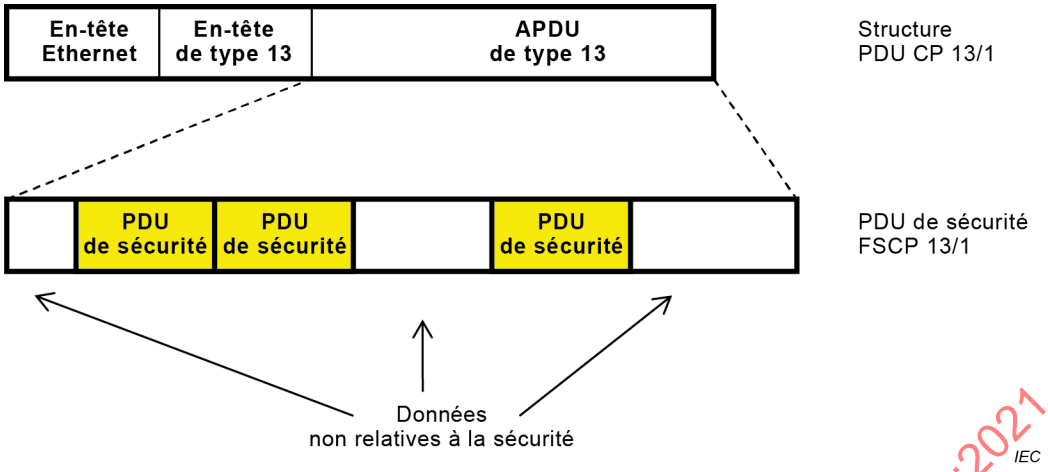


Figure 18 – PDU de sécurité dans un PDU CP 13/1

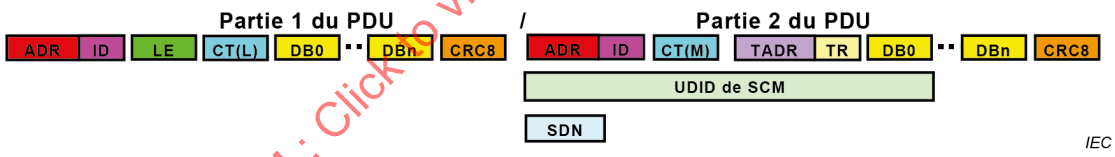
7.1.1.2 PDU de sécurité de base

La structure de PDU de sécurité de base doit correspondre à celle représentée à la Figure 19 et à la Figure 20. Le PDU de sécurité de base doit être constitué de deux sous-parties et peut transporter jusqu'à 240 octets de données utiles. Pour le taux d'erreurs résiduelles concernant la longueur du PDU de sécurité, voir 9.5.6.

Selon la charge utile, deux types de PDU de sécurité de base doivent être utilisés:

– 1 à 8 octets de charge utile: un format de PDU conforme à la Figure 19 doit être utilisé (avec un CRC 8 bits);

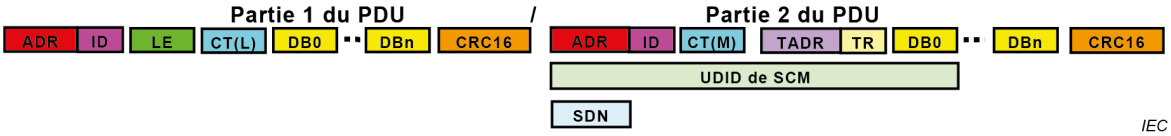
– à partir de 9 octets de charge utile: un format de PDU conforme à la Figure 20 doit être utilisé (avec un CRC 16 bits).



Légende

n nombre d'octets de charge utile

Figure 19 – PDU de sécurité de base pour *n* = 0 à 8 octets de données utiles



Légende

n nombre d'octets de charge utile

Figure 20 – PDU de sécurité de base à partir de 9 octets de données utiles

Le Tableau 4 spécifie le PDU de sécurité de base. Il doit être constitué de deux parties appelées parties 1 et 2 du PDU. Les données de la partie 2 du PDU des télégrammes SPDO et SSDO (et non des télégrammes SNMT) sont en outre codées avec l'UDID du SCM en utilisant une opération XOR logique (voir 7.1.10).

Tableau 4 – Format de PDU de sécurité de base

Décalage d'octet [a]	Décalage de bit							
	7	6	5	4	3	2	1	0
0	ADR (bits 0-7), voir 7.1.2							
1	ID, voir 7.1.3						ADR (8,9)	
2	LE, voir 7.1.4							
3	CT (bits 0-7), voir 7.1.5							
4 – n+3	DB 0 à DB n, voir 7.1.6							
n+4 – n+4+o	CRC-8/CRC-16, voir 7.1.7							
n+5+o	ADR (bits 0-7) XOR SDN (bits 0-7)							
n+6+o	ID						ADR (8,9) XOR SDN (8,9)	
n+7+o	CT (Bit 8-15)							
n+8+o	TADR (bits 0-7), voir 7.1.8							
n+9+o	TR, voir 7.1.9						TADR (8,9)	
n+10+o – 2n+9+o	DB 0 à DB n							
2n+10+o – 2n+10+2o	CRC-8/CRC-16							
NOTE Les lignes en gris indiquent les données de la partie 2 du PDU.								
[a] Par rapport au début du PDU de sécurité de base, n doit être le nombre d'octets de charge utile dans le champ DB 0 à DB n, où 0 ≤ n ≤ 240. o doit être le décalage de correction du CRC si 0 ≤ n ≤ 8, dans ce cas o = 0 (CRC 8); si 9 ≤ n ≤ 240, dans ce cas o = 1 (CRC-16).								

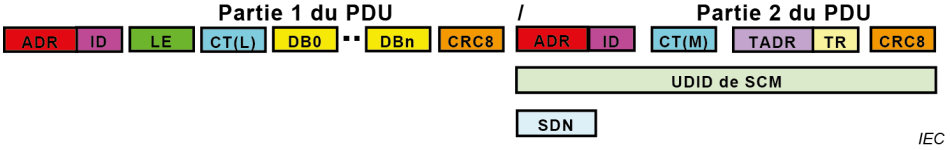
Les champs du PDU de sécurité de base sont définis plus précisément à partir de 7.1.2.

7.1.1.3 PDU de sécurité Slim

La structure du PDU de sécurité Slim doit correspondre à celle représentée à la Figure 21 et à la Figure 22. Le PDU de sécurité Slim doit être constitué de deux sous-parties et peut transporter jusqu'à 240 octets de données utiles. Pour le taux d'erreurs résiduelles concernant la longueur du PDU de sécurité, voir 9.5.6.

Selon la longueur de charge utile, deux types de PDU de sécurité Slim doivent être utilisés:

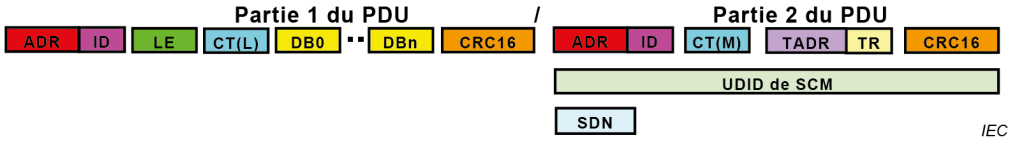
– 1 à 8 octets de charge utile: un format de PDU conforme à la Figure 21 doit être utilisé (avec un CRC 8 bits);
– à partir de 9 octets de charge utile: un format de PDU conforme à la Figure 22 doit être utilisé (avec un CRC 16 bits).



Légende

n nombre d'octets de charge utile

Figure 21 – PDU de sécurité Slim pour *n* = 0 à 8 octets de données utiles



Légende

n nombre d'octets de charge utile

Figure 22 – PDU de sécurité Slim à partir de 9 octets de données utiles

Le Tableau 5 spécifie le PDU de sécurité Slim. Il doit être constitué de deux parties appelées parties 1 et 2 du PDU. Les données de la partie 2 du PDU sont en outre codées avec l'UDID du SCM en utilisant une opération XOR logique (voir 7.1.10).

Tableau 5 – Format de PDU de sécurité Slim

Décalage d'octet [a]	Décalage de bit							
	7	6	5	4	3	2	1	0
0	ADR (bits 0-7), voir 7.1.2							
1	ID, voir 7.1.3						ADR (8,9)	
2	LE, voir 7.1.4							
3	CT (bits 0-7), voir 7.1.5							
4 – n+3	DB 0 à DB n, voir 7.1.6							
n+4 – n+4+o	CRC-8/CRC-16, voir 7.1.7							
n+5+o	ADR (bits 0-7) XOR SDN (bits 0-7)							
n+6+o	ID						ADR (8,9) XOR SDN (8,9)	
n+7+o	CT (Bit 8-15)							
n+8+o	TADR (bits 0-7), voir 7.1.8							
n+9+o	TR, voir 7.1.9						TADR (8,9)	
n+10+o – n+10+2o	CRC-8/CRC-16							
NOTE Les lignes en gris indiquent les données de la partie 2 du PDU.								
[a] Par rapport au début du PDU de sécurité Slim, n doit être le nombre d'octets de charge utile dans le champ DB 0 à DB n, où 0 ≤ n ≤ 240. o doit être le décalage de correction du CRC si 0 ≤ n ≤ 8, dans ce cas o = 0 (CRC 8); si 9 ≤ n ≤ 240, dans ce cas o = 1 (CRC-16).								

Les champs du PDU de sécurité sont définis plus précisément à partir de 7.1.2.

7.1.1.4 PDU de sécurité avec un compteur 40 bits

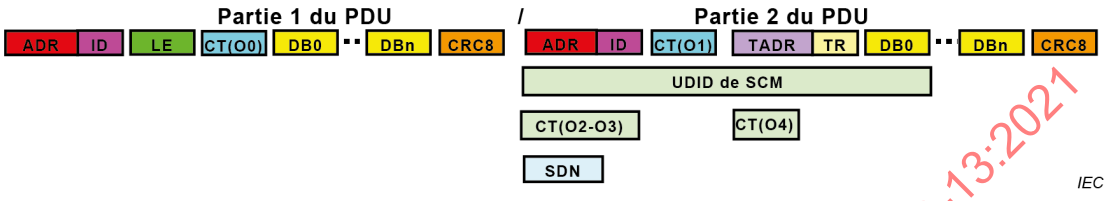
FSCP 13/1 peut utiliser un compteur 40 bits. Les 40 bits sont constitués des 16 bits de champs CT explicitement réservés et de 24 bits supplémentaires encodés sur la structure de la partie 2 du PDU. Seuls les télégrammes de données SPDO qui ne sont pas utilisés pour la synchronisation temporelle doivent être utilisés pour les compteurs 40 bits.

Comme l'encodage des PDU avec un compteur 40 bits n'est pas le même que celui des PDU avec un compteur 16 bits, il est nécessaire de prévoir une phase d'initialisation (voir 7.6.6).

La structure du PDU de sécurité avec un compteur 40 bits doit correspondre à celle représentée à la Figure 23 et à la Figure 24. Le PDU de sécurité avec un compteur 40 bits doit être constitué de deux sous-parties et peut transporter jusqu'à 240 octets de données utiles. Pour le taux d'erreurs résiduelles concernant la longueur du PDU de sécurité, voir 9.5.6.

Selon la longueur de charge utile, deux types de PDU de sécurité avec un compteur 40 bits doivent être utilisés:

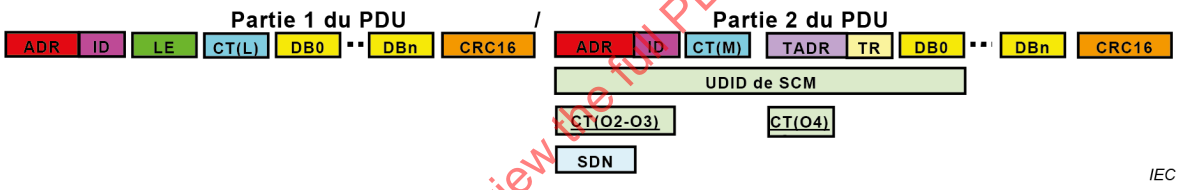
– 1 à 8 octets de charge utile: un format de PDU conforme à la Figure 23 doit être utilisé (avec un CRC 8 bits);
– à partir de 9 octets de charge utile: un format de PDU conforme à la Figure 24 doit être utilisé (avec un CRC 16 bits).



Légende

n nombre d'octets de charge utile

Figure 23 – PDU de sécurité avec un compteur 40 bits pour *n* = 0 à 8 octets de données utiles



Légende

n nombre d'octets de charge utile

Figure 24 – PDU de sécurité avec un compteur 40 bits à partir de 9 octets de données utiles

Le Tableau 6 spécifie le PDU de sécurité avec un compteur 40 bits. Il doit être constitué de deux parties appelées parties 1 et 2 du PDU. Les données de la partie 2 du PDU sont en outre codées avec l'UDID du SCM et les bits 16 à 39 du champ CT à l'aide d'opérations XOR logiques (voir 7.1.10 et 7.1.5).

Tableau 6 – PDU de sécurité avec un compteur 40 bits

Décalage d'octet [a]	Décalage de bit							
	7	6	5	4	3	2	1	0
0	ADR (bits 0-7), voir 7.1.2							
1	ID, voir 7.1.3						ADR (8,9)	
2	LE, voir 7.1.4							
3	CT (bits 0-7), voir 7.1.5							
4 – n+3	DB 0 à DB n, voir 7.1.6							
n+4 – n+4+o	CRC-8/CRC-16, voir 7.1.7							
n+5+o	ADR (bits 0-7) XOR CT (bits 16-23) XOR SDN (bits 0-7)							
n+6+o	ID XOR CT (26-31)						ADR (8,9) XOR CT (24,25) XOR SDN (8,9)	
n+7+o	CT (Bit 8-15)							
n+8+o	TADR (bits 0-7) XOR CT (bits 32-39), voir 7.1.8							
n+9+o	TR, voir 7.1.9						TADR (8,9)	
n+10+o – n+10+2o	CRC-8/CRC-16							
NOTE Les lignes en gris indiquent les données de la partie 2 du PDU.								
[a] Par rapport au début du PDU de sécurité avec le champ CT sur 40 bits, n doit être le nombre d'octets de charge utile dans le champ DB 0 à DB n, où $0 \leq n \leq 240$. o doit être le décalage de correction du CRC. Si $0 \leq n \leq 8$, alors o = 0 (CRC 8). Si $9 \leq n \leq 240$, alors o = 1 (CRC-16).								

Les champs du PDU de sécurité sont définis plus précisément à partir de 7.1.2.

7.1.2 Champ d'adresse (ADR)

Le champ ADR définit l'adresse de sécurité (SADR) du nœud de sécurité (SN). Le champ doit être une valeur de 10 bits. Le champ ADR doit accepter une plage de valeurs de 1 à 1 023 (0 est réservé et ne doit pas être utilisé).

Le champ ADR dans la partie 2 du PDU doit également être utilisé pour transporter les informations du numéro de domaine de sécurité (SDN) en utilisant une opération XOR logique (voir 7.1.10).

Le champ ADR doit contenir la SADR du SN qui a envoyé le PDU (adresse d'origine) ou reçu le PDU (adresse de destination). L'adresse d'origine doit être la SADR du SN qui envoie le PDU; l'adresse de destination doit être la SADR du SN qui reçoit le télégramme.

En fonction du contenu du champ ADR de la partie 1 du PDU, la couche de communication non relative à la sécurité est capable de transporter les données au nœud de destination désigné sans effectuer une analyse supplémentaire du champ ID ou d'autres attributs de PDU.

Dans le cas d'un PDU de sécurité avec un compteur 40 bits, le champ ADR de la partie 2 du PDU doit également être utilisé pour transporter des parties de la valeur CT à l'aide d'une opération XOR logique (voir 7.1.5).

7.1.3 Champ d'identification de PDU (ID)

Le champ ID doit identifier le type de PDU. Le Tableau 7 spécifie le codage du champ ID. Des codes réservés pourront être utilisés dans les futures éditions du présent document. Actuellement, ils ne doivent pas être utilisés et doivent être ignorés s'ils sont reçus. Pour identification du type de PDU, les bits 5, 6 et 7 doivent être utilisés. Chaque type de PDU possède différents types de télégrammes qui doivent être spécifiés par les bits 2, 3 et 4 du champ ID.

Tableau 7 – Champ d'identification de PDU (ID)

Décalage de bit								Type de PDU
7	6	5	4	3	2	1	0	
0	0	0	0	0	x [a]	Bit 9 du champ d'adresse	Bit 8 du champ d'adresse	Non admis (PDU illicite)
0	x	x	x	x	x			Réservé
1	0	0	x	x	x			Réservé
1	0	1	x	x	x			SNMT
1	1	0	x	x	x			SPDO
1	1	1	x	x	x			SSDO

[a] Un x dans ce tableau indique le type de télégramme pour un type de PDU spécifique.

Le Tableau 8 spécifie toutes les combinaisons de champs ID valides.

Tableau 8 – Combinaisons de champs ID utilisées

Décalage de bit								Type de télégramme
7	6	5	4	3	2	1	0	
1	0	1	0	0	0	Bit 9 du champ d'adresse	Bit 8 du champ d'adresse	SNMT_Request_UDID
1	0	1	0	0	1			SNMT_Response_UDID
1	0	1	0	1	0			SNMT_Assign_SADR
1	0	1	0	1	1			SNMT_SADR_assigned
1	0	1	1	0	0			SNMT_Service_Request
1	0	1	1	0	1			SNMT_Service_Response
1	0	1	1	1	1			SNMT_SN_reset_guarding_SCM
1	1	0	0	0	x[a]			SPDO_Data_Only
1	1	0	0	1	x[a]			SPDO_Data_with_Time_Request
1	1	0	1	0	x[a]			SPDO_Data_with_Time_Response
1	1	1	0	0	0			SSDO_Service_Request
1	1	1	0	0	1			SSDO_Service_Response
1	1	1	0	1	0			SSDO_Service_Request_Slim
1	1	1	0	1	1			SSDO_Service_Response_Slim

[a] Bit valide de connexion, voir 7.2.2.

Les PDU SSDO et SNMT doivent être divisés en télégrammes de demande et de réponse (voir Tableau 9). Le bit 2 du champ ID doit être utilisé pour identifier le télégramme: demande ou réponse. Les bits 0 et 1 de l'octet où se trouve le champ ID sont utilisés par le champ d'adresse (voir Tableau 4).

Tableau 9 – Identifiant de demande/réponse

Décalage de bit								Description
7	6	5	4	3	2	1	0	
x	x	x	x	x	0			Demande SSDO/SNMT
x	x	x	x	x	1			Réponse SSDO/SNMT

7.1.4 Champ de longueur (LE)

Le champ LE spécifie le nombre d'octets de données utiles dans le PDU. La plage de valeurs valides doit être de 0 à 240. La valeur de LE doit déterminer le type de CRC utilisé pour le PDU (voir Tableau 10).

Tableau 10 – Type de CRC en fonction de LE

Valeur de LE	CRC utilisé
0 – 8	CRC-8
9 – 240	CRC-16
241 – 255	Réservé

7.1.5 Champ de temps consécutifs (CT)

7.1.5.1 Généralités

La base de temps du champ CT doit être définie dans un objet SOD 1200h de sous-index 03h (voir 7.5.4.13).

7.1.5.2 Champ de temps consécutifs (CT) – 16 bits

Le champ de Temps consécutifs (CT) doit être divisé en une partie LSB envoyée dans la partie 1 du PDU et une partie MSB envoyée dans la partie 2 du PDU. Les deux octets constituent le jeu de données CT dans une plage de 0 à 65 535. Dans le cas d'un SPDO, lorsque le PDU est envoyé, le champ CT doit être utilisé pour la valeur du temporisateur interne du SN (voir 7.2); dans le cas d'un SSDO, le champ CT décrit le numéro de demande d'accès au SOD (voir 7.3).

7.1.5.3 Champ de temps consécutifs (CT) – 40 bits

Pour les 16 bits de poids faible, les mêmes champs que pour le champ CT 16 bits doivent être utilisés, c'est-à-dire que les bits 0 à 7 doivent être transmis dans le champ CT de la partie 1 du PDU et les bits 8 à 15 doivent être transmis par le champ CT de la partie 2 du PDU.

Les bits 16 à 31 doivent être codés en effectuant une opération XOR sur les deux premiers octets de la partie 2 du PDU (ADR et ID) et les bits 32 à 39 doivent être codés en effectuant une opération XOR sur les 8 bits de poids faible de la TADDR dans la partie 2 du PDU.

7.1.6 Champ de données utiles (DB0 à DBn)

Ce champ doit contenir les données utiles. Des données utiles d'une longueur de 0 à 240 octets doivent être prises en charge. Des données utiles d'une longueur de 0 octet peuvent être utilisées pour les services de synchronisation temporelle (voir 7.2.4, 7.2.5).

7.1.7 Champ de contrôle de redondance cyclique (CRC-8/CRC-16)

Le champ CRC doit contenir la valeur du contrôle de redondance cyclique pour la partie de PDU à laquelle il appartient, qui doit être calculée à l'aide de l'un des polynômes indiqués dans le Tableau 11. Le type (CRC-8 ou CRC-16) dépend des valeurs du champ de longueur (LE) et du champ d'identification de PDU (ID).

Tableau 11 – Polynômes du CRC pour les SPDU

Utilisation prévue	Polynôme	Polynôme	Polynôme selon [35]
données utiles jusqu'à 8 octets	$X^8+X^5+X^3+X^2+X+1$	$12F_{hex}$	0x97
données utiles à partir de 9 octets, services SSDO SLIM uniquement	$X^{16}+X^{14}+X^{12}+X^{11}+X^8+X^5+X^4+X^2+1$	15935_{hex}	0xAC9A
données utiles à partir de 9 octets, tous les services sauf SSDO SLIM	$X^{16}+X^{14}+X^{13}+X^{12}+X^{10}+X^8+X^6+X^4+X^3+X+1$	$1755B_{hex}$	0xBAAD

Les règles suivantes doivent s'appliquer pour la génération du CRC:

1) la valeur de départ est 0 pour le CRC-8 et pour chaque polynôme CRC-16;
2) le bit de poids fort du premier octet est décalé en premier;
3) le CRC doit être calculé sur tous les champs de la partie du PDU, sauf le CRC;
   i) Partie 1 du PDU: ADR, ID, LE, CT(L/0-7), DB0 à DBn;
   ii) Partie 2 du PDU: ADR, ID, CT(H/8-15), TADR, TR, DB0 à DBn.

Le CRC-8 doit satisfaire aux règles de codage d'un type de données UNSIGNED8.

Le CRC-16 doit satisfaire aux règles de codage d'un type de données UNSIGNED16.

L'Article A.1 fournit un exemple de code pour le calcul du CRC.

7.1.8 Champ d'adresse de demande de temps (TADR)

Le champ TADR spécifie:

– l'adresse de sécurité (SADR) d'un SN qui répond à la Demande de synchronisation temporelle dans le cas de SPDO (voir 7.2); ou
– la SADR (d'origine ou de destination) dans le cas de SSDO (voir 7.3) et de SNMT (voir 7.4).

7.1.9 Champ de numéro distinctif de demande de temps (TR)

Le champ TR doit être un nombre compris dans la plage de 0 à 63, qui doit être écrit en miroir par l'appareil afin de distinguer ce message d'autres télégrammes de demande de temps.

De plus, ce champ doit être utilisé pour négocier les bits de fonctionnalités du PDU de sécurité (voir 7.6.6.2).

7.1.10 UDID de codage SCM (UDID de SCM)

L'UDID de 6 octets de SCM doit être codé par une opération OU exclusif au niveau du bit avec les six premiers octets de données utiles de la partie 2 du PDU des télégrammes SPDO et SSDO. Les télégrammes SNMT ne doivent pas être codés de cette manière.

L'UDID du SCM est connu pour chaque appareil (voir 7.4.3.12); le SN doit vérifier chaque télégramme SPDO et SSDO reçu afin de s'assurer que l'UDID correct a été utilisé pour le codage des télégrammes.

NOTE Il est ainsi possible de détecter les éventuelles discordances de séparation de domaines (voir 6.1.3.3.3).

7.2 Objet de données de processus de sécurité (SPDO)

7.2.1 Généralités

Les objets de données de processus de sécurité (SPDO) doivent être utilisés pour la transmission de données de processus cruciales pour la sécurité, ainsi que pour la transmission de données de demande/réponse de synchronisation temporelle issues d'un producteur SPDO et destinées aux consommateurs SPDO. La réception d'un SPDO par un consommateur SPDO doit être déterminée par l'adresse du producteur SPDO.

Seuls les PDU de sécurité de base doivent être utilisés pour les SPDO.

7.2.2 Types de télégrammes SPDO

Le champ ID (bits 2 à 4) doit identifier le type de télégramme SDPO. Trois types de télégrammes doivent être définis:

– télégramme de données uniquement;
– données avec télégramme de demande de temps; et
– données avec télégramme de réponse de temps.

Le Tableau 12 spécifie le codage du champ ID.

Tableau 12 – Types de télégrammes SPDO (bits 2, 3 et 4 du champ ID)

Numéro de bit du champ ID								Type de PDU
7	6	5	4	3	2	1	0	
1	1	0	0	0	X[a]	Bits 8 et 9 du champ d'adresse		SPDO_Data_Only
1	1	0	0	1	X[a]			SPDO_Data_with_Time_Request
1	1	0	1	0	X[a]			SPDO_Data_with_Time_Response
1	1	0	1	1	X[a]			Réservé

[a] Le bit 2 doit être utilisé comme un "bit valide de connexion". Si la synchronisation temporelle de tous les RxSPDO à synchroniser avec le TxSPDO est correcte et que les données utiles dans le TxSPDO sont valides, le bit doit être mis à 1_b. Dans le cas contraire, il doit être mis à 0_b.

Exemple Bit valide de connexion

Un SN possède 3 RxSPDOs et 2 TxSPDOs. Les RxSPDO 1 et 2 sont synchronisés à l'aide de TxSPDO 1, et le RxSPDO 3 est synchronisé à l'aide de TxSPDO 2.

La synchronisation de RxSPDO 1 est incorrecte.

La synchronisation des RxSPDO 2 et 3 est correcte.

Le TxSPDO 1 est envoyé avec un bit valide de connexion mis à 0_b (RxSPDO 1 incorrect & RxSPDO 2 correct -> incorrect).

Le TxSPDO 2 est envoyé avec un bit valide de connexion mis à 1_b (RxSPDO 3 correct).

7.2.3 Télégramme de données uniquement

Le télégramme de données uniquement (voir Figure 25) doit être utilisé pour transmettre les données de processus d'un SN (producteur SPDO) sans demande ou réponse de synchronisation temporelle. Les données peuvent être consommées par un autre SN (consommateur de SPDO).

En même temps que les données (Data), le producteur SPDO doit fournir son adresse SN (ADR) ainsi que le nombre d'octets de données utiles transmis (LE) et la valeur courante de son temporisateur interne (CT).

Dans une communication SPDO avec des valeurs CT sur 40 bits, les 24 bits de poids fort sont encodés sur différentes parties du télégramme. Voir 7.1.5.3 pour plus d'informations sur l'encodage.

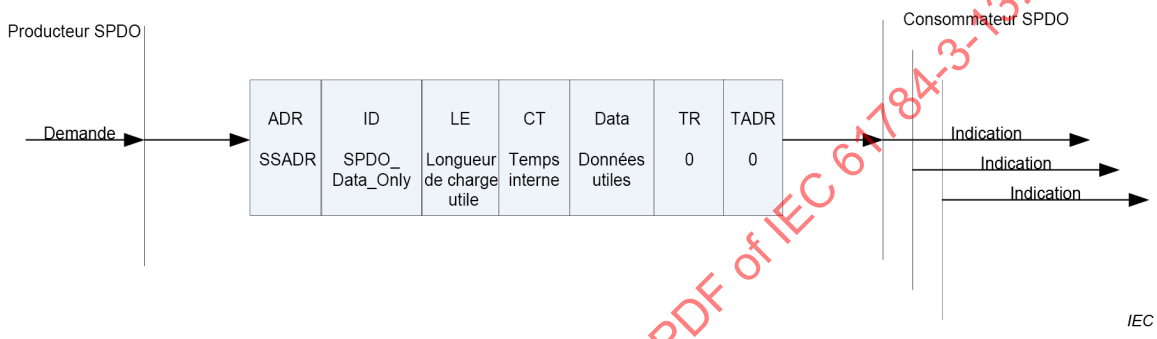


Figure 25 – Télégramme SPDO_Data_Only

Le Tableau 13 spécifie les champs d'un télégramme SPDO_Data_Only.

Tableau 13 – Champs du télégramme SPDO_Data_Only

Champ	Information	Contenu/Valeur
ADR	Adresse du producteur de SPDO	SSADR
ID	SPDO_Data_Only	11000xxxb
LE	Nombre d'octets de données utiles	0 – 240 (voir 7.1.4)
CT	Valeur de temps interne du producteur de SPDO	Temps
Données	Données utiles	Données
TR	Non utilisé/Demande de fonctionnalité (voir 7.6.6.2)	0
TADR	Non utilisé	0

7.2.4 Données avec télégramme de demande de temps

Les données avec télégramme de demande de temps (voir Figure 26) doivent être utilisées pour transmettre les données de processus d'un SN (producteur SPDO) avec une demande de synchronisation temporelle à un autre SN indiqué par TADR. Les données peuvent être consommées par un autre SN (consommateur de SPDO).

En même temps que les données (Data), le producteur SPDO fournit son adresse SN (ADR), le type de télégramme (ID) ainsi que le nombre d'octets de données utiles (LE) transmis, la valeur courante de son temporisateur interne (CT), une valeur de compteur qui indique le nombre de demandes de temps (TR) et l'adresse du SN auquel il est demandé de fournir son temps interne courant (TADR).

Dans une communication SPDO avec des valeurs CT sur 40 bits, le champ ID de la partie 2 du PDU doit être codé par une opération XOR avec la confirmation de la demande de fonctionnalité SPDO par le producteur de SPDO, qui reproduit les informations de la demande.

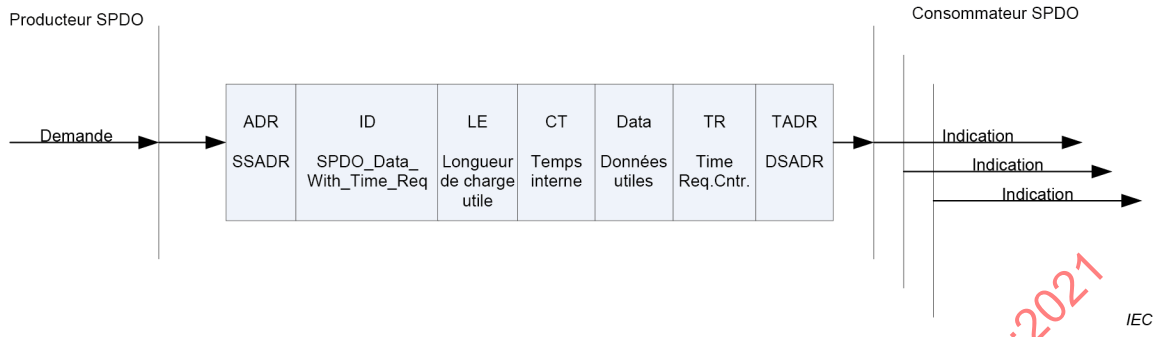


Figure 26 – Télégramme SPDO_Data_with_Time_Request

Le Tableau 14 spécifie les champs d'un télégramme SPDO_Data_with_Time_Request.

Tableau 14 – Champs du télégramme SPDO_Data_with_Time_Request

Champ	Information	Contenu/Valeur
ADR	Adresse du producteur de SPDO	SSADR
ID	SPDO_Data_with_Time_Request	11001xxxb
LE	Nombre d'octets de données utiles	0 – 240 (voir 7.1.4)
CT	Valeur de temps interne de SN	Temps
Données	Données utiles	Données
TR	Compteur de demande de temps interne	0 – 63
TADR	Adresse du SN dont les informations temporelles sont demandées	DSADR

7.2.5 Données avec télégramme de réponse de temps

Les données avec télégramme de réponse de temps (voir Figure 27) doivent être utilisées pour transmettre des données de processus d'un SN (producteur SPDO) en même temps que la valeur du temporisateur interne du nœud concerné. Les données peuvent être consommées par un autre SN (consommateur de SPDO).

En même temps que les données (Data), le producteur SPDO fournit son adresse SN (ADR), le type de télégramme (ID) ainsi que le nombre d'octets de données utiles (LE) transmis, la valeur courante de son temporisateur interne (CT), la valeur du compteur qui indique le nombre de demandes de temps (TR) et l'adresse du SN qui a émis la demande (TADR).

Dans une communication SPDO avec des valeurs CT sur 40 bits, aucun encodage supplémentaire n'a lieu dans la trame.

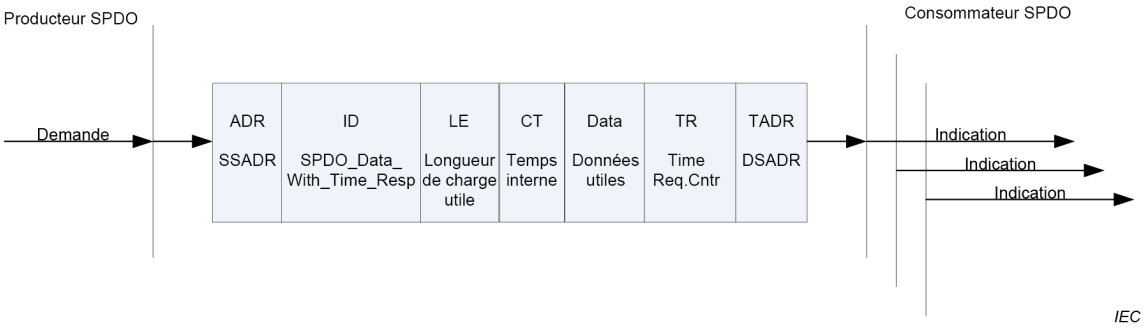


Figure 27 – Télégramme SPDO_Data_with_Time_Response

Le Tableau 15 spécifie les champs d'un télégramme SPDO_Data_with_Time_Response.

Tableau 15 – Champs du télégramme SPDO_Data_with_Time_Response

Champ	Information	Contenu/Valeur
ADR	Adresse du producteur de SPDO	SSADR
ID	SPDO_Data_with_Time_Response	11010xxxb
LE	Nombre d'octets de données utiles	0 – 240 (voir 7.1.4)
CT	Valeur de temps interne du producteur de SPDO	Temps
Données	Données utiles	Données
TR	Valeur du compteur de demande de temps pour demande correspondante	0 – 63
TADR	Adresse du SN qui a émis la demande d'informations temporelles	DSADR

7.3 Objet de données de service de sécurité (SSDO)

7.3.1 Généralités

Les objets de données de service de sécurité (SSDO) doivent être utilisés pour l'accès au dictionnaire d'objets (SOD) d'un serveur SSDO par un client SSDO.

Les télégrammes SSDO doivent être identifiés par les bits 7, 6 et 5 (111b) de l'ID. Le bit 4 de l'ID doit identifier un accès au SOD (0b). Le bit 3 doit identifier le type de PDU de sécurité utilisé qui doit être soit un PDU de sécurité de base (0b) soit un PDU de sécurité Slim (1b) en fonction du type de service SSDO. Le bit 2 doit établir une distinction entre les télégrammes de demande (0b) et de réponse (1b).

7.3.2 Types de télégrammes SSDO

Le champ ID identifie le télégramme. Les types de télégrammes suivants sont spécifiés:

– Demande de service SSDO
– Réponse de service SSDO

Le Tableau 16 spécifie le codage du champ ID.

Tableau 16 – Types de télégrammes SSDO (bits 2, 3 et 4 du champ ID)

Numéro de bit du champ ID								Type de télégramme
7	6	5	4	3	2	1	0	
1	1	1	0	0	0			SSDO_Service_Request
1	1	1	0	0	1			SSDO_Service_Response
1	1	1	0	1	0			SSDO_Service_Request_Slim
1	1	1	0	1	1			SSDO_Service_Response_Slim

Le télégramme de demande de service SSDO (SSDO_Service_Request) doit être utilisé par un client SSDO pour accéder au SOD d'un serveur SSDO. Avec une demande SSDO_Service_Request, un client SSDO doit fournir l'adresse SN du serveur (ADR), la longueur de données utiles (LE) et le numéro de la demande d'accès SANo (champ CT). Le premier octet de données utiles doit être utilisé comme l'"Octet Commande" (SACmd, voir Tableau 17) et doit spécifier le type d'accès (lecture/écriture), l'état du transfert (réussite/abandon), le type de transfert (accéléré/segmenté), le bit de bascule, "initialiser" ou dernier segment du transfert et le mode de transfert (normal/bloc). L'index de l'entrée d'accès au dictionnaire d'objets doit être spécifié dans les octets de données 1 et 2; le sous-index de l'entrée d'accès doit être spécifié dans l'octet de données 3. L'adresse SN du client SSDO doit être fournie dans TADR.

Le télégramme de réponse de service SSDO (SSDO_Service_Response) doit être utilisé par un serveur SSDO pour répondre à une demande d'un client SSDO. Le numéro de demande d'accès au SOD (SANo) doit être unique pour chaque connexion et être incrémenté de 1 à chaque nouveau télégramme de demande d'accès au SOD. Le télégramme de réponse doit contenir la valeur SANo de la demande de service correspondante.

La longueur maximale de données utiles doit être de 240 octets.

Le codage des bits de la commande d'accès au SOD (SACmd) est spécifié dans le Tableau 17.

Tableau 17 – Codage binaire de la commande d'accès au SOD (SACmd)

Numéro de bit	Nom	Valeur	Signification
0	Type d'accès	0	Accès au SOD en lecture
		1	Accès au SOD en écriture
1	Préchargement	0	Aucun préchargement dans le transfert SSDO (par défaut)
		1	Activation du préchargement dans le transfert SSDO
2	Abandon du transfert	0	Transmission réussie
		1	Abandon du transfert
3	Segmentation	0	Accès accéléré au SOD
		1	Accès segmenté au SOD
4	Bascule	0	Ce bit doit changer pour chaque segment suivant et ne doit pas changer de valeur s'il s'agit d'une répétition de la transmission. Le bit de bascule du premier segment doit être mis à 0. Le bit de bascule doit être identique pour les télégrammes de demande et de réponse.
		1	
5	Lancer le transfert	0	Ne pas lancer le transfert
		1	Lancer le transfert
6	Transfert du segment de fin	0	Il reste des segments à transférer
		1	Il ne reste plus de segments à transférer

Numéro de bit	Nom	Valeur	Signification
7	Transfert de blocs (n'est plus utilisé)	0	Réservé

7.3.3 Services et protocoles SSDO

Deux types de services SDO utilisent différents PDU de sécurité:

– le type de service SSDO normal qui utilise le PDU de sécurité de base peut être utilisé pour tous les services SSDO;

– le type de service SSDO slim qui utilise le PDU de sécurité Slim ne peut être utilisé que pour télécharger les données d'initialisation vers un SN. Si les données ne se trouvent pas deux fois dans le PDU de sécurité, une vérification des données doit être effectuée. Cela doit être effectué uniquement en passant le SN à l'état opérationnel si le CRC du SOD est correct.

Les services SSDO suivants doivent être définis:

– téléchargement aval SSDO, divisé en:
  - lancer téléchargement aval SSDO;
  - lancer téléchargement aval SSDO avec préchargement;
  - segmenter téléchargement aval SSDO;
  - segmenter téléchargement aval SSDO avec préchargement;
  - segment de fin de SSDO;
– téléchargement amont SSDO, divisé en:
  - lancer téléchargement amont SSDO;
  - lancer téléchargement amont SSDO avec préchargement;
  - segmenter téléchargement amont SSDO;
  - segmenter téléchargement amont SSDO avec préchargement;
  - segment de fin de SSDO;
– abandonner SSDO.

La Figure 28 définit les protocoles de téléchargement aval SSDO segmenté et accéléré, ainsi que leurs homologues avec préchargement.

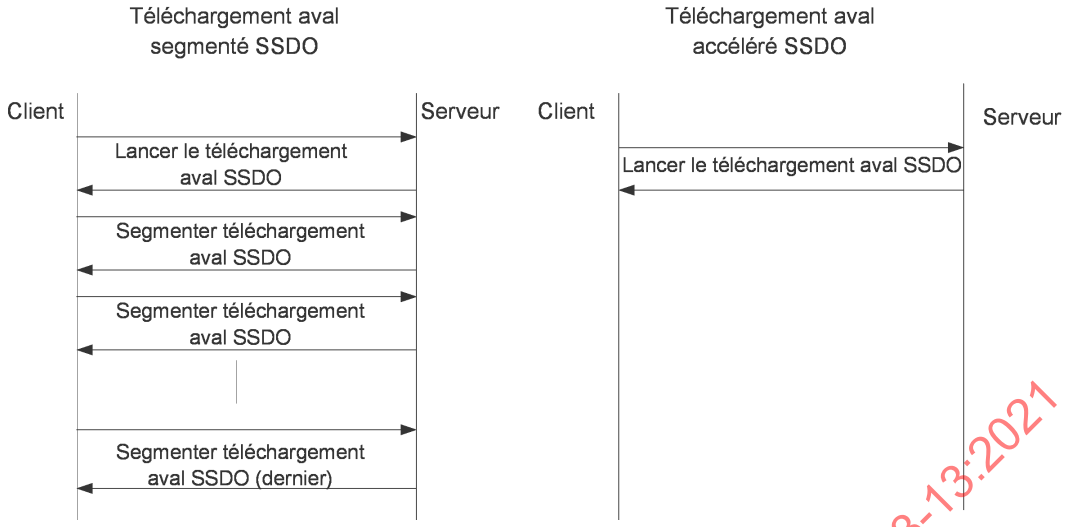


Figure 28 – Protocoles de téléchargement aval SSDO

La Figure 29 définit les protocoles de téléchargement amont SSDO segmenté et accéléré.

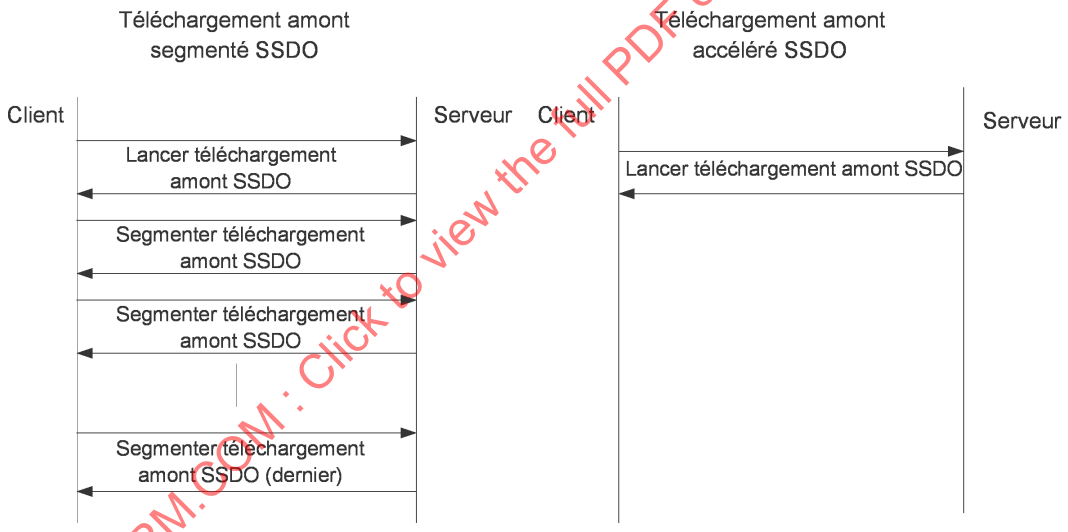


Figure 29 – Protocoles de téléchargement amont SSDO

7.3.4 Lancer téléchargement aval SSDO

La Figure 30 spécifie le protocole Lancer téléchargement aval.

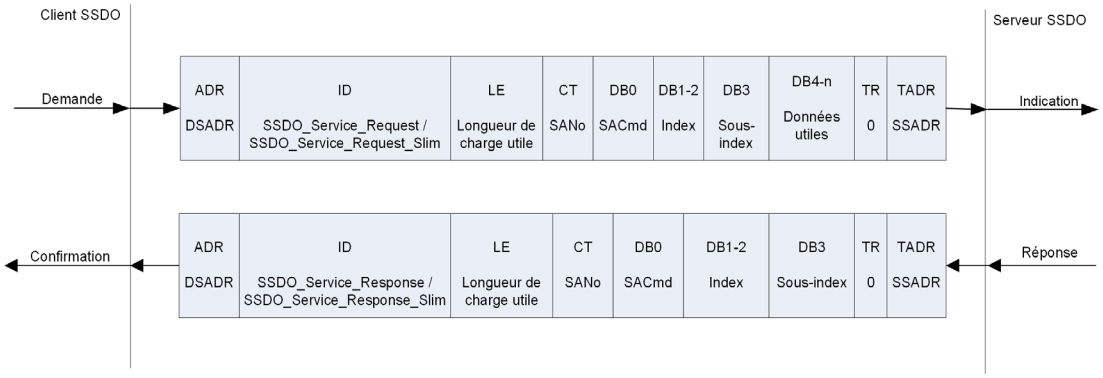


Figure 30 – Protocole Lancer téléchargement aval SSDO

Le Tableau 18 spécifie les champs d'un télégramme SSDO_Service_Request Lancer téléchargement aval.

Tableau 18 – Champs du télégramme SSDO_Service_Request Lancer téléchargement aval

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Longueur de données utiles	4 – 240 (voir 7.1.4)
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non autorisé
DB0	Commande d'accès au SOD (SACmd) accéléré ou segmenté	00100001b ou 00101001b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
DB4 – DBn	Données utiles (en cas de téléchargement aval segmenté, DB4 à DB7 contient la taille de données utiles à télécharger. Si la taille des données est 0, elle n'est pas indiquée.)	Données utiles
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 19 spécifie les champs d'un télégramme SSDO_Service_Response Lancer téléchargement aval.

Tableau 19 – Champs du télégramme SSDO_Service_Response Lancer téléchargement aval

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Longueur de données utiles	4
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non autorisé
DB0	Commande d'accès au SOD (SACmd) accéléré ou segmenté	00100001b ou 00101001b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
TR	Non utilisé	0
TADR	Adresse du serveur SSDO	SSADR

7.3.5 Segmenter téléchargement aval SSDO

La Figure 31 spécifie le protocole Segmenter téléchargement aval.

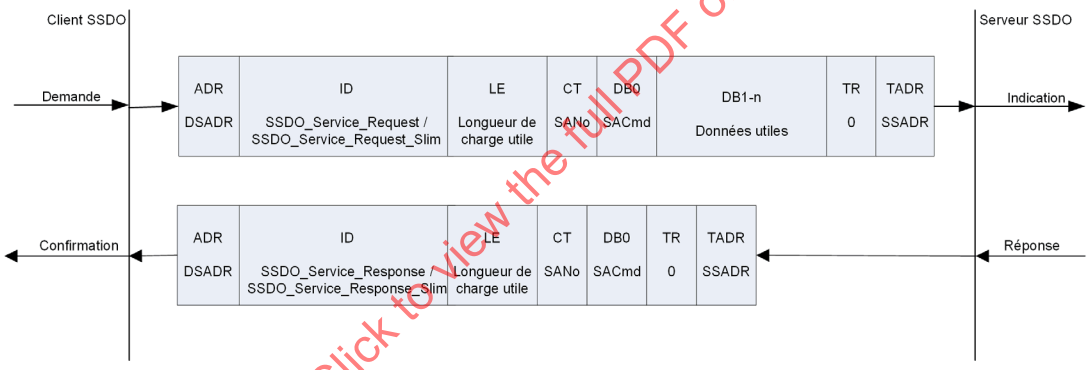


Figure 31 – Protocole Segmenter téléchargement aval SSDO

Le Tableau 20 spécifie les champs d'un télégramme SSDO_Service_Request Segmenter téléchargement aval.

Tableau 20 – Champs du télégramme SSDO_Service_Request Segmenter téléchargement aval

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Longueur de données utiles	1 – 240 (voir 7.1.4)
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non autorisé
DB0	Segment de milieu ou de fin de la commande d'accès au SOD (SACmd)	000x1001b ou 010x1001b
DB1 – DBn	Données utiles	Données utiles
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 21 spécifie les champs d'un télégramme SSDO_Service_Response Segmenter téléchargement aval.

Tableau 21 – Champs du télégramme SSDO_Service_Response Segmenter téléchargement aval

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Longueur de données utiles	1
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non autorisé
DB0	Segment de milieu ou de fin de la commande d'accès au SOD (SACmd)	000x1001b ou 010x1001b
TR	Non utilisé	0
TADR	Adresse du serveur SSDO	SSADR

7.3.6 Lancer téléchargement aval SSDO avec préchargement

La Figure 32 spécifie le protocole Lancer téléchargement aval avec préchargement.

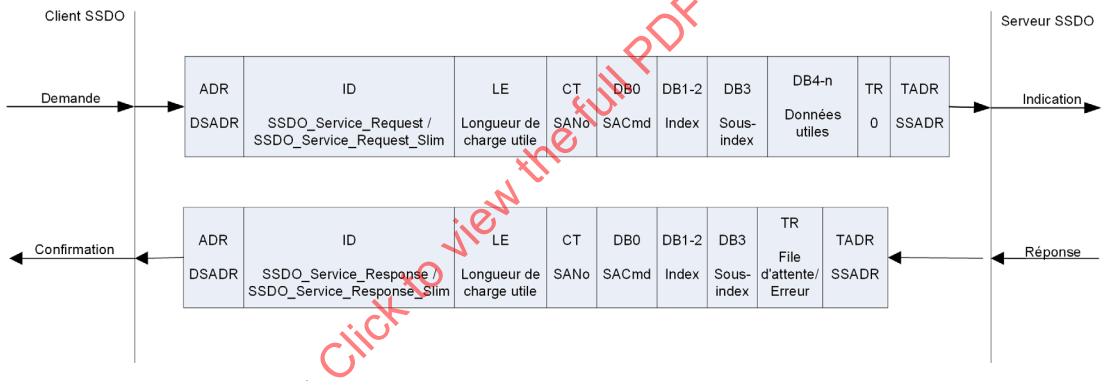


Figure 32 – Protocole Lancer téléchargement aval SSDO avec préchargement

Le Tableau 22 spécifie les champs d'un télégramme SSDO_Service_Request Lancer téléchargement aval avec préchargement.

Tableau 22 – Champs du télégramme SSDO_Service_Request Lancer téléchargement aval avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Longueur de données utiles	4 – 240 (voir 7.1.4)
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non autorisé
DB0	Commande d'accès au SOD (SACmd) accéléré ou segmenté avec préchargement	00100011b ou 00101011b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
DB4 – DBn	Données utiles (en cas de téléchargement aval segmenté, DB4 à DB7 contient la taille de données utiles à télécharger. Si la taille des données est 0, elle n'est pas indiquée.)	Données utiles

Champ	Information	Contenu/Valeur
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 23 spécifie les champs d'un télégramme SSDO_Service_Response Lancer téléchargement aval avec préchargement.

Tableau 23 – Champs du télégramme SSDO_Service_Response Lancer téléchargement aval avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Longueur de données utiles	4
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non autorisé
DB0	Commande d'accès au SOD (SACmd) accéléré ou segmenté	00100011b ou 00101011b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
TR	Bits 2 à 5: Taille de la file d'attente Bits 6 à 7: Indicateur d'erreur	0 – 15 0xC
TADR	Adresse du serveur SSDO	SSADR

7.3.7 Segmenter téléchargement aval SSDO avec préchargement

La Figure 33 spécifie le protocole Segmenter téléchargement aval avec préchargement.

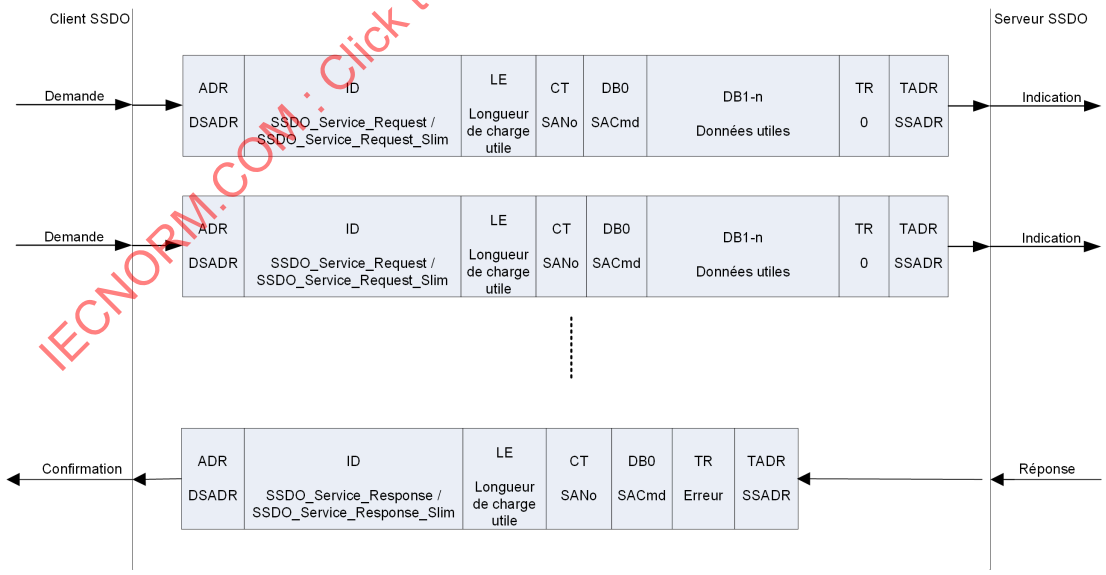


Figure 33 – Protocole Segmenter téléchargement aval SSDO avec préchargement

Le Tableau 24 spécifie les champs d'un télégramme SSDO_Service_Request Segmenter téléchargement aval avec préchargement.

Tableau 24 – Champs du télégramme SSDO_Service_Request Segmenter téléchargement aval avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request SSDO_Service_Request_Slim	111000xxb 111010xxb
LE	Longueur de données utiles	1 – 240 (voir 7.1.4)
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non autorisé
DB0	Segment de milieu ou de fin de la commande d'accès au SOD (SACmd)	000x1011b ou 010x1011b
DB1 – DBn	Données utiles	Données utiles
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 25 spécifie les champs d'un télégramme SSDO_Service_Response Segmenter téléchargement aval avec préchargement.

Tableau 25 – Champs du télégramme SSDO_Service_Response Segmenter téléchargement aval avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response SSDO_Service_Response_Slim	111001xxb 111011xxb
LE	Longueur de données utiles	1
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non autorisé
DB0	Segment de milieu ou de fin de la commande d'accès au SOD (SACmd)	000x1011b ou 010x1011b
TR	Indicateur d'erreur	Erreur: 1xxxxxxxb Succès: 0xxxxxxxb
TADR	Adresse du serveur SSDO	SSADR

7.3.8 Lancer téléchargement amont SSDO

La Figure 34 spécifie le protocole Lancer téléchargement amont SSDO. Le télégramme Lancer téléchargement amont doit toujours être en accéléré. Selon la réponse reçue, le client SSDO doit poursuivre avec un transfert accéléré ou segmenté.

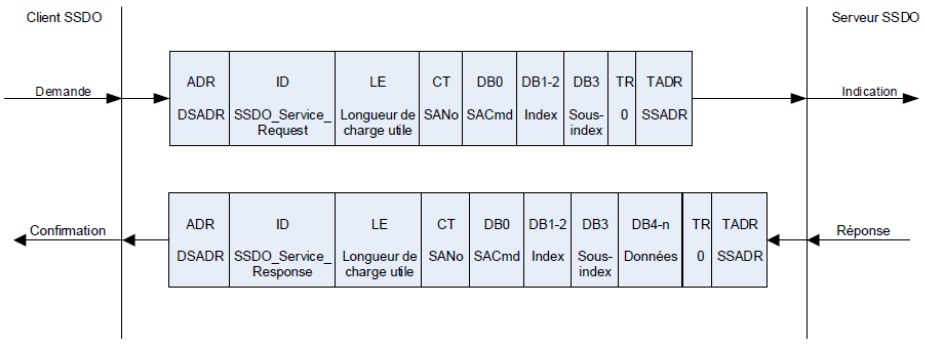


Figure 34 – Protocole Lancer téléchargement amont SSDO

Le Tableau 26 spécifie les champs d'un télégramme SSDO_Service_Request Lancer téléchargement amont.

Tableau 26 – Champs du télégramme SSDO_Service_Request Lancer téléchargement amont

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Longueur de données utiles	4
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non admis
DB0	Commande d'accès au SOD (SACmd) en accéléré	00100000b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 27 spécifie les champs d'un télégramme SSDO_Service_Response Lancer téléchargement amont.

Tableau 27 – Champs du télégramme SSDO_Service_Response Lancer téléchargement amont

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Longueur de données utiles	4 – 240 (voir 7.1.4)
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non admis
DB0	Commande d'accès au SOD (SACmd) accéléré ou segmenté	00100000b ou 00101000b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
DB4 – DBn	Données utiles (en cas de téléchargement amont segmenté, DB4 à DB7 contient la taille de données utiles du bloc. Si la taille des données est 0, elle n'est pas indiquée.)	Données utiles
TR	Non utilisé	0
TADR	Adresse du serveur SSDO	SSADR

7.3.9 Segmenter téléchargement amont SSDO

La Figure 35 définit le protocole Segmenter téléchargement amont. Le client SSDO demande toujours un segment de milieu. Sur la base de la réponse que le client reçoit du serveur SSDO, le client SSDO doit décider si le serveur SSDO a envoyé un segment de milieu ou de fin.

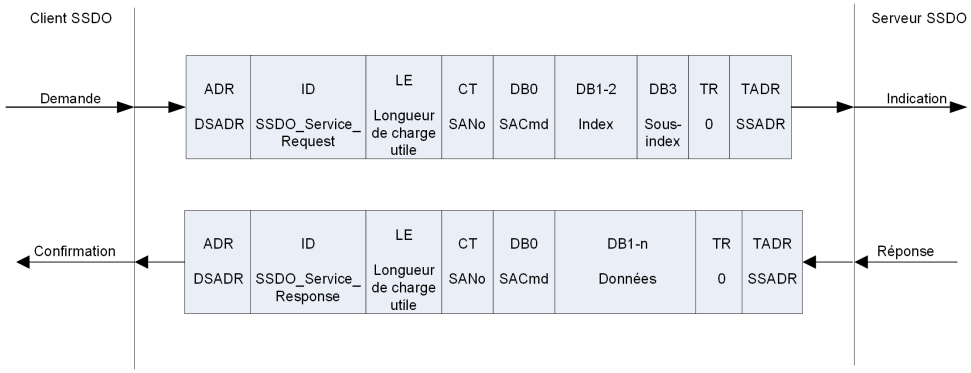


Figure 35 – Protocole Segmenter téléchargement amont SSDO

Le Tableau 28 spécifie les champs d'un télégramme SSDO_Service_Request Segmenter téléchargement amont.

Tableau 28 – Champs du télégramme SSDO_Service_Request Segmenter téléchargement amont

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Longueur de données utiles	4
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non admis
DB0	Segment du milieu de la commande d'accès au SOD (SACmd)	000x1000b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 29 spécifie les champs d'un télégramme SSDO_Service_Response Segmenter téléchargement amont.

Tableau 29 – Champs du télégramme SSDO_Service_Response Segmenter téléchargement amont

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Longueur de données utiles	1 – 240 (voir 7.1.4)
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non admis
DB0	Segment de milieu ou de fin de la commande d'accès au SOD (SACmd)	000x1000b ou 010x1000b
DB1 – DBn	Données utiles	Données utiles
TR	Non utilisé	0
TADR	Adresse du serveur SSDO	SSADR

7.3.10 Lancer téléchargement amont SSDO avec préchargement

La Figure 36 spécifie le protocole Lancer téléchargement amont de bloc avec préchargement. Le télégramme Lancer téléchargement amont de bloc doit toujours être en accéléré. Selon la réponse reçue, le client SSDO doit poursuivre avec un transfert accéléré ou segmenté. La taille de la file d'attente transmise par le serveur doit représenter le nombre maximal de demandes que le serveur peut mettre en file d'attente pour le téléchargement amont.

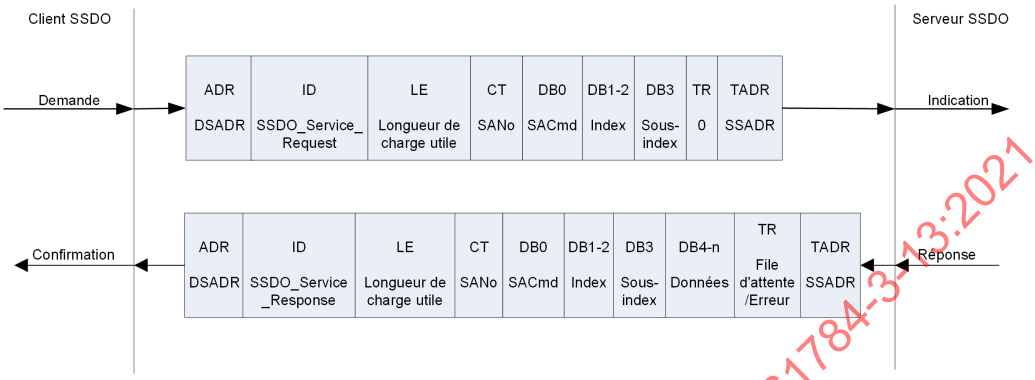


Figure 36 – Protocole Lancer téléchargement amont SSDO avec préchargement

Le Tableau 30 spécifie les champs d'un télégramme SSDO_Service_Request Lancer téléchargement amont avec préchargement.

Tableau 30 – Champs du télégramme SSDO_Service_Request Lancer téléchargement amont avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à atteindre	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Longueur de données utiles	4
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non admis
DB0	Commande d'accès au SOD (SACmd) en accéléré	00100010b
DB1, DB2	Index de l'entrée SOD à atteindre	0 à 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 à 255
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 31 spécifie les champs d'un télégramme SSDO_Service_Response Lancer téléchargement amont avec préchargement.

Tableau 31 – Champs du télégramme SSDO_Service_Response Lancer téléchargement amont avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Longueur de données utiles	4 – 240 (voir 7.1.4)
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non admis
DB0	Commande d'accès au SOD (SACmd) accéléré ou segmenté	00100010b ou 00101010b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
DB4 – DBn	Données utiles (en cas de téléchargement amont segmenté, DB4 à DB7 contient la taille de données utiles du bloc. Si la taille des données est 0, elle n'est pas indiquée.)	Données utiles
TR	Bits 2 à 5: Taille de la file d'attente Bits 6 à 7: Indicateur d'erreur	0 – 15 0xC
TADR	Adresse du serveur SSDO	SSADR

7.3.11 Segmenter téléchargement amont SSDO avec préchargement

La Figure 37 spécifie le protocole Segmenter téléchargement amont de bloc. Le client SSDO demande toujours un segment de milieu. Sur la base de la réponse que le client reçoit du serveur SSDO, le client SSDO doit décider si le serveur SSDO a envoyé un segment de milieu ou de fin.

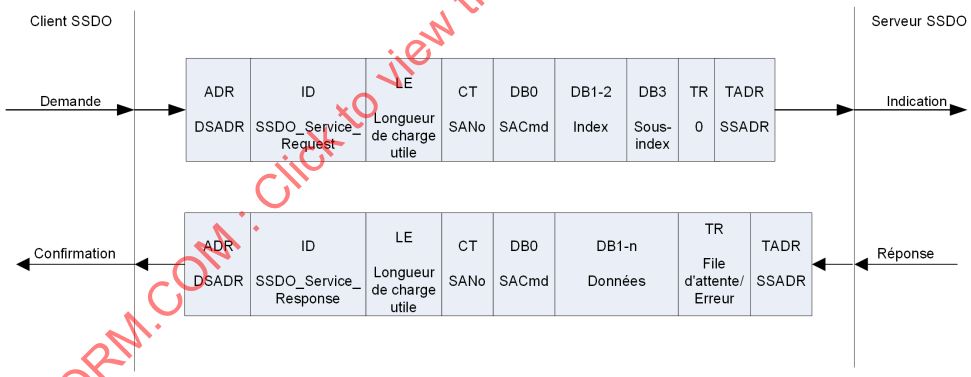


Figure 37 – Protocole Segmenter téléchargement amont SSDO avec préchargement

Le Tableau 32 spécifie les champs d'un télégramme SSDO_Service_Request Segmenter téléchargement amont avec préchargement.

Tableau 32 – Champs du télégramme SSDO_Service_Request Segmenter téléchargement amont avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du serveur SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Request	111000xxb
LE	Longueur de données utiles	4
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non admis
DB0	Segment du milieu de la commande d'accès au SOD (SACmd)	000x1010b
DB1, DB2	Index de l'entrée SOD à atteindre	0 – 65 535
DB3	Sous-index de l'entrée SOD à atteindre	0 – 255
TR	Non utilisé	0
TADR	Adresse du client SSDO	SSADR

Le Tableau 33 spécifie les champs d'un télégramme SSDO_Service_Response Segmenter téléchargement amont avec préchargement.

Tableau 33 – Champs du télégramme SSDO_Service_Response Segmenter téléchargement amont avec préchargement

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à qui adresser la réponse	DSADR
ID	SSDO_Service_Response	111001xxb
LE	Longueur de données utiles	4 – 240 (voir 7.1.4)
CT	Echo du numéro de demande d'accès au SOD	1 à 65 535, 0 non admis
DB0	Segment de milieu ou de fin de la commande d'accès au SOD (SACmd)	000x1010b ou 010x1010b
DB1 – DBn	Données utiles	Données utiles
TR	Indicateur d'erreur	Erreur: 1xxxxxxxb Succès: 0xxxxxxxb
TADR	Adresse du serveur SSDO	SSADR

7.3.12 Abandonner SSDO

La Figure 38 spécifie le protocole Abandonner SSDO.

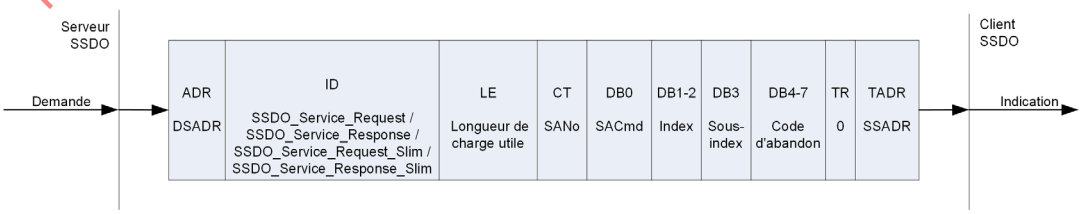


Figure 38 – Protocole Abandonner SSDO

Le Tableau 34 spécifie les champs d'un télégramme Abandonner SSDO.

Tableau 34 – Champs du télégramme Abandonner SSDO

Champ	Information	Contenu/Valeur
ADR	Adresse du client SSDO à informer	DSADR
ID	SSDO_Service_Request SSDO_Service_Response SSDO_Service_Request_Slim SSDO_Service_Response_Slim	111000xxb 111000xxb 111010xxb 111011xxb
LE	Longueur de données utiles	8
CT	Numéro de demande d'accès au SOD (SANo)	1 à 65 535, 0 non admis
DB0	Commande d'accès au SOD (SACmd)	00000100b
DB1, DB2	Index de l'entrée SOD	0 – 65 535
DB3	Sous-index de l'entrée SOD	0 – 255
DB4 – DB7	Code d'abandon (voir Tableau 35)	Code d'abandon
TR	Non utilisé	0
TADR	Adresse du serveur SSDO qui a initié l'abandon	SSADR

Les données utiles d'un télégramme Abandonner SSDO doivent contenir un code d'erreur comme spécifié dans le Tableau 35. Le code d'abandon doit être encodé en UNSIGNED32. Les codes d'abandon qui ne sont pas indiqués dans le Tableau 35 doivent être réservés.

Tableau 35 – Codes d'abandon SSDO

Code d'abandon [a]	Description
05030000h	Réservé
05040000h	Expiration du délai de protocole SSDO
05040001h	ID de commande client/serveur non valide ou inconnu
05040002h	Taille de bloc non valide
05040003h	Numéro de séquence non valide
05040004h	Réservé
05040005h	Mémoire insuffisante
06010000h	Accès vers un objet non pris en charge
06010001h	Tentative de lecture d'un objet en écriture seule
06010002h	Tentative d'écriture d'un objet en lecture seule
06020000h	L'objet n'existe pas dans le dictionnaire d'objets
06040041h	L'objet ne peut pas être mappé avec le SPDO
06040042h	Le nombre et la longueur des objets à mapper dépasseraient la longueur du SPDO
06040043h	Incompatibilité générale du paramètre
06040047h	Incompatibilité interne générale dans l'appareil
06060000h	Echec de l'accès dû à une erreur matérielle
06070010h	Le type de données ne correspond pas, la longueur du paramètre de service ne correspond pas
06070012h	Le type de données ne correspond pas, la longueur du paramètre de service est trop importante
06070013h	Le type de données ne correspond pas, la longueur du paramètre de service est trop courte
06090011h	Le sous-index n'existe pas
06090030h	La plage de valeurs du paramètre est dépassée (uniquement pour l'accès en écriture)
06090031h	Valeur trop élevée du paramètre écrit

Code d'abandon [a]	Description
06090032h	Valeur trop faible du paramètre écrit
06090036h	La valeur maximale est inférieure à la valeur minimale
08000000h	Erreur générale
08000020h	Les données ne peuvent pas être transmises ou enregistrées dans l'application
08000021h	Les données ne peuvent pas être transmises ou enregistrées dans l'application en raison du contrôle local
08000022h	Les données ne peuvent pas être transmises ou enregistrées dans l'application en raison de l'état actuel de l'appareil.
08000023h	Les données ne peuvent pas être transmises ou enregistrées dans l'application, car l'application est occupée
[a] Comme indiqué par le suffixe "h", le code d'abandon est exprimé en notation hexadécimale.	

7.3.13 Préchargement SSDO

Le préchargement est une méthode, où le serveur SSDO informe le client SSDO de la taille du tampon de message entrant, ce qui permet au client d'envoyer des messages jusqu'à ce que le tampon soit rempli. Les canaux de transport asynchrones sont ainsi mieux utilisés.

Le préchargement SSDO est une fonctionnalité facultative. Si la fonctionnalité est demandée par le client, mais qu'elle n'est pas prise en charge par le serveur, le client et le serveur doivent utiliser les techniques de transfert segmenté classiques pour échanger des données.

Le préchargement SSDO doit utiliser le champ TR pour transporter les informations du serveur SSDO vers le client en ce qui concerne la taille du tampon ainsi que les messages manquants.

Pour le préchargement SSDO, le champ TR doit être utilisé comme spécifié dans le Tableau 36.

Tableau 36 – Utilisation du champ TR pour le préchargement SSDO

Numéros de bit du champ TR							
7	6	5	4	3	2	1	0
Erreur		Taille du tampon du serveur				Utilisés par la TADR	

Le Tableau 37 spécifie les champs de bits dans le champ TR utilisé par le préchargement SSDO.

Tableau 37 – Champs de bits dans le champ TR pour le préchargement SSDO

Champ	Information
Erreur	Les bits 6 et 7 doivent être mis à la valeur maximale (0xCx) si le paquet reçu par le serveur n'est pas celui attendu, qui est déterminé par la numérotation SANo. Le numéro SANo de la trame dans laquelle les bits 6-7 sont définis à la valeur maximale doit être le numéro du dernier paquet reçu correctement. Le client doit retransmettre les trames à partir de ce point, en commençant par la prochaine trame après ce numéro SANo.
Taille du tampon du serveur	Les bits 2 à 5 sont déterminés par la réponse à l'initialisation et définissent la taille du tampon côté serveur. La valeur donnée doit commencer par 0, où 0 est définie comme une taille tampon de 1, ce qui permet une taille tampon maximale de 16.

7.4 Gestion de réseau de sécurité (SNMT)

7.4.1 Généralités

La gestion de réseau de sécurité (SNMT) repose sur les nœuds et suit une structure logique maître/esclave. Le maître SNMT doit être mis en œuvre par un SCM ou un outil de configuration. Les services SNMT suivants doivent être fournis:

– demande d'UDID;
– attribution d'adresse de nœud;
– services étendus qui doivent être désignés par le champ DB0 qui comporte les services suivants:
  • contrôle de l'état de communication;
  • sauvegarde du nœud;
  • attribution d'adresse de nœud supplémentaire.

Seuls les PDU de sécurité de base doivent être utilisés pour la SNMT.

7.4.2 Types de télégrammes SNMT

Le champ ID (bits 2 à 4) doit identifier le type de télégramme SNMT. Le Tableau 38 spécifie les types de télégrammes SNMT:

Tableau 38 – Types de télégrammes SNMT (bits 2, 3 et 4 du champ ID)

Numéro de bit du champ ID								Type de télégramme	Service
7	6	5	4	3	2	1	0		
1	0	1	0	0	0			SNMT_Request_UDID	Demande d'UDID
1	0	1	0	0	1			SNMT_Response_UDID	
1	0	1	0	1	0			SNMT_Assign_SADR	Attribution d'adresse de nœud
1	0	1	0	1	1			SNMT_SADR_assigned	
1	0	1	1	0	0			SNMT_Service_Request	Services étendus
1	0	1	1	0	1			SNMT_Service_Response	
1	0	1	1	1	1			SNMT_SN_reset_guarding_SCM	Délai d'expiration de la réinitialisation de sauvegarde

7.4.3 Services et protocoles SNMT

7.4.3.1 Demande/réponse d'UDID

La demande/réponse d'UDID doit permettre à un maître SNMT de récupérer l'UDID d'un esclave SNMT spécifique. Pour cela, le maître SNMT doit envoyer la demande d'UDID à un esclave SNMT spécifique. L'esclave SNMT doit répondre même si la valeur dans l'adresse ADR ne correspond pas à la SADR interne des SN.

NOTE 1 Cela peut se produire si aucune SADR n'a été préalablement attribuée au SN.

Le champ TADR de la réponse doit contenir la SADR de la demande d'UDID.

NOTE 2 Si le SN n'a pas de SADR valide et envoie sa propre SADR, la réponse ne peut être attribuée correctement à la demande correspondante.

Une erreur d'exécution de ce protocole doit être indiquée par un UDID 00-00-00-00-00-00 dans la réponse.

La Figure 39 spécifie le protocole de demande d'UDID.

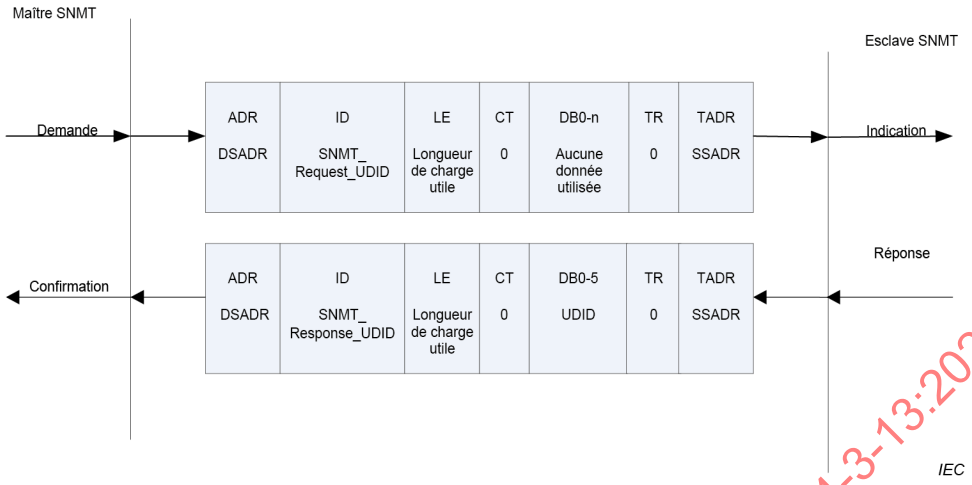


Figure 39 – Protocole Demande/réponse d'UDID

Le Tableau 39 spécifie les champs d'un télégramme SNMT_Request_UDID.

Tableau 39 – Champs du télégramme SNMT_Request_UDID

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT de destination	DSADR
ID	SNMT_Request_UDID	101000xxb
LE	Longueur de données utiles	0
CT	Non utilisé	0
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 40 spécifie les champs d'un télégramme SNMT_Response_UDID.

Tableau 40 – Champs du télégramme SNMT_Response_UDID

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	SNMT_Response_UDID	101001xxb
LE	Longueur de données utiles	6
CT	Non utilisé	0
DB0 – DB5	UDID du SN	UDID
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.4.3.2 Attribution de SADR

L'attribution d'une SADR doit être utilisée par un maître SNMT pour établir la SADR d'un esclave SNMT spécifique. Le protocole doit en outre attribuer le numéro de domaine de sécurité (SDN) à l'esclave SNMT. Le SDN doit être établi au sein du champ d'adresse de la partie 2 du PDU en utilisant une opération XOR logique (voir 7.1). Un SN doit envoyer une réponse uniquement si l'UDID reçu correspond à l'UDID propre au SN.

Si l'information SDN/SADR demandée par le maître SNMT ne peut pas être attribuée, l'esclave SNMT doit répondre au moyen de sa SADR/son SDN actuel. Le Maître SNMT doit alors comparer le SDN/la SADR et l'UDID de la réponse au SDN/à la SADR et à l'UDID définis afin de vérifier que l'attribution a abouti.

NOTE Des échecs d'attribution peuvent se produire si la SADR/le SDN des nœuds est codé en fixe au moyen d'un commutateur et que la SADR/le SDN ne correspond pas à la valeur SCM.

Une erreur d'exécution de ce service doit être indiquée par un UDID 00-00-00-00-00-00 dans la réponse.

La Figure 40 spécifie le protocole d'attribution d'une SADR.

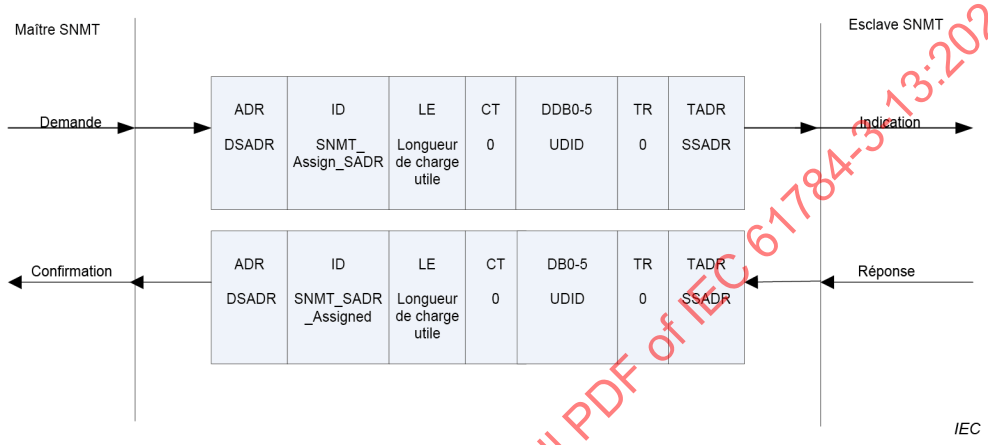


Figure 40 – Protocole Attribution d'une SADR

Le Tableau 41 spécifie les champs d'un télégramme SNMT_Assign_SADR.

Tableau 41 – Champs du télégramme SNMT_Assign_SADR

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à attribuer	DSADR
ID	SNMT_Assign_SADR	101010xxb
LE	Longueur de données utiles	6
CT	Non utilisé	0
DB0 – DB5	UDID du SN	UDID
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 42 spécifie les champs d'un télégramme SNMT_SADR_Assigned.

Tableau 42 – Champs du télégramme SNMT_SADR_Assigned

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	SNMT_SADR_Assigned	101011xxb
LE	Longueur de données utiles	6
CT	Non utilisé	0
DB0 – DB5	UDID du SN	UDID
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.4.3.3 Réinitialiser intervalle de sauvegarde de nœud

Le service Réinitialiser intervalle de sauvegarde de nœud doit être utilisé par le SN pour déclencher une sauvegarde du nœud sur le SCM. Le SCM doit sauvegarder tous les SN, qu'il y ait ou non correspondance entre la SADR ou le SDN du télégramme et la SADR et le SDN du SCM.

Le télégramme doit toujours être envoyé avec la SADR et le SDN à 1.

NOTE Cela s'explique par le fait que le télégramme est envoyé avant attribution d'une SADR ou d'un SDN au nœud.

La Figure 41 spécifie le protocole Réinitialiser intervalle de sauvegarde de nœud.

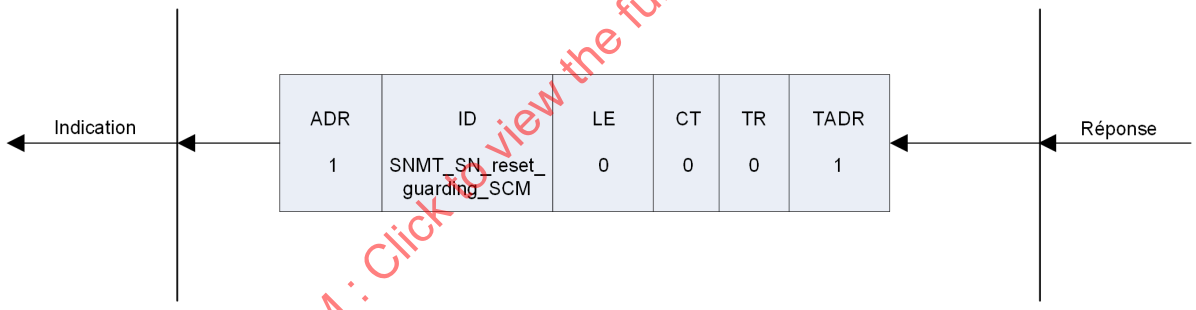


Figure 41 – Protocole Réinitialiser intervalle de sauvegarde de nœud

Le Tableau 43 spécifie les champs d'un télégramme SNMT_SN_reset_guarding_SCM.

Tableau 43 – Champs du télégramme SNMT_SN_reset_guarding_SCM

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	1
ID	SNMT_SN_reset_guarding_SCM	$101111xx_b$
LE	Longueur de données utiles	0
CT	Non utilisé	0
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	1

7.4.3.4 Services étendus de SNMT

Les services étendus SNMT sont spécifiés par les champs ID et DB0 du PDU FSPC 13/1 (voir Tableau 44 et Tableau 45).

Tableau 44 – Types de télégrammes de demande SNMT

Numéro de bit de DB0								Type de télégramme SNMT
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	SNMT_SN_set_to_PRE_OP
0	0	0	0	0	0	1	0	SNMT_SN_set_to_OP
0	0	0	0	0	1	0	0	SNMT_SCM_set_to_STOP
0	0	0	0	0	1	1	0	SNMT_SCM_set_to_OP
0	0	0	0	1	0	0	0	SNMT_SCM_guard_SN
0	0	0	0	1	0	1	0	SNMT_assign_additional_SADR
0	0	0	0	1	1	0	0	SNMT_SN_ACK
0	0	0	0	1	1	1	0	SNMT_assign_UDID_SCM
0	0	0	1	0	0	0	0	SNMT_assign_Init_CT

Tableau 45 – Types de télégrammes de réponse SNMT

Numéro de bit de DB0								Type de télégramme SNMT
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	SNMT_SN_status_PRE_OP
0	0	0	0	0	0	1	1	SNMT_SN_status_OP
0	0	0	0	0	1	0	1	SNMT_assigned_additional_SADR
0	0	0	0	0	1	1	1	SNMT_SN_FAIL
0	0	0	0	1	0	0	1	SNMT_SN_busy
0	0	0	0	1	1	1	1	SNMT_assigned_UDID_SCM
0	0	0	1	0	0	0	1	SNMT_assigned_Init_CT

7.4.3.5 SN mis à l'état préopérationnel

Le service SN mis à l'état Préopérationnel doit être utilisé pour faire passer un esclave SNMT de l'état de communication Opérationnel à Préopérationnel.

La Figure 42 spécifie le protocole SN mis à l'état préopérationnel.

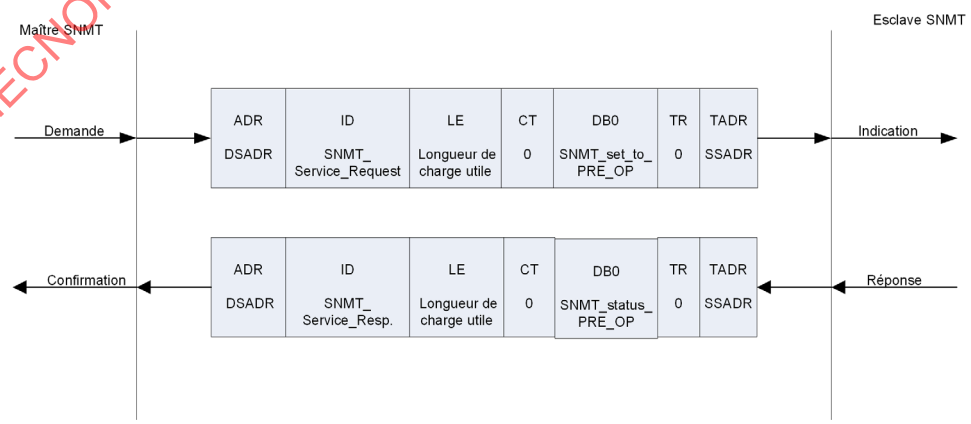


Figure 42 – Protocole SN mis à l'état préopérationnel

Le Tableau 46 spécifie les champs d'un télégramme SNMT_SN_set_to_PRE_OP.

Tableau 46 – Champs du télégramme SNMT_SN_set_to_PRE_OP

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SN_set_to_PRE_OP	00000000b
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 47 spécifie les champs d'un télégramme SNMT_SN_status_PRE_OP.

Tableau 47 – Champs du télégramme SNMT_SN_status_PRE_OP

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SN_status_PRE_OP	00000001b
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.4.3.6 SN mis à l'état Opérationnel

Le service SN mis à l'état Opérationnel doit être utilisé pour faire passer un esclave SNMT de l'état de communication Préopérationnel à Opérationnel. La réponse doit être soit l'information que le SN est passé à l'état Opérationnel, soit une information d'erreur que le SCM utilise pour tenir son application informée.

La Figure 43 spécifie le protocole SN mis à l'état Opérationnel.

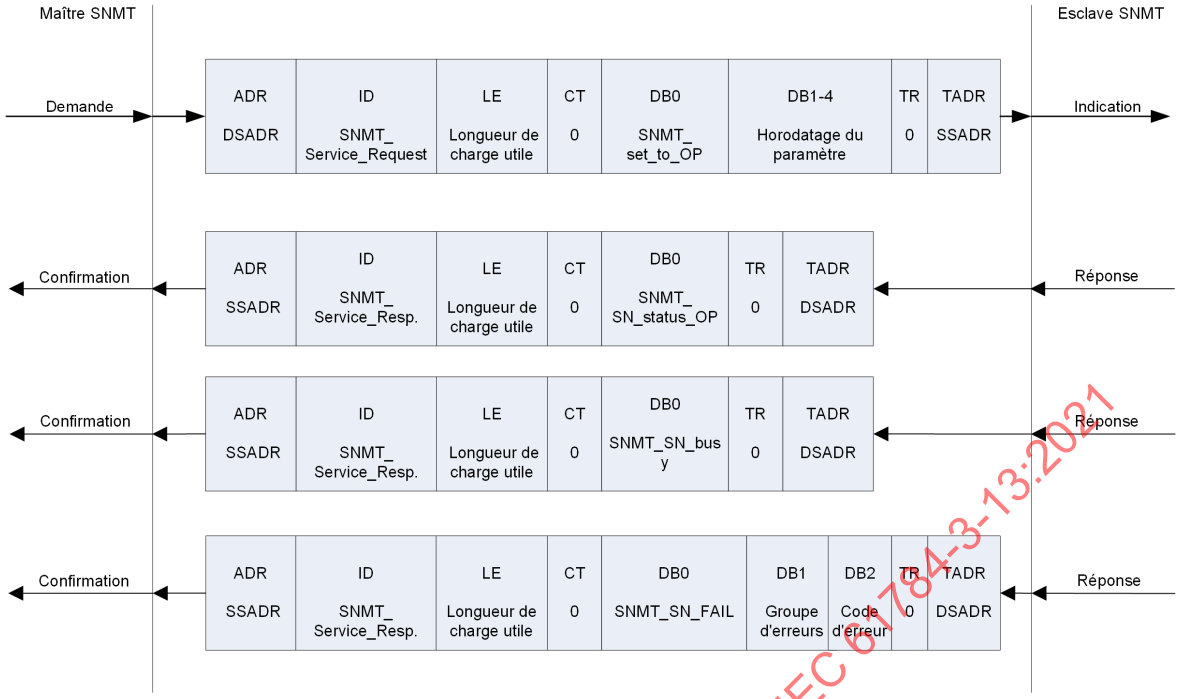


Figure 43 – Protocole SN mis à l'état opérationnel

Le Tableau 48 spécifie les champs d'un télégramme SNMT_SN_set_to_OP.

Tableau 48 – Champs du télégramme SNMT_SN_set_to_OP

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	5
CT	Non utilisé	0
DB0	SNMT_SN_set_to_OP	00000010b
DB1 – DB4	Horodatage du paramètre	Voir 7.5.4.9
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 49 spécifie les champs d'un télégramme SNMT_SN_status_OP.

Tableau 49 – Champs du télégramme SNMT_SN_status_OP

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SN_status_OP	00000011b
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

Le Tableau 50 spécifie les champs d'un télégramme SNMT_SN_busy.

Tableau 50 – Champs du télégramme SNMT_SN_busy

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SN_busy [a]	00001001b
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR
[a] L'application doit répondre à un télégramme SNMT_SN_busy (par exemple pendant le calcul du CRC ou lors de la sauvegarde des paramètres).		

Le Tableau 51 spécifie les champs d'un télégramme SNMT_SN_FAIL.

Tableau 51 – Champs du télégramme SNMT_SN_FAIL

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	3
CT	Non utilisé	0
DB0	SNMT_SN_FAIL	00000111b
DB1	Groupe d'erreurs (voir Tableau 52)	0 – 255
DB2	Code d'erreur (voir Tableau 53)	0 – 255
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

Le Tableau 52 spécifie les valeurs du groupe d'erreurs SNMT_SN_FAIL.

Tableau 52 – Valeurs du groupe d'erreurs SNMT_SN_FAIL

Valeur	Groupe d'erreurs
0	Appareil
1	Application
2	Paramètre
3	Spécifique au fournisseur
4	Pile FSCP 13/1
5	Paramètre supplémentaire nécessaire
6 – 255	Réservé pour une utilisation future

Le Tableau 53 spécifie les valeurs du code d'erreur SNMT_SN_FAIL.

Tableau 53 – Valeurs du code d'erreur SNMT_SN_FAIL

Valeur	Code d'erreur
0	Par défaut
1 – 255	Spécifique au fournisseur

Un SN peut mettre en œuvre des jeux de paramètres supplémentaires. Si un SN renvoie une commande SN_FAIL avec un code ErrorGroup de 5, le code d'erreur définit le jeu de paramètres supplémentaires demandé par le nœud.

En cas de transmission erronée du jeu de paramètres, le nœud doit répondre par un code d'erreur correspondant qui indique quel paramètre n'a pas été transmis correctement.

Le Tableau 54 spécifie les valeurs du code d'erreur SNMT_SN_FAIL lorsque le code du groupe d'erreurs est égal à 5 (voir Tableau 52).

Tableau 54 – Valeurs du code d'erreur SNMT_SN_FAIL en cas de code de groupe d'erreurs égal à 5

Valeur	Code d'erreur
00h – 0Fh	Jeux de paramètres 1-16
10h – EFh	Réservé
F0h – FFh	Défaillance lors de la réception du jeu de paramètres 1-16

7.4.3.7 Acquitter SN de SNMT

Le service Acquitter SN doit être utilisé par le SCM pour acquitter une erreur indiquée (voir 7.4.3.6) au niveau du SN.

La Figure 44 spécifie le protocole Acquitter SN.

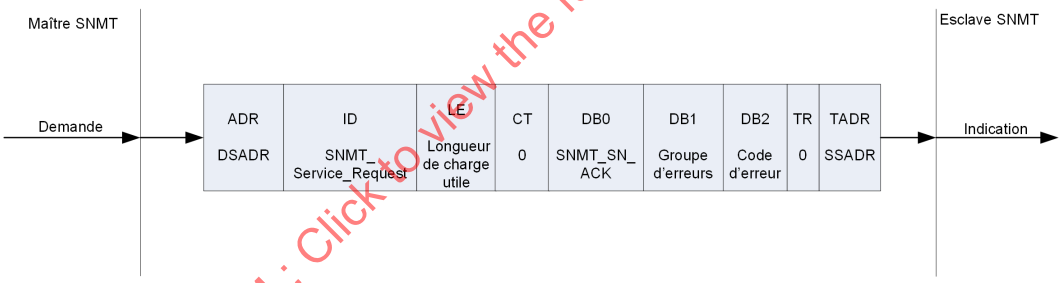


Figure 44 – Protocole Acquitter SN

Le Tableau 55 spécifie les champs d'un télégramme SNMT_SN_ACK.

Tableau 55 – Champs du télégramme SNMT_SN_ACK

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	3
CT	Non utilisé	0
DB0	SNMT_SN_ACK	00001100b
DB1	Groupe d'erreurs (voir Tableau 52)	0 – 255
DB2	Code d'erreur (voir Tableau 53)	0 – 255
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

7.4.3.8 SCM mis à l'arrêt

Le service SCM mis à l'arrêt doit être utilisé pour mettre hors tension un SCM (de l'état de communication Opérationnel à Arrêté) pour configurer le système au moyen d'un outil de configuration externe.

La Figure 45 spécifie le protocole SCM mis à l'état arrêté.

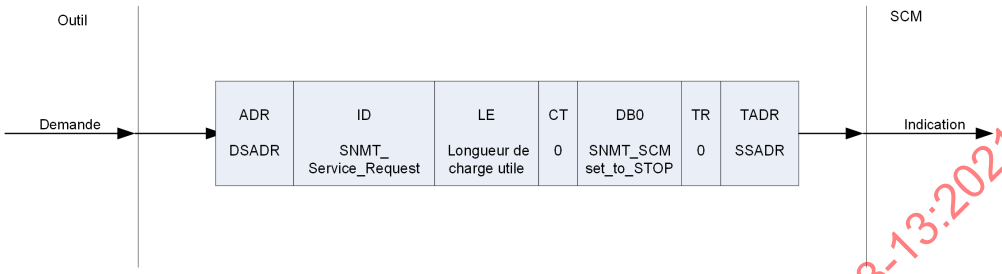


Figure 45 – Protocole SN mis à l'arrêt

Le Tableau 56 spécifie les champs d'un télégramme SNMT_SCM_set_to_STOP.

Tableau 56 – Champs du télégramme SNMT_SCM_set_to_STOP

Champ	Information	Contenu/Valeur
ADR	Adresse du SCM à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SCM_set_to_STOP	00000100b
TR	Non utilisé	0
TADR	Adresse de l'OUTIL	SSADR

7.4.3.9 SCM mis à l'état opérationnel

Le service SCM mis à l'état Opérationnel doit être utilisé pour faire passer le SCM de l'état de communication Arrêté à Opérationnel. Ce service peut être utilisé après l'achèvement de la configuration au moyen d'un outil externe.

La Figure 46 spécifie le protocole SCM mis à l'état opérationnel.

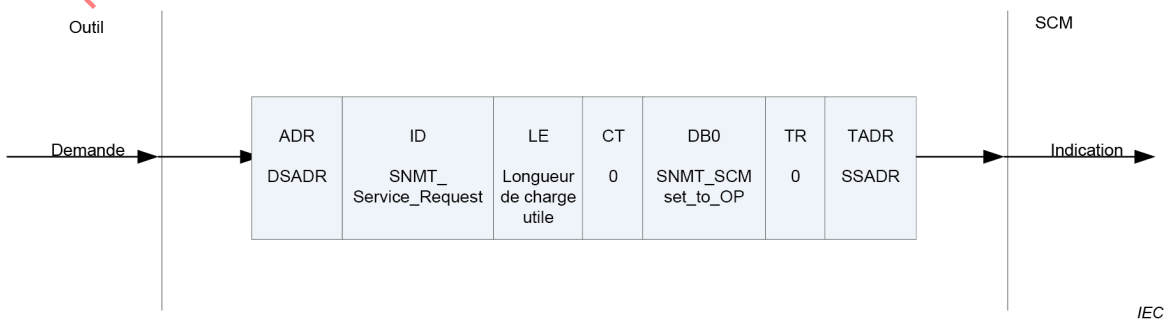


Figure 46 – Protocole SCM mis à l'état opérationnel

Le Tableau 57 spécifie les champs d'un télégramme SNMT_SCM_set_to_OP.

Tableau 57 – Champs du télégramme SNMT_SCM_set_to_OP

Champ	Information	Contenu/Valeur
ADR	Adresse du SCM à atteindre	DSADR
ID	SNMT_Service_Request	101100xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SCM_set_to_OP	00000110b
TR	Non utilisé	0
TADR	Adresse de l'OUTIL	SSADR

7.4.3.10 Sauvegarde du nœud

Le service de sauvegarde du nœud doit être utilisé pour sauvegarder un SN à l'état de communication Opérationnel. Le SN doit répondre en indiquant son état. Si un SN ne reçoit pas un télégramme SNMT_SCM_guard_SN pendant sa durée de vie (voir 7.5.4.6), il doit repasser à l'état Préopérationnel. La signalisation d'une perte de demande ou de réponse de l'application doit être spécifique au fournisseur.

La Figure 47 spécifie le protocole de sauvegarde du nœud.

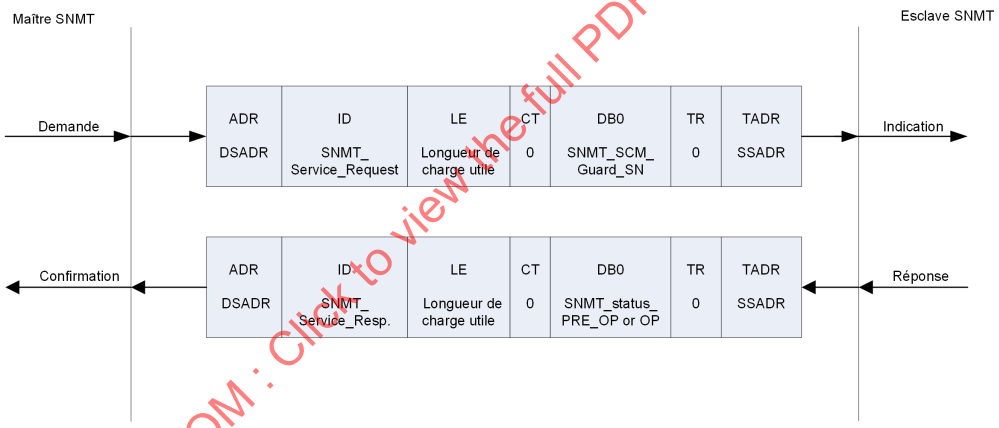


Figure 47 – Protocole de sauvegarde du nœud

Le Tableau 58 spécifie les champs d'un télégramme SNMT_SCM_guard_SN.

Tableau 58 – Champs du télégramme SNMT_SCM_guard_SN

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SCM_guard_SN	00001000b
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 59 spécifie les champs des télégrammes SNMT_SN_status_PRE_OP/SNMT_SN_status_OP.

Tableau 59 – Champs des télégrammes SNMT_SN_status_OP/SNMT_SN_status_OP

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	1
CT	Non utilisé	0
DB0	SNMT_SN_status_PRE_OP ou SNMT_SN_status_OP	00000001b ou 00000011b
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.4.3.11 Attribution de SADR supplémentaire

Le service Attribution de SADR supplémentaire doit être utilisé pour attribuer une SADR supplémentaire à un esclave SNMT. Ce service doit uniquement être utilisé si un SN ne prend pas en charge plusieurs TxSPDO (voir 7.5.4.18).

Si l'esclave SNMT ne prend pas en charge plusieurs TxSPDO, il doit répondre par une valeur SADR égale à 0 (DB1, DB2 dans le télégramme de réponse).

La Figure 48 spécifie le protocole d'attribution d'une SADR supplémentaire.

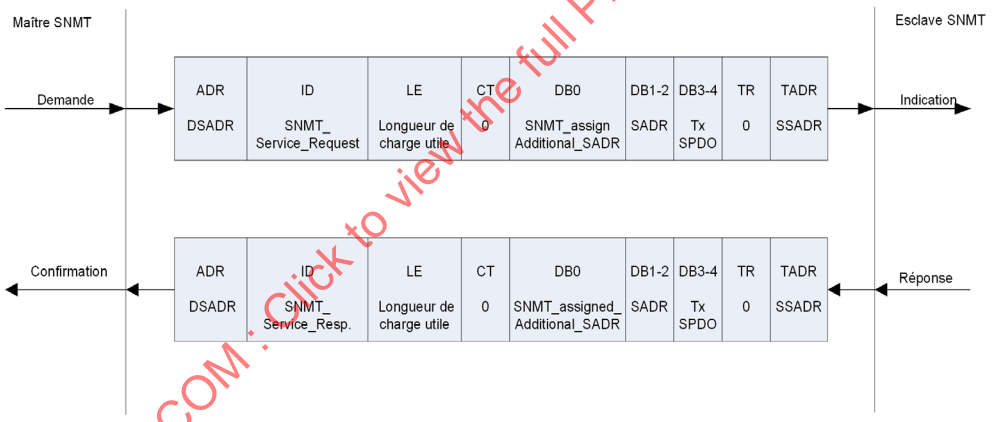


Figure 48 – Protocole Attribution d'une SADR supplémentaire

Le Tableau 60 spécifie les champs d'un télégramme SNMT_assign_additional_SADR.

Tableau 60 – Champs du télégramme SNMT_assign_additional_SADR

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	5
CT	Non utilisé	0
DB0	SNMT_assign_additional_SADR	00001010b
DB1, DB2	SADR supplémentaire à attribuer	SADR
DB3, DB4	TxSPDO auquel est attribuée la SADR supplémentaire	2 à 1 023
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 61 spécifie les champs d'un télégramme SNMT_assigned_additional_SADR.

Tableau 61 – Champs du télégramme SNMT_assigned_additional_SADR

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	5
CT	Non utilisé	0
DB0	SNMT_assigned_additional_SADR	00000101b
DB1, DB2	SADR qui a été attribuée	SADR
DB3, DB4	TxSPDO auquel a été attribuée la SADR	2 à 1 023
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.4.3.12 Attribution d'UDID de SCM

Le service Attribution d'UDID de SCM doit être utilisé pour attribuer l'UDID du SCM à l'esclave SNMT.

NOTE L'UDID du SCM est codé dans tous les SPDO et SSDO. L'esclave SNMT a donc besoin de l'UDID du SCM pour décoder ces télégrammes.

Une erreur d'exécution de ce service doit être indiquée par une valeur d'UDID 00-00-00-00-00-00 dans la réponse.

La Figure 49 spécifie le protocole d'attribution d'UDID de SCM.

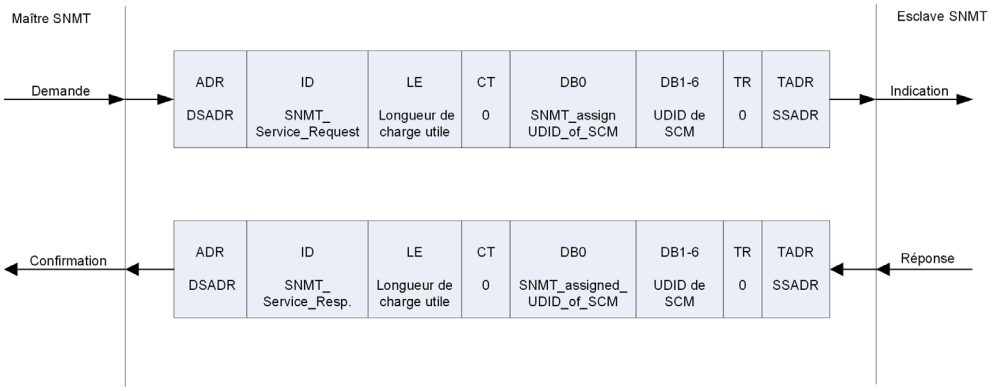


Figure 49 – Protocole Attribution d'UDID de SCM

Le Tableau 62 spécifie les champs d'un télégramme SNMT_assign_UDID_of_SCM.

Tableau 62 – Champs du télégramme SNMT_assign_UDID_of_SCM

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	7
CT	Non utilisé	0
DB0	SNMT_assign_UDID_of_SCM	00001110b
DB1 – DB6	UDID du SCM	UDID
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 63 spécifie les champs d'un télégramme SNMT_assigned_UDID_of_SCM.

Tableau 63 – Champs du télégramme SNMT_assigned_UDID_of_SCM

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	7
CT	Non utilisé	0
DB0	SNMT_assigned_UDID_of_SCM	00001111b
DB1 – DB6	UDID du SCM	UDID
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.4.3.13 Initialisation de CT sur SN

En cas d'utilisation du compteur SPDO 40 bits (voir 7.1.1.4), le service d'attribution de CT d'initialisation doit être utilisé pour amorcer l'horloge CT d'un SN spécifique avec une valeur aléatoire issue du SCM. Cette valeur initiale doit être la nouvelle valeur de départ du compteur CT sur le SN.

Sinon, les SN fournissent déjà une valeur CT qui leur est propre ou le service d'attribution de valeur CT d'initialisation peut également être utilisé.

La Figure 50 spécifie le protocole d'attribution de la valeur CT d'initialisation.

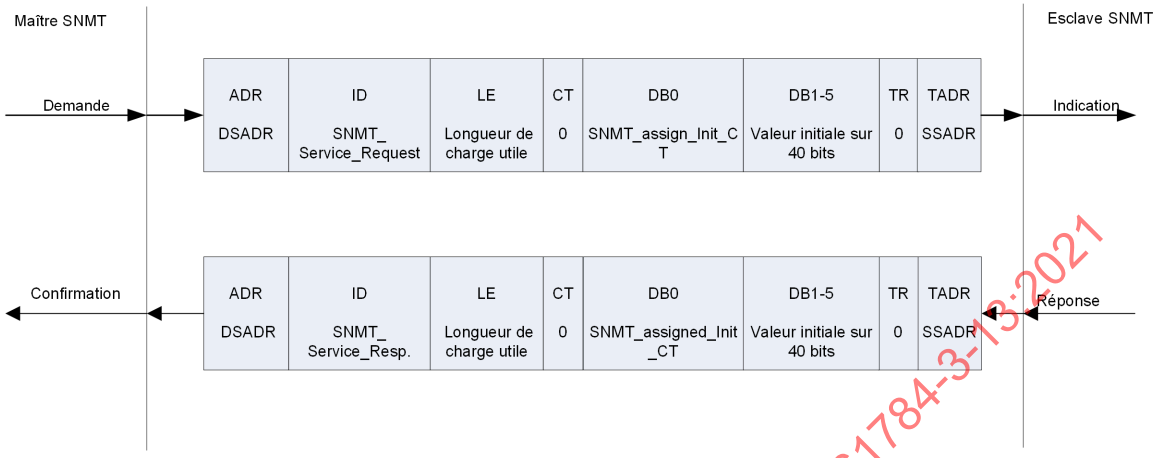


Figure 50 – Protocole Attribution de valeur CT d'initialisation

Le Tableau 64 spécifie les champs d'un télégramme SNMT_assign_Init_CT.

Tableau 64 – Champs du télégramme SNMT_assign_Init_CT

Champ	Information	Contenu/Valeur
ADR	Adresse de l'esclave SNMT à atteindre	DSADR
ID	Demande de service SNMT	101100xxb
LE	Longueur de données utiles	7
CT	Non utilisé	0
DB0	SNMT_assign_Init CT	00010000b
DB1 – DB5	Valeur initiale sur 40 bits	UDID
TR	Non utilisé	0
TADR	Adresse du maître SNMT	SSADR

Le Tableau 65 spécifie les champs d'un télégramme SNMT_assigned_Init_CT.

Tableau 65 – Champs du télégramme SNMT_assigned_Init_CT

Champ	Information	Contenu/Valeur
ADR	Adresse du maître SNMT	DSADR
ID	Réponse de service SNMT	101101xxb
LE	Longueur de données utiles	7
CT	Non utilisé	0
DB0	SNMT_assigned_Init CT	00010001b
DB1 – DB5	Valeur initiale sur 40 bits	UDID
TR	Non utilisé	0
TADR	Adresse de l'esclave SNMT	SSADR

7.5 Dictionnaire d'objets de sécurité (SOD)

7.5.1 Généralités

Les sous-classes suivantes spécifient la structure et les entrées d'un dictionnaire d'objets de sécurité, qui doivent être communes à l'ensemble des appareils.

7.5.2 Définition d'une entrée de dictionnaire d'objets

7.5.2.1 Généralités

Une entrée de dictionnaire d'objets doit être définie par les éléments suivants:

– index;
– objet;
– type d'objet;
– nom;
– type de données;
– catégorie;
– accès;
– plage de valeurs;
– valeur par défaut;
– mapping SPDO.

7.5.2.2 Index

L'index doit décrire la position des objets dans le dictionnaire d'objets. Il référence le champ de données exigé à la manière d'une adresse. Un index doit être déclaré en valeur hexadécimale.

7.5.2.3 Objet

Un objet peut être subdivisé en sous-index. Le sous-index doit être utilisé pour référencer des champs de données dans un objet composite tel qu'une matrice ou un enregistrement. Pour la définition des sous-index, voir 7.5.2.14.

7.5.2.4 Type d'objet

Le type d'objet doit indiquer quel type d'objet se trouve à cet index particulier dans le dictionnaire d'objets. Les types d'objets doivent être tels que spécifiés dans le Tableau 66.

Tableau 66 – Définition des types d'objets

Type d'objet	Commentaires	Code
NULL	Une entrée de dictionnaire sans champ de données	0
DOMAIN	Grande quantité variable de données, par exemple le code d'un programme exécutable	2
DEFTYPE	Définition de type de données de base (par exemple BOOLEAN, UNSIGNED16, REAL32)	5
DEFSTRUCT	Définit un type d'enregistrement	6
VAR	Valeur unique (UNSIGNED8, BOOLEAN, REAL32, INTEGER16, VISIBLE STRING – CHAÎNE VISIBLE, par exemple)	7
ARRAY	Objet à champs de données multiples, où chaque champ de données est une simple variable du MÊME type de données de base, par exemple une matrice de UNSIGNED16, etc. Le sous-index 0 est du type UNSIGNED8 et ne fait donc pas partie des données de la MATRICE.	8
RECORD	Objet à champ de données multiples dont les champs de données peuvent être des combinaisons de variables simples. Le sous-index 0 est du type UNSIGNED8 et ne fait donc pas partie des données d'ENREGISTREMENT	9

7.5.2.5 Nom

Le nom doit fournir une description textuelle simple de la fonction de cet objet particulier. Le nom doit être conforme à l'IEC 61131-3. Un nom doit être constitué des éléments suivants:

- le préfixe du domaine qui indique l'association de l'objet à un domaine fonctionnel et qui utilise jusqu'à 4 caractères en majuscules suivis d'un trait bas;
- un nom (verbal) constitué de mots, chacun doit commencer par une majuscule suivie par des minuscules ou des chiffres, sans traits bas ni espaces;
- le suffixe du type de données qui indique le type de données de l'objet (trait bas suivi de 5 caractères en majuscules ou de chiffres).

La longueur totale du nom ne doit pas dépasser 32 caractères.

7.5.2.6 Type de données

Le type de données doit fournir des informations sur le type de données de l'objet. Il comprend les types de données de base prédéfinis suivants: BOOLÉEN (BOOLEAN), nombre à virgule flottante, entier UNSIGNED (NON SIGNÉ), entier signé, chaîne visible d'octets et DOMAINE (DOMAIN). Il comprend également les types de données composés prédéfinis et peut aussi inclure des types spécifiques au fabricant ou à l'appareil.

Les éléments suivants ne doivent pas être définis:

- les enregistrements d'enregistrements;
- les matrices d'enregistrements; ou
- les enregistrements qui comportent des matrices comme champs de cet enregistrement.

Lorsqu'un objet est une matrice ou un enregistrement, le sous-index doit être utilisé pour référencer un champ de données de type de données de base au sein de l'objet.

Le type de données doit être un type de données de base pour les objets de type ARRAY (MATRICE) ou un type de données composé pour les objets de type RECORD (ENREGISTREMENT).

7.5.2.7 Catégorie

La catégorie définit si l'objet est obligatoire (M) ou facultatif (O). Un objet obligatoire doit être mis en œuvre dans un appareil. Un objet facultatif peut être mis en œuvre dans un appareil. La prise en charge de certains objets ou fonctionnalités peut cependant nécessiter la mise en œuvre d'objets liés. Dans ce cas, les relations sont décrites dans la spécification particulière de l'objet.

La catégorie peut être conditionnelle (Cond) si la catégorie M/O d'un objet dépend de la mise en œuvre d'un autre objet.

La catégorie doit faire référence à un objet de type de données composé complet et non à un sous-index particulier. Un objet obligatoire peut être constitué de sous-index obligatoires et facultatifs. Ceci signifie que l'objet doit être pris en charge, mais que certains de ses sous-index sont facultatifs. Il est également admis de définir des objets facultatifs avec des sous-index obligatoires. Cela signifie que ces sous-index doivent être pris en charge si l'objet est mis en œuvre.

7.5.2.8 Accès

L'accès définit les droits d'accès à un objet particulier. Le point de vue adopté est de l'extérieur vers l'intérieur de l'appareil.

Les droits d'accès sont spécifiés dans le Tableau 67.

Tableau 67 – Attributs d'accès pour des objets de données

Droit d'accès	Description
rw	accès en lecture et en écriture
wo	accès en écriture uniquement
ro	accès en lecture seule
Const	accès en lecture seule, valeur constante

Les objets facultatifs doivent être répertoriés dans le dictionnaire d'objets avec l'attribut rw; ils peuvent être implémentés en lecture seule. Les exceptions sont définies dans la spécification particulière de l'objet.

7.5.2.9 Plage de valeurs

Cette entrée doit indiquer la plage de valeurs de l'objet correspondant. Elle peut comprendre une ou plusieurs valeurs distinctes ou des plages de valeurs. Si l'élément indiqué est un identifiant de type de données, la plage de valeurs complète du type de données mentionné doit être admise.

7.5.2.10 Valeur par défaut

Cette entrée doit indiquer la valeur d'initialisation qui doit être attribuée par la mise en œuvre du profil. Pour les objets qui peuvent être mappés, cette valeur doit également représenter la valeur de l'état de sécurité.

7.5.2.11 Mapping SPDO

Cette entrée indique si une entrée peut être mappée avec un message SPDO.

Les valeurs de mapping SPDO doivent être telles que spécifiées dans le Tableau 68.

Tableau 68 – Attributs de mapping SPDO pour des objets de données

Valeur	Description
Opt	L'objet peut être mappé avec un SPDO
Non	L'objet ne doit pas être mappé avec un SPDO

7.5.2.12 Exemple de définition d'objet de type de données de base

Une définition complète d'objet de type de données de base (Type d'objet = VAR ou DOMAIN) est fournie dans le Tableau 69.

Tableau 69 – Exemple de définition d'objet de type de données de base

Elément	Valeur
Index	1234h
Nom	SSDO_VarDummyParameter_S16
Type d'objet	VAR
Type de données	INTEGER16
Plage de valeurs	-15 000 à 10 000
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Opt

7.5.2.13 Exemple de définition d'objet de type de données composé

Les définitions d'objets de type de données composé (Type d'objet = ARRAY ou RECORD) sont données sous forme réduite, car Accès, Plage de valeurs, Valeur par défaut et Mapping SPDO doivent être définis individuellement pour les sous-index.

Le Tableau 70 donne un exemple de définition d'objet de type de données composé.

Tableau 70 – Exemple de définition d'objet de type de données composé

Elément	Valeur
Index	2345h
Nom	SSDO_ArrayDummyParameter_AU16
Type d'objet	ARRAY
Type de données	UNSIGNED16
Catégorie	M

7.5.2.14 Définition de sous-index

7.5.2.14.1 Généralités

Les types d'objets composés (objets ARRAY, RECORD) doivent comporter jusqu'à 256 éléments de données. Chaque élément de données doit être identifié par un sous-index de type UNSIGNED8.

Les "sous-index" doivent être interprétés comme spécifié dans le Tableau 71.

Tableau 71 – Interprétation des sous-index

Sous-index	Nom
00h	NumberOfEntries
01h à FEh	Données spécifiques à l'objet
FEh	ChecksumOfObject
FFh	StructureOfObject

7.5.2.14.2 Sous-index 00h – NumberOfEntries

NumberOfEntries doit décrire le sous-index disponible le plus élevé (hors FFh). Cette entrée doit être codée en UNSIGNED8, quel que soit le type de l'objet. Si l'objet existe, NumberOfEntries doit être obligatoire. Le type de données et la catégorie ne doivent pas être représentés dans les descriptions NumberOfEntries.

Dans le présent document, NumberOfEntries est spécifié sous la forme indiquée dans le Tableau 72. Les éléments compris entre <...> doivent être définis en fonction du contexte spécifique.

Tableau 72 – Spécification du sous-index NumberOfEntries

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries
Plage de valeurs	<valeurs valides, par exemple 1 à 15>
Valeur par défaut	<valeur par défaut, par exemple 15>
Accès	<accès, par exemple rw>
Mapping SPDO	Non

Pour les matrices (ARRAY), NumberOfEntries doit exprimer le nombre d'éléments occupés dans une liste (l'accès doit être rw). Pour les enregistrements (RECORD), NumberOfEntries doit être constant (l'accès doit être ro ou Const).

7.5.2.14.3 Sous-index 01h – FEh – Données spécifiques à l'objet

Les sous-index compris entre 01h et FEh doivent contenir les données utiles de l'objet. L'index accessible le plus élevé doit être donné par NumberOfEntries.

Dans le présent document, les données spécifiques aux objets de type RECORD sont spécifiées sous la forme indiquée dans le Tableau 73. Les éléments compris entre <...> doivent être définis en fonction du contexte spécifique.

Tableau 73 – Spécification du sous-index d'objet de type RECORD

Elément	Valeur
Sous-index	<sous-index en notation hexadécimale, par exemple 01h>
Nom	<nom>
Type de données	<type de données, par exemple UNSIGNED8>
Plage de valeurs	<valeurs valides, par exemple 1 à 255>
Valeur par défaut	<valeur par défaut, par exemple 1>
Catégorie	<obligatoire (M) ou facultatif (O), par exemple M>
Accès	<accès, par exemple rw>
Mapping SPDO	<mapping possible, par exemple Opt>

Le nom doit être constitué d'un texte descriptif suivi d'un suffixe de type de données. Le 7.5.2.5 doit s'appliquer à la seule exception que le préfixe du domaine n'est pas pris en compte.

Si une entrée Nom est définie dans une définition de type de données, le sous-index Nom doit être identique à cette entrée.

La catégorie doit faire référence au sous-index correspondant uniquement. Obligatoire (M) doit indiquer que le sous-index doit être appliqué si l'objet est utilisé. Un sous-index obligatoire ne doit pas forcer l'objet parent à être obligatoire.

Dans le présent document, les données spécifiques aux objets de type ARRAY sont spécifiées sous la forme indiquée dans le Tableau 74. Les éléments compris entre <...> doivent être définis en fonction du contexte spécifique.

Tableau 74 – Spécification du sous-index d'objet de type ARRAY

Elément	Valeur
Sous-index	<sous-index en notation hexadécimale, par exemple 01h>
Nom	<nom>
Plage de valeurs	<type de données, par exemple UNSIGNED16>
Valeur par défaut	<valeur par défaut, par exemple 0>
Catégorie	<obligatoire (M) ou facultatif (O), par exemple M>
Accès	<accès, par exemple rw>
Mapping SPDO	<mapping possible, par exemple Opt>

Le nom doit être un texte descriptif. Le 7.5.2.5 doit s'appliquer à la seule exception que le préfixe du domaine et le suffixe du type de données ne sont pas pris en compte. Si une entrée Nom est définie dans une définition de type de données, le sous-index Nom doit être identique à cette entrée.

La catégorie doit faire référence au sous-index correspondant uniquement. Obligatoire (M) doit indiquer que le sous-index doit être appliqué si l'objet est utilisé. Un sous-index obligatoire ne doit pas forcer l'objet parent à être obligatoire.

NOTE Malgré l'équivalence fonctionnelle de l'ensemble des sous-index d'une matrice, plusieurs entrées de sous-index peuvent être définies pour un objet particulier afin de montrer les différences des options Catégorie, Accès et/ou Mapping SPDO des sous-index.

7.5.2.14.4 Sous-index FFh – StructureOfObject

Le sous-index FFh peut représenter la structure de l'objet en fournissant le type de données et le type d'objet. Quel que soit le type d'objet, il est codé UNSIGNED32.

Le codage de StructureOfObject est spécifié dans le Tableau 75.

Tableau 75 – Codage de StructureOfObject

	MSB [a]			LSB [b]
Numéro de bit	31 – 24	23 – 16	15 – 8	7 – 0
Valeur	00h	Type de données (voir Tableau 76) (octet de poids fort)	(octet de poids faible)	Type d'objet (voir Tableau 66)

[a] Octet de poids fort.

[b] Octet de poids faible.

NOTE La prise en charge du sous-index FFh est facultative. Si elle est utilisée dans l'ensemble du dictionnaire d'objets et que la structure des types de données composés est également fournie, il est possible de télécharger (amont) l'ensemble de la structure du dictionnaire d'objets.

Si StructureOfEntry n'est pas pris en charge, le sous-index FFh doit être réservé.

StructureOfObject n'est pas indiqué dans les définitions d'objets de cette spécification.

7.5.3 Spécification d'une entrée de type de données

7.5.3.1 Généralités

Les types de données de base doivent être placés dans le dictionnaire d'objets à des fins de définition uniquement. Cependant, les index de la plage 0001h à 0007h peuvent être mappés afin de définir l'espace approprié du RxSPDO (mapping factice) comme n'étant pas utilisé par l'appareil (ignorer/don't care). Les index 0009h à 000Bh et 000Fh ne doivent pas être mappés avec un SPDO.

La plage d'index 0001h à 04FFh doit être utilisée pour les entrées du dictionnaire d'objets qui spécifient le type de données.

Le Tableau 76 spécifie les types de données utilisés dans le SOD.

Tableau 76 – Types de données du dictionnaire d'objets

Index	Objet	Nom
0001h	DEFTYPE	BOOLEAN
0002h	DEFTYPE	INTEGER8
0003h	DEFTYPE	INTEGER16
0004h	DEFTYPE	INTEGER32
0005h	DEFTYPE	UNSIGNED8
0006h	DEFTYPE	UNSIGNED16
0007h	DEFTYPE	UNSIGNED32
0008h	DEFTYPE	REAL32
0009h	DEFTYPE	VISIBLE_STRING
000Ah	DEFTYPE	OCTET_STRING
000Bh	DEFTYPE	UNICODE_STRING
000Ch	Réservé	
000Dh	Réservé	
000Eh	Réservé	
000Fh	DEFTYPE	DOMAIN
0010h	DEFTYPE	INTEGER24
0011h	DEFTYPE	REAL64
0012h	DEFTYPE	INTEGER40
0013h	DEFTYPE	INTEGER48
0014h	DEFTYPE	INTEGER56
0015h	DEFTYPE	INTEGER64
0016h	DEFTYPE	UNSIGNED24
0017h	Réservé	
0018h	DEFTYPE	UNSIGNED40
0019h	DEFTYPE	UNSIGNED48
001Ah	DEFTYPE	UNSIGNED56
001Bh	DEFTYPE	UNSIGNED64
001Ch – 001Fh	Réservé	
0020h	DEFSTRUCT	SSDO_LifeGuardRecord_TYPE
0021h	DEFSTRUCT	SINF_DevVendorInfoRecord_TYPE
0022h	DEFSTRUCT	SCOM_CommonCommParamRecord_TYPE
0023h	DEFSTRUCT	SSDO_CommParamRecord_TYPE
0024h	DEFSTRUCT	SNMT_CommParamRecord_TYPE

Index	Objet	Nom
0025h	DEFSTRUCT	SPDO_RxCommParamRecord_TYPE
0026h	DEFSTRUCT	SPDO_TxCommParamRecord_TYPE
0027h	DEFSTRUCT	SSCM_SADRDVIListRecord_TYPE
0028h	DEFSTRUCT	SSCM_UDIDSADRListRecord_TYPE
0029h	DEFSTRUCT	SSCM_SpecificParametersRecord_TYPE
002Ah – 003Fh	Réservé	

7.5.3.2 Types de données de base

Pour les types de données de base (Type d'objet = DEFTYPE), voir 5.5.2.

Un appareil peut éventuellement indiquer la longueur des types de données de base codée comme UNSIGNED32 pour l'accès en lecture de l'index qui fait référence au type de données.

EXEMPLE L'index 0007h (UNSIGNED32) contient la valeur 20h=32d, car le type de données UNSIGNED32 est codé au moyen d'une séquence binaire de 32 bits. Si la longueur est variable (par exemple 000Fh = Domain), l'entrée contient 0h.

7.5.3.3 Types de données composés

Pour les types de données composés pris en charge, un appareil peut, en option, fournir la structure de ce type de données pour l'accès en lecture à l'index de type de données correspondant. Le sous-index 0 donne ensuite le nombre d'entrées à cet index, sans compter les sous-index 0 et 255; les sous-index suivants contiennent le type de données selon le Tableau 76, codé en UNSIGNED16.

Le Tableau 77 et le Tableau 78 donnent un exemple de définition du type de données composé qui décrit la structure de l'objet d'identité à l'index 0021h.

Tableau 77 – Exemple de description du type de données composé 0021h

Elément	Valeur
Index	0021h
Nom	SINF_DevVendorInfoRecord_TYPE
Type d'objet	DEFSTRUCT

Tableau 78 – Exemple de description des sous-index composés 0021h

Sous-index	Nom du composant	Valeur	Type de données
00h	NumberOfEntries	06h	UNSIGNED16
01h	VendorId_U32	07h	UNSIGNED32
02h	ProductCode_U32	07h	UNSIGNED32
03h	RevisionNo_U32	07h	UNSIGNED32
04h	SerialNo_U32	07h	UNSIGNED32
05h	FirmWareChecksum_U32	07h	UNSIGNED32
06h	ParameterChecksum_U16	06h	UNSIGNED16
07h	ParameterTimestamp_U32	07h	UNSIGNED32

Une distinction doit être établie entre les types de données spécifiques au fabricant qui sont normalisés (simples) et ceux qui sont composés en tentant de lire le sous-index 1h: Pour un type de données composé, l'appareil doit renvoyer une valeur et le sous-index 0h contient le numéro du sous-index suivant. Pour un type de données normalisé, l'appareil doit abandonner l'accès au SOD, car aucun sous-index 1h n'est disponible.

7.5.4 Description des objets

7.5.4.1 Généralités

Cette sous-classe et les sous-classes suivantes décrivent les entrées du dictionnaire d'objets normalisées pour un SN et un SCM. La mise en œuvre de ces objets peut être différente de la description fournie dans la spécification.

EXEMPLE La SADR principale peut être un paramètre en lecture seule si elle doit être établie au moyen d'un commutateur DIP.

Les objets normalisés FSCP 13/1 sont spécifiés dans les tableaux ci-après (du Tableau 79 au Tableau 89).

Tableau 79 – Objets normalisés

Index	Type d'objet	Nom	Type de données	Acc	M/O
1001h	VAR	Registre d'erreurs	UNSIGNED8	ro	M
1002h	VAR	Registre d'états du fabricant	UNSIGNED32	ro	O
1003h	ARRAY	Champ d'erreurs prédéfini	UNSIGNED32	rw	O
:::	:::	:::	:::	:::	:::
100Ch	RECORD	Sauvegarde	SSDO_LifeGuardRecord_TYPE (20h)	-	M
100Dh	VAR	Temps de rafraîchissement pour la réinitialisation de sauvegarde	UNSIGNED32	rw	M
100Eh	VAR	Nombre de nouvelles tentatives pour la réinitialisation de sauvegarde	UNSIGNED8	rw	O
1018h	RECORD	Informations de fournisseur d'appareil	SINF_DevVendorInfoRecord_TYPE (21h)	-	M
1019h	VAR	ID unique d'appareil	OCTET_STRING	ro	M
101Ah	DOMAIN	Téléchargement aval de paramètres	DOMAIN	rw	O
101Bh	RECORD	Paramètres spécifiques au SCM	SSCM_SpecificParametersRecord_TYPE (29h)	-	O

Tableau 80 – Objets de communication communs

Index	Type d'objet	Nom	Type de données	Acc	M/O
1200h	RECORD	Paramètres de communication communs	SCOM_CommonCommParamRecord_TYPE (22h)	-	M
1201h	RECORD	Paramètres de communication SSDO	SSDO_CommParamRecord_TYPE (23h)	-	C[a]
1202h	RECORD	Paramètres de communication SNMT	SNMT_CommParamRecord_TYPE (24h)	-	C

[a] Les objets doivent être uniquement disponibles si le nœud peut être un client SSDO.

Tableau 81 – Objets de communication des SPDO de réception

Index	Type d'objet	Nom	Type de données	Acc	M/O
1400h	RECORD	Paramètres de communication du 1er RxSPDO	SPDO_RxCommParamRecord_TYPE (25h)	rw	M[a]
1401h	RECORD	Paramètres de communication du 2e RxSPDO	SPDO_RxCommParamRecord_TYPE (25h)	rw	M[a]
:::	:::	:::	:::	:::	:::
17FEh	RECORD	Paramètres de communication du 1 023e RxSPDO	SPDO_RxCommParamRecord_TYPE (25h)	rw	M[a]
[a] Si un appareil prend en charge les SPDO, les entrées de paramètres de communication SPDO et les entrées de mapping SPDO correspondantes dans le dictionnaire d'objets sont obligatoires.					

Tableau 82 – Objets de mapping des SPDO de réception

Index	Type d'objet	Nom	Type de données	Acc	M/O
1800h	ARRAY	Paramètres de mapping du 1er RxSPDO	UNSIGNED32	rw	M[a]
1801h	ARRAY	Paramètres de mapping du 2e RxSPDO	UNSIGNED32	rw	M[a]
:::	:::	:::	:::	:::	:::
1BFEh	ARRAY	Paramètres de mapping du 1 023e RxSPDO	UNSIGNED32	rw	M[a]
[a] Si un appareil prend en charge les SPDO, les entrées de paramètres de communication SPDO et les entrées de mapping SPDO correspondantes dans le dictionnaire d'objets sont obligatoires.					

Tableau 83 – Objets de communication des SPDO d'émission

Index	Type d'objet	Nom	Type de données	Acc	M/O
1C00h	RECORD	Paramètres de communication du 1er TxSPDO	SPDO_TxCommParamRecord_TYPE (26h)	rw	M[a]
1C01h	RECORD	Paramètres de communication du 2e TxSPDO	SPDO_TxCommParamRecord_TYPE (26h)	rw	M[a]
:::	:::	:::	:::	:::	:::
1FFEh	RECORD	Paramètres de communication du 1 023e TxSPDO	SPDO_TxCommParamRecord_TYPE (26h)	rw	M[a]
[a] Si un appareil prend en charge les SPDO, les entrées de paramètres de communication SPDO et les entrées de mapping SPDO correspondantes dans le dictionnaire d'objets sont obligatoires.					

Tableau 84 – Paramètre utilisateur (inscriptible à tout moment)

Index	Type d'objet	Nom	Type de données	Acc	M/O
2800h	ARRAY	1er paramètre utilisateur	UNSIGNED32	rw	O
:::	:::	:::	:::	:::	O
2FFFh	ARRAY	2 048e paramètre utilisateur	UNSIGNED32	rw	O

Tableau 85 – Objets de mapping des SPDO d'émission

Index	Type d'objet	Nom	Type de données	Acc	M/O
C000h	ARRAY	Paramètres de mapping du 1er TxSPDO	UNSIGNED32	rw	M[a]
C001h	ARRAY	Paramètres de mapping du 2e TxSPDO	UNSIGNED32	rw	M[a]
:::	:::	:::	:::	:::	:::
C3FEh	ARRAY	Paramètres de mapping du 1 023e TxSPDO	UNSIGNED32	rw	M[a]
[a] Si un appareil prend en charge les SPDO, les entrées de paramètres de communication SPDO et les entrées de mapping SPDO correspondantes dans le dictionnaire d'objets sont obligatoires.					

Tableau 86 – Liste de DVI – SADR

Index	Type d'objet	Nom	Type de données	Acc	M/O
C400h	RECORD	1re liste de DVI – SADR	SSCM_SADRDVIListRecord_TYPE (27h)	rw	C[a]
C401h	RECORD	2e liste de DVI – SADR	SSCM_SADRDVIListRecord_TYPE (27h)	rw	C[a]
:::	:::	:::	:::	:::	:::
C7FEh	RECORD	1 023e liste de DVI – SADR	SSCM_SADRDVIListRecord_TYPE (27h)	rw	C[a]
[a] La mise en œuvre n'est nécessaire que si l'appareil comporte une fonctionnalité SCM.					

Tableau 87 – Liste de SADR supplémentaires

Index	Type d'objet	Nom	Type de données	Acc	M/O
C801h	ARRAY	1re SADR	UNSIGNED16	rw	C[a]
C802h	ARRAY	2e SADR	UNSIGNED16	rw	C[a]
:::	:::	:::	:::	:::	:::
CBFFh	ARRAY	1 023e SADR	UNSIGNED16	rw	C[a]
[a] La mise en œuvre n'est nécessaire que si l'appareil comporte une fonctionnalité SCM.					

Tableau 88 – Liste d'UDID – SADR

Index	Type d'objet	Nom	Type de données	Acc	M/O
CC01h	RECORD	1re liste d'UDID – SADR	SSCM_UDIDSADRListRecord_TYPE (28h)	rw	C[a]
CC02h	RECORD	2e liste d'UDID – SADR	SSCM_UDIDSADRListRecord_TYPE (28h)	rw	C[a]
:::	:::	:::	:::	:::	:::
CFFFh	RECORD	1 023e liste d'UDID – SADR	SSCM_UDIDSADRListRecord_TYPE (28h)	rw	C[a]
[a] La mise en œuvre n'est nécessaire que si l'appareil comporte une fonctionnalité SCM.					

Tableau 89 – Liste de paramètres supplémentaires

Index	Type d'objet	Nom	Type de données	Acc	M/O
E400h	RECORD	1re liste de paramètres supplémentaires	SSCM_ADDPARAListRecord_TYPE (2Ah)	rw	C[a]
:::	:::	:::	:::	:::	:::
E7FEh	RECORD	1 023e liste de paramètres supplémentaires	SSCM_ADDPARAListRecord_TYPE (2Ah)	rw	C[a]
[a] La mise en œuvre n'est nécessaire que si l'appareil comporte une fonctionnalité SCM.					

7.5.4.2 Objet 1001h: Registre d'erreurs

Cet objet doit être un registre d'erreurs pour l'appareil. L'appareil peut mapper des erreurs internes dans cet octet. Si la valeur de l'objet est zéro, l'appareil doit être dans un état correct et toutes les autres valeurs doivent signaler que l'appareil est dans un état d'erreur.

L'objet 1001h Registre d'erreurs est spécifié dans le Tableau 90.

Tableau 90 – Objet 1001h Registre d'erreurs

Elément	Valeur
Index	1001h
Nom	SERR_GeneralError_U8
Type d'objet	VAR
Type de données	UNSIGNED8
Plage de valeurs	UNSIGNED8
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Oui
Pertinent pour la somme de contrôle SOD	Non

Les valeurs de l'objet 1001h Registre d'erreurs sont spécifiées dans le Tableau 91.

Tableau 91 – Interprétation des valeurs de l'objet 1001h Registre d'erreurs

Bit	M/O	Description
0	M	Erreur générique
1	O	Courant
2	O	Tension
3	O	Température
4	O	Erreur de communication (dépassement, état d'erreur)
5	O	Spécifique au profil de l'appareil
6	O	Réservé (toujours à 0)
7	O	Spécifique au fabricant

7.5.4.3 Objet 1002h: Registre d'états du fabricant

Cet objet doit être un registre d'états destiné au fabricant. Dans le présent document, seuls la taille et l'emplacement de cet objet doivent être définis.

L'objet 1002h Registre d'états du fabricant est spécifié dans le Tableau 92.

Tableau 92 – Objet 1002h Registre d'états du fabricant

Elément	Valeur
Index	1002h
Nom	SERR_ManufacturerStatus_U32
Type d'objet	VAR
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	O
Accès	ro
Mapping SPDO	Oui
Pertinent pour la somme de contrôle SOD	Non

7.5.4.4 Objet 1003h: Champ d'erreurs prédéfini

L'objet d'index 1003h doit stocker les erreurs qui se sont produites sur l'appareil. Chaque nouvelle erreur doit être stockée au sous-index 01h. Les erreurs plus anciennes doivent être déplacées vers le sous-index immédiatement supérieur. De ce fait, il fournit un historique des erreurs. Le sous-index 0 doit fournir le nombre d'erreurs dans la liste. La mise à zéro du sous-index 0 doit être utilisée pour effacer la liste.

L'objet 1003h Champ d'erreurs prédéfini est spécifié dans les tableaux ci-après (du Tableau 93 au Tableau 96).

Tableau 93 – Objet 1003h Champ d'erreurs prédéfini

Elément	Valeur
Index	1003h
Nom	SERR_ErrorHistory_AU32
Type d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	O
Pertinent pour la somme de contrôle SOD	Non

Tableau 94 – Objet 1003h sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	1 – 254
Valeur par défaut	1
Accès	rw
Mapping SPDO	Non
[a] Nombre d'erreurs.	

Tableau 95 – Objet 1003h sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	Error_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Erreur dans la liste historique.	

Tableau 96 – Objet 1003h sous-index 02h à FEh

Elément	Valeur
Sous-index	02h à FEh
Nom	Error_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	O
Accès	ro
Mapping SPDO	Non
[a] Erreur dans la liste historique.	

7.5.4.5 Objet 1004h: Statistiques d'erreur

L'objet 1004h spécifie les compteurs de statistiques d'erreurs du Tableau 97 au Tableau 112.

Tableau 97 – Objet 1003h Statistiques d'erreurs

Elément	Valeur
Index	1004h
Nom	SERR_ErrorStatistics_AU32
Type d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	O
Pertinent pour la somme de contrôle SOD	Non

Tableau 98 – Objet 1004h sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	0Eh
Valeur par défaut	0Eh
Accès	rw
Mapping SPDO	Non
[a] Nombre d'entrées de statistiques d'erreur.	

Tableau 99 – Objet 1004h sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	SFS_Length [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] La longueur de charge utile définie pour le PDU n'est pas la même que celle des données reçues.	

Tableau 100 – Objet 1004h sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	SFS_too_long [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] PDU de sécurité trop long.	

Tableau 101 – Objet 1004h sous-index 03h

Elément	Valeur
Sous-index	03h
Nom	SFS_PDU_ID [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Les ID de PDU ne sont pas les mêmes dans les parties 1 et 2 du PDU.	

Tableau 102 – Objet 1004h sous-index 04h

Elément	Valeur
Sous-index	04h
Nom	SFS_SADR_invalid [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] SADR invalide.	

Tableau 103 – Objet 1004h sous-index 05h

Elément	Valeur
Sous-index	05h
Nom	SFS_SDN_invalid [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] SDN invalide.	

Tableau 104 – Objet 1004h sous-index 06h

Elément	Valeur
Sous-index	06h
Nom	SFS_TADR_invalid [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] TADR invalide.	

Tableau 105 – Objet 1004h sous-index 07h

Elément	Valeur
Sous-index	07h
Nom	SFS_CRC1_invalid [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] CRC de la partie 1 du PDU invalide.	

Tableau 106 – Objet 1004h sous-index 08h

Elément	Valeur
Sous-index	08h
Nom	SFS_CRC2_invalid [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] CRC de la partie 2 du PDU invalide.	

Tableau 107 – Objet 1004h sous-index 09h

Elément	Valeur
Sous-index	09h
Nom	SFS_Data [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] La charge utile n'est pas la même dans les parties 1 et 2 du PDU.	

Tableau 108 – Objet 1004h sous-index 0Ah

Elément	Valeur
Sous-index	0Ah
Nom	CYC_Reject [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Télégrammes cycliques rejetés, car ils ne s'appliquent pas à un SN.	

Tableau 109 – Objet 1004h sous-index 0Bh

Elément	Valeur
Sous-index	0Bh
Nom	CYC_Error [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Erreurs génériques de télégrammes cycliques, télégrammes destinés à un SN.	

Tableau 110 – Objet 1004h sous-index 0Ch

Elément	Valeur
Sous-index	0Ch
Nom	ACYC_Reject [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Télégrammes SNMT/SSDO rejetés, car ils ne s'appliquent pas à un SN.	

Tableau 111 – Objet 1004h sous-index 0Dh

Elément	Valeur
Sous-index	0Dh
Nom	ACYC_Retry [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Nombre de tentatives SNMT/SSDO.	

Tableau 112 – Objet 1004h sous-index 0Eh

Elément	Valeur
Sous-index	0Eh
Nom	COMMON_Counter [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Compteur d'erreurs courantes.	

7.5.4.6 Objet 100Ch: Sauvegarde

Cet objet définit le protocole de sauvegarde. L'intervalle de sauvegarde doit être l'intervalle de temps au cours duquel il convient que le SN reçoive un signal de durée de vie du SCM. Le facteur de durée de vie multiplié par l'intervalle de sauvegarde donne la durée de vie du nœud. L'intervalle de temps pour l'envoi du signal de durée de vie aux SN doit être calculé en divisant le GuardTime_U32 (Intervalle de sauvegarde U32) par le nombre de nœuds. L'intervalle de sauvegarde pour un SN dans le SCM commence après réception de la dernière réponse de sauvegarde du SN.

L'objet 100Ch Sauvegarde est spécifié dans les tableaux ci-après (du Tableau 113 au Tableau 116).

Tableau 113 – Objet 100Ch Sauvegarde

Elément	Valeur
Index	100Ch
Nom	SSDO_LifeGuard_REC
Type d'objet	RECORD
Type de données	SSDO_LifeGuardRecord_TYPE
Catégorie	M
Pertinent pour la somme de contrôle SOD	Oui

Tableau 114 – Objet 100Ch sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	2
Valeur par défaut	2
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 115 – Objet 100Ch sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	GuardTime_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	10 000 000
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Intervalle de sauvegarde pour le protocole de durée de vie. La base de temps de ce paramètre est définie par CommonCommParam_REC. ConsecutiveTimeBase_I8. La plage des valeurs peut être limitée à une valeur inférieure à 232. Cette limite est spécifique au fournisseur et/ou est due à des raisons de mise en œuvre.	

Tableau 116 – Objet 100Ch sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	LifeTimeFactor_U8 [a]
Type de données	UNSIGNED8
Plage de valeurs	1 – 255
Valeur par défaut	2
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Nombre de fois où le GuardTime_U32 peut expirer avant que le nœud ne repasse de l'état opérationnel à l'état préopérationnel.	

7.5.4.7 Objet 100Dh: Intervalle de rafraîchissement de la réinitialisation de sauvegarde

Cet objet contient l'intervalle de rafraîchissement de la réinitialisation de sauvegarde (voir 7.7.6.3). L'intervalle de rafraîchissement doit être le délai entre chaque signal SNMT_SN_reset_guarding_SCM envoyé par le SN à son SCM d'accompagnement pendant les phases opérationnelles et préopérationnelles. Le nombre de tentatives lors du démarrage peut être défini par 100Eh (voir 7.5.4.8).

L'objet 100Dh Intervalle de rafraîchissement de la réinitialisation de sauvegarde est spécifié dans le Tableau 117.

Tableau 117 – Objet 100Dh Intervalle de rafraîchissement de la réinitialisation de sauvegarde

Elément	Valeur
Index	100Dh
Nom	SNMT_ResetGuarding_U32 [a]
Type d'objet	VAR
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	100 000
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	Non
[a] Ce sous-index indique le temps de rafraîchissement pour le service SNMT_SN_reset_guarding_SCM. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

7.5.4.8 Objet 100Eh: Nombre de nouvelles tentatives pour la réinitialisation de sauvegarde

Cet objet contient le nombre de nouvelles tentatives qu'un SN effectue pour contacter ses SCM associés en utilisant la réinitialisation de sauvegarde (voir 7.7.6.3) à l'état préopérationnel.

NOTE L'intervalle entre chaque message est défini par l'objet 100Dh (voir 7.5.4.7).

L'objet 100Eh est facultatif. S'il n'est pas indiqué, un nombre infini de nouvelles tentatives doit être admis par hypothèse.

L'objet 100Eh Nombre de nouvelles tentatives pour la réinitialisation de sauvegarde est spécifié dans le Tableau 118.

Tableau 118 – Objet 100Eh Nombre de nouvelles tentatives pour la réinitialisation de sauvegarde

Elément	Valeur
Index	100Eh
Nom	SNMT_NoResetGuarding_U8 [a]
Type d'objet	VAR
Type de données	UNSIGNED8
Plage de valeurs	UNSIGNED8
Valeur par défaut	Spécifique au fabricant
Catégorie	O
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	Non

[a] Ce sous-index indique le nombre maximal de télégrammes Réinitialisation de sauvegarde envoyés à l'état préopérationnel (voir 7.7.6.3). Si la valeur est 255, le télégramme Réinitialisation de sauvegarde doit être répété à l'infini. Si la valeur est mise entre 0 et 254, les télégrammes Réinitialisation de sauvegarde ne doivent plus être envoyés après le nombre défini de répétitions.

7.5.4.9 Objet 1018h: Informations de fournisseur d'appareil

7.5.4.9.1 Généralités

Cet objet doit contenir les informations de l'appareil spécifiques au fournisseur. Il doit également définir des appareils compatibles. Un appareil qui possède un ID fournisseur identique, un code produit, un numéro de révision majeure et un numéro de révision mineure supérieurs ou égaux à ceux d'un autre appareil doit être compatible. Le fabricant d'un appareil doit vérifier et assurer la compatibilité des modifications de révisions mineures pour un appareil de sécurité.

L'objet 1018h Informations de fournisseur d'appareil est spécifié dans les tableaux ci-après (du Tableau 119 au Tableau 129).

Tableau 119 – Objet 1018h Informations de fournisseur d'appareil

Elément	Valeur
Index	1018h
Nom	SINF_DevVendorInfo_REC
Type d'objet	RECORD
Type de données	SINF_DevVendorInfoRecord_TYPE
Catégorie	M
Pertinent pour la somme de contrôle SOD	Non

Tableau 120 – Objet 1018h sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	9
Valeur par défaut	9
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 121 – Objet 1018h sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	VendorID_32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] ID fournisseur selon la liste de fournisseurs EPSG.	

Tableau 122 – Objet 1018h sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	ProductCode_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Le code produit spécifique au fabricant identifie une version d'appareil spécifique.	

Tableau 123 – Objet 1018h sous-index 03h

Elément	Valeur
Sous-index	03h
Nom	RevisionNumber_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Voir 7.5.4.9.	

Tableau 124 – Objet 1018h sous-index 04h

Elément	Valeur
Sous-index	04h
Nom	SerialNumber_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Numéro de série de l'appareil.	

Tableau 125 – Objet 1018h sous-index 05h

Elément	Valeur
Sous-index	05h
Nom	FirmWareChecksum_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Somme de contrôle utilisée pour vérifier l'intégrité du micrologiciel stocké dans l'appareil. Le calcul est spécifique au fournisseur.	

Tableau 126 – Objet 1018h sous-index 06h

Elément	Valeur
Sous-index	06h
Nom	ParameterChecksum_DOM [a]
Type de données	DOMAIN
Plage de valeurs	DOMAIN
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Cet objet inclut le contenant de la somme de contrôle des paramètres du SOD qui doit être calculée comme décrit en 7.5.4.9.3.	

Tableau 127 – Objet 1018h sous-index 07h

Elément	Valeur
Sous-index	07h
Nom	ParameterTimestamp_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] L'horodatage de ce paramètre doit être le même que celui indiqué dans ParameterChecksum_DOM. L'horodatage de paramètre est créé par l'outil de configuration et est vérifié dans chaque appareil au démarrage par le SCM (voir 7.7.12.2). Le format d'horodatage est la seconde depuis 1.1.1970 (horodatage UNIX).	

Tableau 128 – Objet 1018h sous-index 08h

Elément	Valeur
Sous-index	08h
Nom	Stack_ChkSum_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Somme de contrôle de la mise en œuvre du FSCP 13/1 utilisée pour créer le micrologiciel de l'appareil.	

Tableau 129 – Objet 1018h sous-index 09h

Elément	Valeur
Sous-index	09h
Nom	Stack_Version_U32 [a]
Type de données	UNSIGNED16
Plage de valeurs	UNSIGNED16
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Voir 7.5.4.9.4.	

7.5.4.9.2 Numéro de révision du fabricant

Le numéro de révision spécifique au fabricant doit être constitué d'un numéro de révision majeure et d'un numéro de révision mineure. Le numéro de révision majeure doit identifier le comportement de l'appareil spécifique. Si la fonctionnalité de l'appareil est étendue, le numéro de révision majeure doit être incrémenté. Le numéro de révision mineure doit identifier différentes versions qui ont le même comportement d'appareil.

Le Tableau 130 spécifie le codage de RevisionNumber_U32.

Tableau 130 – Structure du numéro de révision

31	16	15	0
Numéro de révision majeure		Numéro de révision mineure	
Octet de poids fort			Octet de poids faible

7.5.4.9.3 Somme de contrôle des paramètres

L'objet Somme de contrôle des paramètres est un domaine qui doit contenir l'horodatage et un CRC 32 bits comme spécifié dans le Tableau 131 L'horodatage doit également être indiqué dans l'objet 1018h sous-index 7h. Chaque CRC doit être utilisé pour vérifier l'exactitude jusqu'à 4 kbits (4 096 bits/512 octets) des variables SOD. Le polynôme qui doit être utilisé pour les CRC 32 bits dans la somme de contrôle des paramètres est spécifié dans le Tableau 132.

Tableau 131 – Structure du domaine de somme de contrôle des paramètres

Horodatage
CRC32
CRC32
…

Tableau 132 – Polynôme du CRC pour la somme de contrôle des paramètres

Polynôme	Polynôme	Polynôme selon [41]
$x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{23}+x^{22}+x^{20}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+x^{0}$	$11EDC6F41_{hex}$	0x8F6E37A0

Lorsque le SN est passé à l'état Opérationnel, la totalité du SOD doit être vérifiée par rapport aux CRC contenus dans cet objet. La commutation à l'état opérationnel ne doit être admise que si l'horodatage de la commande SNMT et les CRC sont corrects.

Pour calculer le CRC du SOD, seules les données réelles des objets pertinents doivent être utilisées. Les objets marqués comme non pertinents pour le calcul du CRC ne doivent pas être pris en compte dans le calcul. Etant donné que la quantité maximale de données qui peut être utilisée pour chaque CRC est de 4 kbits, plus d'un CRC peut être nécessaire pour sécuriser le SOD d'un appareil important.

La description de l'écriture des paramètres et du contenant de la somme de contrôle dans l'appareil ne relève pas du domaine d'application du présent document. Il est seulement nécessaire de s'assurer que tous les paramètres de SN ont été écrits sur le SN. Etant donné que l'horodatage et les CRC sont associés dans l'objet 1018h sous-index 6h, seul le SCM doit vérifier l'horodatage afin de s'assurer que l'ensemble paramètre/CRC correct est enregistré sur le SN.

7.5.4.9.4 Version de pile

La version de pile contient la version de la pile FSCP 13/1 utilisée pour créer le micrologiciel de l'appareil et doit se composer d'un numéro de version majeur et d'un numéro de version mineur.

Le Tableau 133 spécifie le codage de Stack_Version_U32.

Tableau 133 – Structure de la version de pile

16	0
Numéro de version majeure	Numéro de version mineure
Octet de poids fort	Octet de poids faible

7.5.4.10 Objet 1019h: ID unique d'appareil

L'objet d'index 1019h doit contenir l'identifiant unique d'appareil (UDID). L'UDID doit être constitué de l'ID du fournisseur MAC et de l'ID unique d'appareil du fournisseur.

NOTE Etant donné que le SCM vérifie l'unicité de l'UDID des SN connectés, l'attribution de l'UDID en cours de fabrication par le fournisseur ne constitue pas une étape de production cruciale pour la sécurité.

La composition de l'UDID doit être la suivante:

3 octets d'ID fournisseur de MAC + 3 octets de numéro unique d'appareil, spécifique au fournisseur (numéro de série, par exemple).

EXEMPLE L'ID fournisseur de MAC 06065h et le numéro de série 471112h donnent l'UDID 00-60-65-47-11-12.

L'UDID 00-00-00-00-00-00 doit être réservé.

L'objet 1019h ID unique d'appareil est spécifié dans le Tableau 134.

Tableau 134 – Objet 1019h ID unique d'appareil

Elément	Valeur
Index	1019h
Nom	SINF_UniqueDeviceID_OSTR6 [a]
Type d'objet	VAR
Type de données	OCTET_STRING6
Plage de valeurs	
Valeur par défaut	0
Catégorie	M
Accès	Const
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	Non
[a] L'UDID doit être un identifiant globalement unique composé de 6 octets.	

7.5.4.11 Objet 101Ah: Téléchargement aval de paramètres

Cet objet doit être utilisé pour recevoir un jeu de paramètres spécifique à l'application, ainsi que des jeux de paramètres supplémentaires en provenance du SCM. Le SN doit déterminer le type de jeu de paramètres reçu et réagir en conséquence.

L'objet 101Ah Téléchargement aval de paramètres est spécifié dans le Tableau 135.

Tableau 135 – Objet 101Ah Téléchargement aval de paramètres

Elément	Valeur
Index	101Ah
Nom	SSDO_ParameterDownload_DOM
Type de données	DOMAIN
Plage de valeurs	DOMAIN
Valeur par défaut	0
Catégorie	O
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	Non

Le Tableau 136 spécifie le format dans lequel doivent être transférés les paramètres.

Tableau 136 – Format de Téléchargement aval de paramètres

	Type de données	Longueur	Description
Elément 1	UINT16	2	Index de l'objet
	UINT8	1	Sous-index de l'objet
	UINT32	4	Longueur de l'objet
	DOMAIN	n	Données de l'objet
Elément 2	UINT16	2	Index de l'objet
	UINT8	1	Sous-index de l'objet
	UINT32	4	Longueur de l'objet
	DOMAIN	n	Données de l'objet
	…	…	…

Le Tableau 137 spécifie l'en-tête supplémentaire pour les jeux de paramètres supplémentaires.

Tableau 137 – Format d'en-tête de paramètre supplémentaire

Type de données	Longueur	Description
UINT8	1	Numéro de jeu de paramètres
UINT8	1	Version d'en-tête (réservée pour une utilisation future)
UINT16	2	SAddr de SN
UINT32	4	Nombre d'octets pour un jeu de paramètres sans en-tête
UINT32	4	CRC pour un jeu de paramètres sans en-tête
UINT32	4	Horodatage du jeu de paramètres

7.5.4.12 Objet 101Bh: Paramètres du SCM

L'objet d'index $101B_h$ doit contenir les paramètres spécifiques au SCM.

L'objet 101Bh Paramètres du SCM est spécifié dans le Tableau 138, le Tableau 139 et le Tableau 140.

Tableau 138 – Objet 101Bh Paramètres du SCM

Elément	Valeur
Index	101Bh
Nom	SSCM_SpecificParameters_REC
Type d'objet	RECORD
Type de données	SSCM_SpecificParametersRecord_TYPE
Catégorie	O
Mapping SPDO	Non

Tableau 139 – Objet 101Bh sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	1
Valeur par défaut	1
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 140 – Objet 101Bh sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	ConfigurationMode_U8
Type de données	UNSIGNED8
Plage de valeurs	0 – 1
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Mode de configuration du SCM selon 6.2.2, 0 = ACM, 1 = MCM.	

7.5.4.13 Objet 1200h: Paramètre de communication commun

Cet objet doit contenir les paramètres de communication communs pour l'appareil.

Le sous-index 3 doit contenir la base de temps consécutifs pour le module. Il n'est pas nécessaire qu'un appareil prenne en charge toutes les bases de temps possibles. Au moins une base de temps de 100 µs (type 2) doit être prise en charge.

L'objet 1200h Paramètre de communication commun est spécifié dans les tableaux ci-après (du Tableau 141 au Tableau 146).

Tableau 141 – Objet 1200h Paramètre de communication commun

Elément	Valeur
Index	1200h
Nom	SCOM_CommonCommParam_REC
Type d'objet	RECORD
Type de données	SCOM_CommonCommParamRecord_TYPE
Catégorie	M
Pertinent pour la somme de contrôle SOD	Oui

Tableau 142 – Objet 1200h sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	4
Valeur par défaut	4
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 143 – Objet 1200h sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	SDN_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 à 1 023
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Numéro de domaine de sécurité de l'appareil.	

Tableau 144 – Objet 1200h sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	SADR_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 à 1 023
Valeur par défaut	0
Catégorie	M
Accès	ro
Mapping SPDO	Non
[a] Adresse de sécurité du SCM au sein du SD.	

Tableau 145 – Objet 1200h sous-index 03h

Elément	Valeur
Sous-index	03h
Nom	ConsecutiveTimeBase_I8 [a]
Type de données	INTEGER8
Plage de valeurs	0 – 3
Valeur par défaut	2
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Identification de la base de temps pour le champ CT, 0 = 1 µs, 1 = 10 µs, 2 = 100 µs, 3 = 1 000 µs.	

La base de temps du champ CT doit être calculée au moyen de la Formule (1).

$$TimeBase = 1\ \mu s \times 10^{ConsecutiveTimeBase\_I8} \quad (1)$$

où:

TimeBase est la base de temps du champ CT en µs;

ConsecutiveTimeBase_I8 est l'identifiant de la base de temps.

Tableau 146 – Objet 1200h sous-index 04h

Elément	Valeur
Sous-index	04h
Nom	UDIDofSCM_OSTR6 [a]
Type de données	OCTET_STRING6
Plage de valeurs	OCTET_STRING6
Valeur par défaut	UDID du propre SN
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] UDID du SCM.	

7.5.4.14 Objet 1201h: Paramètre de communication SSDO

Cet objet définit les paramètres de communication SSDO d'un client SSDO.

L'objet 1201h Paramètre de communication SSDO est spécifié dans les tableaux ci-après (du Tableau 147 au Tableau 150).

Tableau 147 – Objet 1201h Paramètre de communication SSDO

Elément	Valeur
Index	1201h
Nom	SSDO_CommParam_REC
Type d'objet	RECORD
Type de données	SSDO_CommParamRecord_TYPE
Catégorie	C
Pertinent pour la somme de contrôle SOD	Non

Tableau 148 – Objet 1201h sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	2
Valeur par défaut	2
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 149 – Objet 1201h sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	Timeout_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	1 à 2^{32}
Valeur par défaut	500 000
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le délai d'expiration d'un télégramme SSDO. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 150 – Objet 1201h sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	Retries_U8 [a]
Type de données	UNSIGNED8
Plage de valeurs	0 – 255
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le nombre de nouvelles tentatives pour un télégramme SSDO.	

7.5.4.15 Objet 1202h: Paramètre de communication SNMT

Cet objet définit les paramètres de communication SNMT d'un maître SNMT.

L'objet 1202h Paramètre de communication SNMT est spécifié dans les tableaux ci-après (du Tableau 151 au Tableau 154).

Tableau 151 – Objet 1202h Paramètre de communication SNMT

Elément	Valeur
Index	1202h
Nom	SNMT_CommParam_REC
Type d'objet	RECORD
Type de données	SNMT_CommParamRecord_TYPE
Catégorie	C
Pertinent pour la somme de contrôle SOD	Non

Tableau 152 – Objet 1202h sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	2
Valeur par défaut	2
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 153 – Objet 1202h sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	Timeout_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	1 à 2^{32}
Valeur par défaut	500 000
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le délai d'expiration d'un télégramme SNMT. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 154 – Objet 1202h sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	Retries_U8 [a]
Type de données	UNSIGNED8
Plage de valeurs	0 – 255
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le nombre de nouvelles tentatives pour un télégramme SNMT.	

7.5.4.16 Objet 1400h à 17FEh: Paramètre de communication RxSPDO

Ces objets définissent la SADR à partir de laquelle doivent être reçues les données.

Un producteur peut posséder plusieurs SADR (une pour chaque TxSPDO). Par conséquent, un consommateur peut avoir plusieurs RxSPDO pour le même producteur physique.

L'objet 1400h à 17FEh Paramètre de communication RxSPDO est spécifié dans les tableaux ci-après (du Tableau 155 au Tableau 168).

Tableau 155 – Objet 1400h à 17FEh Paramètre de communication RxSPDO

Elément	Valeur
Index	1400h – 17FEh
Nom	SPDO_RxCommParam_XXXh_REC
Type d'objet	RECORD
Type de données	SPDO_RxCommParamRecord_TYPE
Catégorie	O
Pertinent pour la somme de contrôle SOD	Oui

Tableau 156 – Objet 1400h à 17FEh sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	0Ch
Valeur par défaut	0Ch
Accès	ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 157 – Objet 1400h à 17FEh sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	SADR_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 à 1 023
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	
[a] Définit la SADR à partir de laquelle doivent être reçues les données RxSPDO. La mise à zéro de SADR désactive le SPDO.	

Tableau 158 – Objet 1400h à 17FEh sous-index 02h

Elément	Valeur
Sous-index	02h
Nom	SCT_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	1 à 2^{32}
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Définit le temporisateur SCT pour les données de ce RxSPDO (voir 7.7.1.2). La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 159 – Objet 1400h à 17FEh sous-index 03h

Elément	Valeur
Sous-index	03h
Nom	NumberOfConsecutiveTReq_U8 [a]
Type de données	UNSIGNED8
Plage de valeurs	1 – 63
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	
[a] Ce sous-index indique le nombre de demandes de temps consécutifs. Pour les configurations de SCT dynamique, la valeur maximale doit toujours être utilisée.	

Tableau 160 – Objet 1400h à 17FEh sous-index 04h

Elément	Valeur
Sous-index	04h
Nom	TimeDelayTReq_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le délai entre deux ensembles de demandes de temps consécutifs. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 161 – Objet 1400h à 17FEh sous-index 05h

Elément	Valeur
Sous-index	05h
Nom	TimeDelaySync_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	
[a] Ce sous-index indique le délai entre une synchronisation temporelle réussie et la demande de temps suivante. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 162 – Objet 1400h à 17FEh sous-index 06h

Elément	Valeur
Sous-index	06h
Nom	MinTSyncPropagationDelay_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 – 65 535 en cas de SCT dynamique sinon: 1 – 65 535
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le délai de propagation minimal admis entre la demande de temps et la réponse de temps. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 163 – Objet 1400h à 17FEh sous-index 07h

Elément	Valeur
Sous-index	07h
Nom	MaxTSyncPropagationDelay_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 – 65 535 en cas de SCT dynamique sinon: 1 – 65 535
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	
[a] Ce sous-index indique le délai de propagation maximal admis entre la demande de temps et la réponse de temps. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 164 – Objet 1400h à 17FEh sous-index 08h

Elément	Valeur
Sous-index	08h
Nom	MinSPDOPropagationDelay_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 – 65 535 en cas de SCT dynamique sinon: 1 – 65 535
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le délai de propagation minimal admis entre l'émission et la réception d'un SPDO. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 165 – Objet 1400h à 17FEh sous-index 09h

Elément	Valeur
Sous-index	09h
Nom	MaxSPDOPropagationDelay_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 – 65 535 en cas de SCT dynamique sinon: 1 – 65 535
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	
[a] Ce sous-index indique le délai de propagation maximal admis entre l'émission et la réception d'un SPDO. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 166 – Objet 1400h à 17FEh sous-index 0Ah

Elément	Valeur
Sous-index	0Ah
Nom	BestCaseTResDelay_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	0 – 65 535
Valeur par défaut	0
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Ce sous-index indique le délai le plus favorable entre l'émission de la demande de temps et l'établissement de la réponse de temps. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 167 – Objet 1400h à 17FEh sous-index 0Bh

Elément	Valeur
Sous-index	0Bh
Nom	TimeRequestCycle_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	1 à 2^{32}
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
Pertinent pour la somme de contrôle SOD	
[a] Ce sous-index indique la durée du cycle de demandes de temps. La base de temps de ce paramètre est définie par CommonCommParam_REC.ConsecutiveTimeBase_I8.	

Tableau 168 – Objet 1400h à 17FEh sous-index 0Ch

Elément	Valeur
Sous-index	0Ch
Nom	TxSPDONo_U16 [a]
Type de données	UNSIGNED16
Plage de valeurs	1 à 1 023
Valeur par défaut	1
Catégorie	M
Accès	rw
Mapping SPDO	Non
[a] Contient le numéro TxSPDO avec lequel la demande de temps doit être envoyée.	

7.5.4.17 Objet 1800h à 1BFEh: Paramètre de mapping RxSPDO

Cet objet définit les paramètres de mapping RxSPDO. Les paramètres de mapping peuvent être prédéfinis par le fournisseur et être en lecture seule. Une description complète du mapping SPDO est fournie en 7.5.4.24.

L'objet 1800h à 1BFEh Paramètre de mapping RxSPDO est spécifié dans les tableaux ci-après (du Tableau 169 au Tableau 172).

Tableau 169 – Objet 1800h à 1BFEh Paramètre de communication RxSPDO

Elément	Valeur
Index	1800h – 1BFEh
Nom	SPDO_RxMappParam_XXXh_AU32
Type d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	O
Pertinent pour la somme de contrôle SOD	Oui

Tableau 170 – Objet 1800h à 1BFEh sous-index 00h

Elément	Valeur
Sous-index	00h
Nom	NumberOfEntries [a]
Plage de valeurs	0 – 254
Valeur par défaut	0
Accès	rw/ro
Mapping SPDO	Non
[a] Nombre d'entrées dans cet index.	

Tableau 171 – Objet 1800h à 1BFEh sous-index 01h

Elément	Valeur
Sous-index	01h
Nom	SPDOMapping_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	M
Accès	rw/ro
Mapping SPDO	Non
[a] Ce sous-index indique le paramètre de mapping pour ce RxSPDO.	

Tableau 172 – Objet 1800h à 1BFEh sous-index 02h à FEh

Elément	Valeur
Sous-index	02h à FEh
Nom	SPDOMapping_U32 [a]
Type de données	UNSIGNED32
Plage de valeurs	UNSIGNED32
Valeur par défaut	0
Catégorie	O
Accès	rw/ro
Mapping SPDO	Non
[a] Ce sous-index indique le paramètre de mapping pour ce RxSPDO.	

7.5.4.18 Objet 1C00h à 1FFEh: Paramètre de communication TxSPDO

Chaque SN doit posséder au moins un TxSPDO, soit pour l'émission de données (producteur) soit pour la demande de temps (consommateur).

L'objet 1C00h à 1FFEh Paramètre de communication TxSPDO est spécifié dans les tableaux ci-après (du Tableau 173 au Tableau 177).

Tableau 173 – Objet 1C00h à 1FFEh Paramètre de communication TxSPDO

Elément	Valeur
Index	1C00h – 1FFEh
Nom	SPDO_TxCommParam_XXXh_REC
Type d'objet	RECORD
Type de données	SPDO_TxCommParamRecord_TYPE
Catégorie	M/O
Pertinent pour la somme de contrôle SOD	Oui