IEC 61784-3-19

Edition 1.0 2024-10

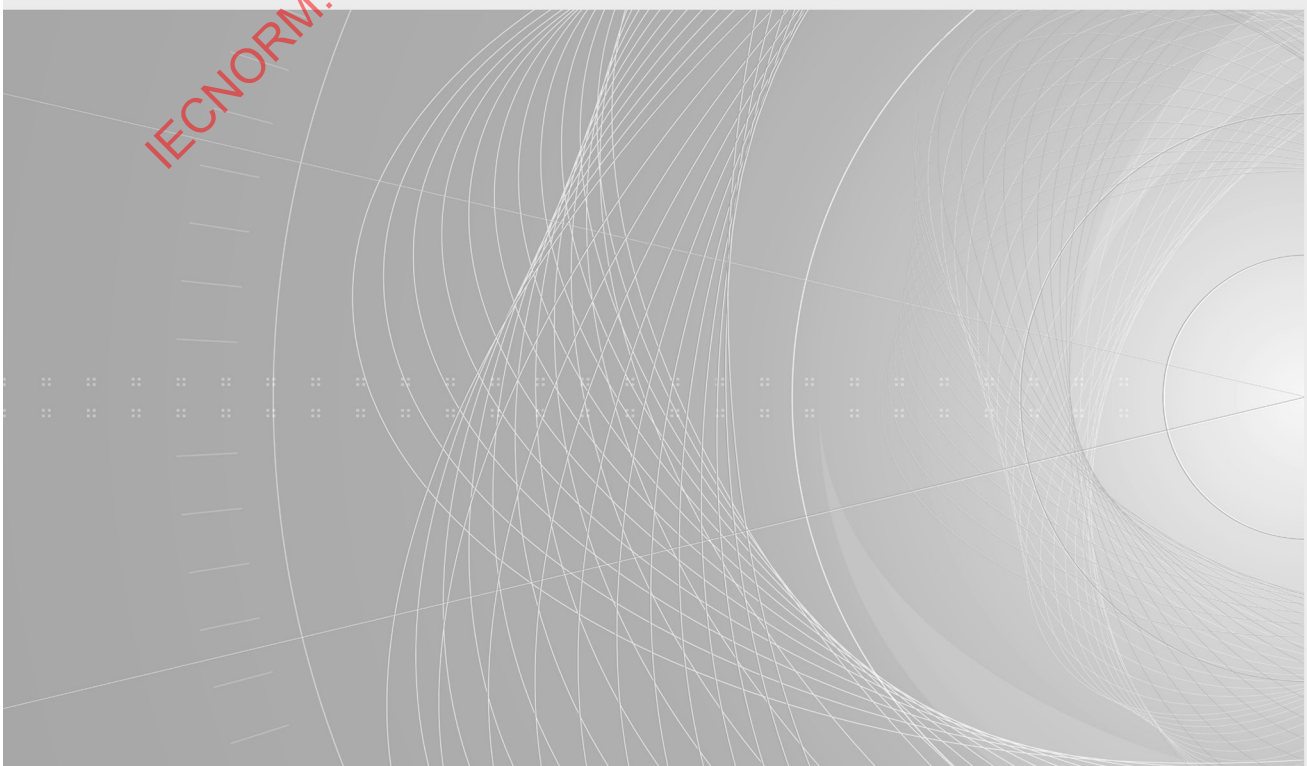
INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Industrial communication networks – Profiles – Part 3-19: Functional safety fieldbuses – Additional specifications for CPF 19

Réseaux de communication industriels – Profils – Partie 3-19: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 19



IEC 61784-3-19:2024-10(en-fr)

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

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

IEC 61784-3-19

Edition 1.0 2024-10

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Industrial communication networks – Profiles – Part 3-19: Functional safety fieldbuses – Additional specifications for CPF 19

Réseaux de communication industriels – Profils – Partie 3-19: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 19

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 25.040.40, 35.100.05

ISBN 978-2-8322-9802-2

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

INDUSTRIAL COMMUNICATION NETWORKS – PROFILES –

Part 3-19: Functional safety fieldbuses – Additional specifications for CPF 19

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) IEC draws attention to the possibility that the implementation of this document may involve the use of a patent. IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had received notice of a patent, which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 61784-3-19 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

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

Draft	Report on voting
65C/1276/CDV	65C/1298/RVC

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/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 webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

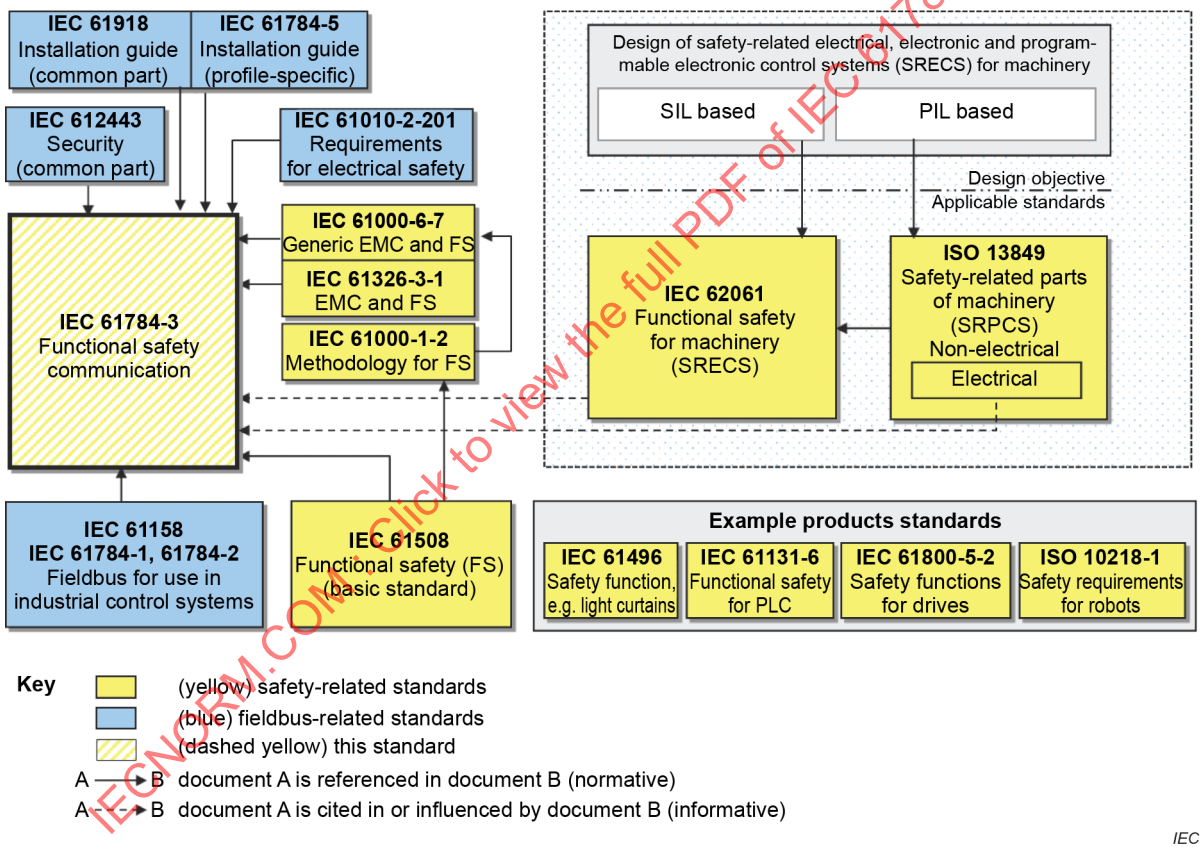
IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The IEC 61158 fieldbus standard series together with its companion standards series 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.

The IEC 61784-3 series explains the relevant principles for functional safety communications with reference to the IEC 61508 series and specifies several safety communication layers (profiles and corresponding protocols) based on the communication profiles and protocol layers of the IEC 61784-1, IEC 61784-2 and IEC 61158 series. 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.

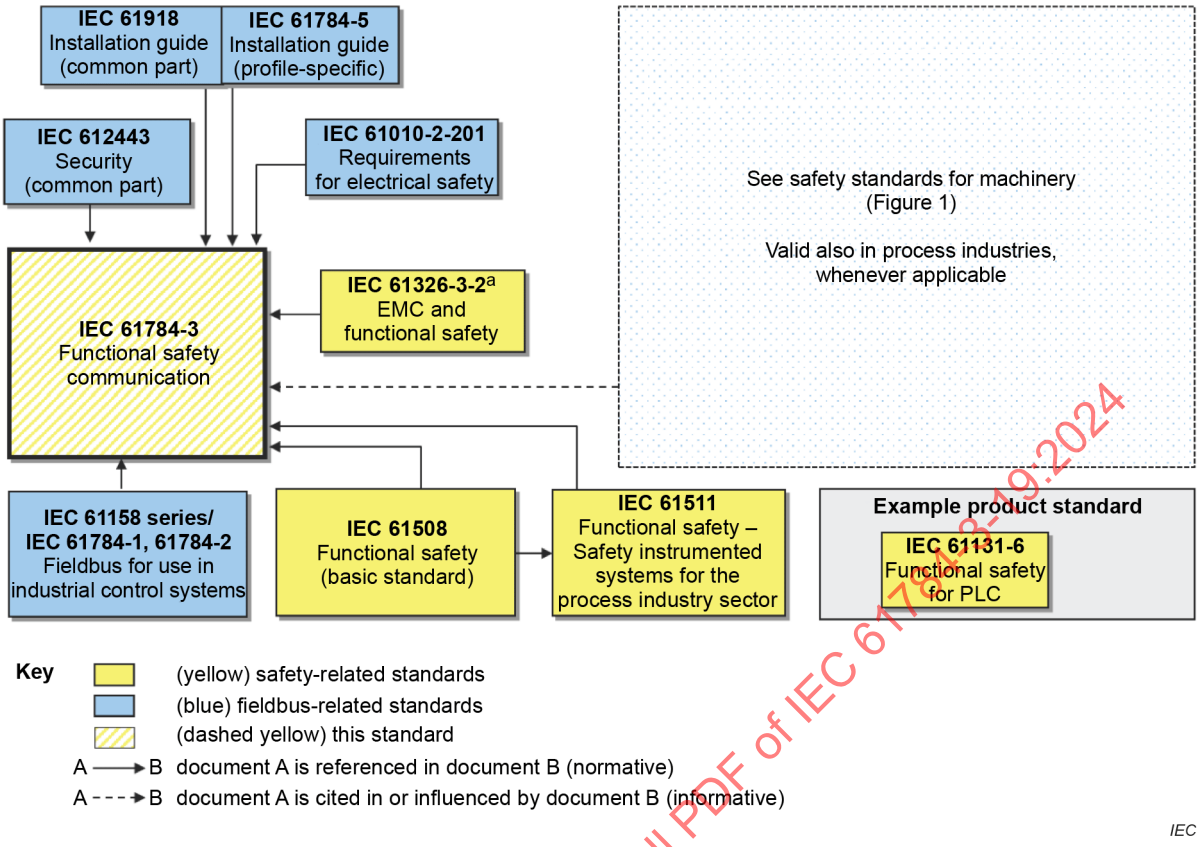
Figure 1 shows the relationships between the IEC 61784-3 series and relevant safety and fieldbus standards in a machinery environment.



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 the IEC 61784-3 series 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 the IEC 61508 series 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 the IEC 61784-3 series 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.

The IEC 61784-3 series describes:

- basic principles for implementing the requirements of the IEC 61508 series 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 the IEC 61784-1 and IEC 61784-2 series, including safety layer extensions to the communication service and protocols sections of the IEC 61158 series.

INDUSTRIAL COMMUNICATION NETWORKS – PROFILES –

Part 3-19: Functional safety fieldbuses – Additional specifications for CPF 19

1 Scope

This part of IEC 61784-3 specifies a safety communication layer (services and protocol) based on IEC 61784-1-19, IEC 61784-2-19 and the IEC 61158 series (Type 24 and Type 27). 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 the IEC 61508 series[1] for functional safety. These mechanisms can 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 61131-2, *Industrial-process measurement and control – Programmable controllers – Part 2: Equipment requirements and tests*

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

IEC 61158-6-24, *Industrial communication networks – Fieldbus specifications – Part 6-24: Application layer protocol specification – Type 24 elements*

IEC 61158-6-27, *Industrial communication networks – Fieldbus specifications – Part 6-27: Application layer protocol specification – Type 27 elements*

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*

[1] In the following pages of this document, "IEC 61508" will be used for "the IEC 61508 series".

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*

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

IEC 61511 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*

IEC 61784-1-19:2023, *Industrial networks – Profiles – Part 1-19: Fieldbus profiles – Communication Profile Family 19*

IEC 61784-2-19:2023, *Industrial networks – Profiles – Part 2-19: Additional real-time fieldbus profiles based on ISO/IEC/IEEE 8802-3 – CPF 19*

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

IEC 61784-5-19, *Industrial networks – Profiles – Part 5: Installation of fieldbuses – Installation profiles for CPF 19*

IEC 62061, *Safety of machinery – Functional safety of safety-related control systems*

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 terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at https://www.electropedia.org/
- ISO Online browsing platform: available at https://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
communication channel
logical *connection* between two end-points within a *communication system*

3.1.1.2
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

3.1.1.3
connection
logical binding between two application objects within the same or different devices

3.1.1.4
CRC
Cyclic Redundancy Check
<value> redundant data derived from, and stored or transmitted together with, a block of data in order to detect data corruption

3.1.1.5
CRC
Cyclic Redundancy Check
<method> procedure used to calculate the redundant data

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

3.1.1.6
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 can 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 to entry have been added.]

3.1.1.7
failure
termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required

Note 1 to entry: Failure can 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 have been replaced.]

3.1.1.8
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 has been deleted.]

3.1.1.9
frame
denigrated synonym for DLPDU

3.1.1.10
master
communication entity able to initiate and schedule communication activities by other stations which may be masters or *slaves*

3.1.1.11
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]

3.1.1.12
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 have been deleted.]

3.1.1.13
safety communication channel
SC
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 modeled as two SCLs connected by a *black channel* or a *defined communication system*, or a *defined channel*.

3.1.1.14
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

3.1.1.15
safety connection
connection that utilizes the safety protocol for communications transactions

3.1.1.16
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.

3.1.1.17
safety device
device designed in accordance with IEC 61508 and which implements the functional safety communication profile

3.1.1.18
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 have been deleted.]

3.1.1.19
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.

3.1.1.20
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 IEC 61508-1:2010, Tables 2 and 3.

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]

3.1.1.21
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.

3.1.1.22
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*.

3.1.1.23
safety-related application
programs designed in accordance with IEC 61508 to meet the SIL requirements of the application

3.1.1.24
safety-related system
system performing *safety functions* according to IEC 61508

3.1.1.25
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

3.1.2 CPF 19: Additional terms and definitions

3.1.2.1
safety master
active communication entity which transmits *safety output data* to a safety slave and receives safety input data from a safety slave

3.1.2.2
safety slave
active communication entity which receives safety output data from a safety master and transmits safety output data to a safety master

3.1.2.3
non-safety master
active communication entity which transmits non-safety output data to a non-safety slave and receives non-safety input data from a non-safety slave

3.1.2.4
non-safety slave
active communication entity which receives non-safety output data from a non-safety master and transmits non-safety output data to a non-safety master

3.1.2.5
node address
identifier of each communication entity

3.1.2.6
safety output data
SPDU transmitted from a safety master to a safety slave

3.1.2.7
safety input data
SPDU transmitted from a safety slave to a safety master

3.1.2.8
safety output user data
user defined data in safety output data

3.1.2.9
safety input user data
user defined data in safety input data

3.1.2.10
fail safe state
state to which safety masters and safety slaves transit in case of error

Note 1 to entry: In this state, safety output user data and safety input user data are all "0".

3.2 Symbols and abbreviated terms

3.2.1 Common symbols and abbreviated terms

CP	communication profile	[IEC 61784-1 (all parts)]
CPF	communication profile family	[IEC 61784-1 (all parts)]
CRC	cyclic redundancy check	
DLL	data link layer	[ISO/IEC 7498-1]
DLPDU	data link protocol data unit	
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)]
FSCP	functional safety communication profile	
PDU	protocol data unit	[ISO/IEC 7498-1]
Pe	bit error probability	
PhL	physical layer	[ISO/IEC 7498-1]
PL	performance level	[ISO 13849-1]
SC	safety communication channel	
SCL	safety communication layer	
SIL	safety integrity level	[IEC 61508-4:2010]
SPDU	safety PDU	

3.2.2 CPF 19: Additional symbols and abbreviated terms

NVS non-volatile storage

LSB least significant bit

3.3 Conventions

Conventions used in this document are defined in IEC 61784-1-19, IEC 61784-2-19, and the IEC 61158 series Type 24 and 27.

4 Overview of FSCP 19 (MECHATROLINK Safety)

CPF 19 (commonly known as MECHATROLINK™ [2]) is defined in IEC 61784-1-19 and IEC 61784-2-19 with fieldbus protocol layers defined in the IEC 61158 series Type 24 and 27. Functional safety communication profile FSCP 19 (MECHATROLINK Safety) is based on CPF 19 and the SCL defined in this part.

An overview of the system structure for FSCP 19 is shown in Figure 3. A non-safety master is shown in control domain 1, it is responsible for establishing and managing the non-safety communication channels in a master-slave relationship with the non-safety slaves using CPF 19.

Safety masters, as shown in control domains 2 and 3, are responsible for establishing and managing the safety communication channels in a master-slave relationship with the safety slaves using FSCP 19. Each safety master shall be limited to a single safety-related control domain.

FSCP 19 allows non-safety communication and safety communication to co-exist. A device can implement the non-safety communication function, the safety communication function, or both.

[2] MECHATROLINK™ is a trade name of Yaskawa Electric Corporation. 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 name MECHATROLINK™. Use of the trade name MECHATROLINK™ requires permission of Yaskawa Electric Corporation and compliance with conditions for their use (such as testing and validation).

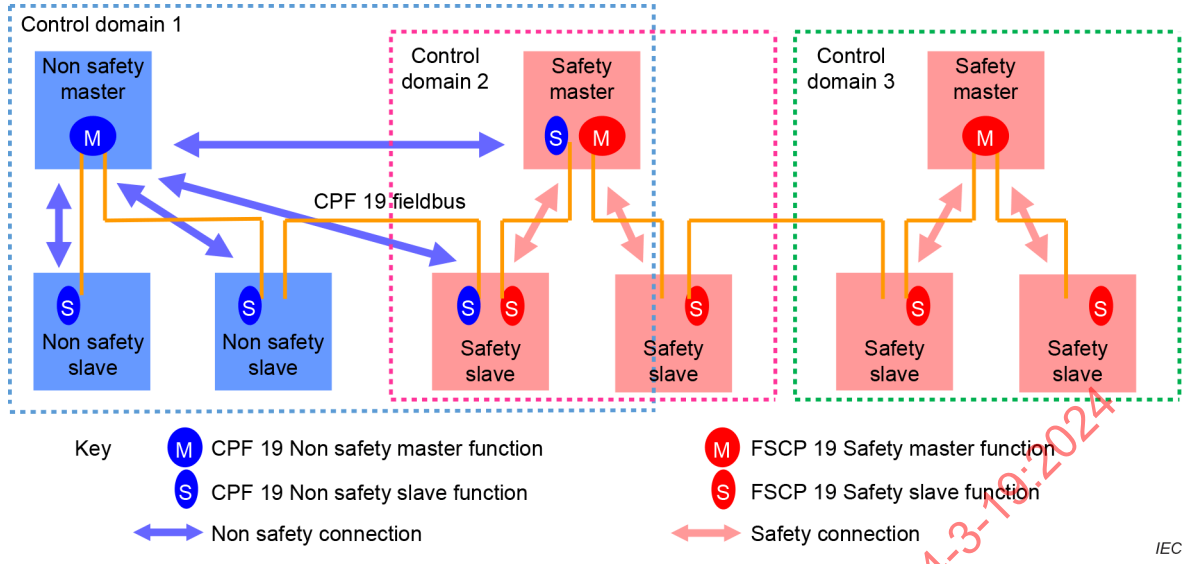


Figure 3 – Basic FSCP 19 system

5 General

5.1 External documents providing specifications for the profile

Additional documents providing more information are under preparation.

5.2 Safety functional requirements

This document specifies the services and protocols for a functional safety communication system based on fieldbus CPF 19. The communication technologies specified in this document shall only be implemented in devices designed in accordance with the requirements of IEC 61508.

The following requirements shall apply to the development of devices that implement FSCP 19 protocols. The same requirements were used in the development of FSCP 19.

- The FSCP 19 protocols are designed to support Safety Integrity Level SIL 3 (refer to IEC 61508).
- Implementations of FSCP 19 shall comply with IEC 61508.
- The basic requirements for the development of the FSCP 19 protocol are in IEC 61784-3.
- Environmental conditions shall comply with IEC 61131-2 for the basic levels and IEC 61326-3-1, IEC 61326-3-2 for the safety margin tests, unless there are specific product standards.
- Unless specified in this part, the CPF 19 requirements shall be unchanged for safety.

5.3 Safety measures

5.3.1 General

Safety measures used in the FSCP 19 are shown in Table 1.

Table 1 – Communication errors and safety measures

Communication errors	Safety measures				
	Sequence number (See 5.3.2)	Time expectation (See 5.3.3)	Connection ID (See 5.3.4)	CRC (See 5.3.5)	Redundancy with cross checking (See 5.3.6)
Corruption				X	X
Unintended repetition	X				
Incorrect sequence	X				
Loss	X	X			
Unacceptable delay		X			
Insertion	X		X		
Masquerade	X		X	X	X
Addressing			X		
Repetition by memory failures within non-safety network products	X				

5.3.2 Sequence number

To detect a network failure, a safety master and safety slaves shall have four types of sequence number for each connection instance, as shown in Table 2.

Table 2 – Sequence number list

Sequence number	Size	Description
Master sequence number	16 bits	Included in SPDU Transmitted from safety master to safety slave Checked by safety slave Initial value is 0x 0001
Extended master sequence number	32 bits	Not included in SPDU Checked by safety slave Initial value is 0x 0000 0000
Slave sequence number	16 bits	Included in SPDU Transmitted from safety slave to safety master Checked by safety master Initial value is 0x 0001
Extended slave sequence number	32 bits	Not included in SPDU Checked by safety master Initial value is 0x 0000 0000

The master sequence number shall be incremented by 1 after the safety master transmits an SPDU. The slave sequence number shall be incremented by 1 after the safety slave transmits an SPDU. Each sequence number shall be incremented from an initial value 1, and this 16-bit value shall rollover; the value after $2^{16}-1$ is 1. Value 0 shall not be used.

When the master sequence number rolls over from $2^{16}-1$ to 1, the extended master sequence number shall be incremented by 1. When the slave sequence number rolls over from $2^{16}-1$ to 1, the extended slave sequence number shall be incremented by 1. Each extended sequence number shall be incremented from initial value 0, and this 32-bit value shall rollover; the value after $2^{32}-1$ is 0.

The safety master and the safety slave shall initialize these variables before establishing a connection. The incrementing procedure of sequence number is shown in Figure 4.

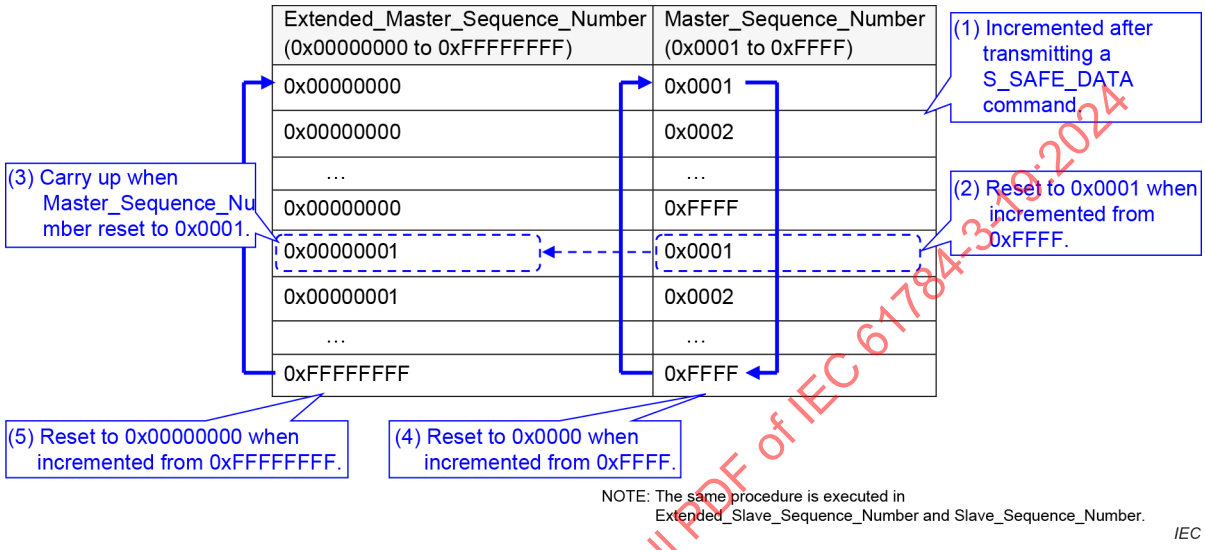


Figure 4 – Incrementing procedure of sequence number

Each sequence number shall be incremented and checked as described above in the transmitting and receiving process for all commands except:

- sequence number in S_NOP command shall always be zero;
- sequence number in S_FAIL_SAFE command shall hold its last value prior to the state transition of the SCL to the fail safe state.

5.3.3 Time expectation

Upon normal reception and then transmission a S_SAFE_DATA command, the SCL shall start a watchdog timer and reset it upon normal reception of a subsequent S_SAFE_DATA command as shown in Figure 5.

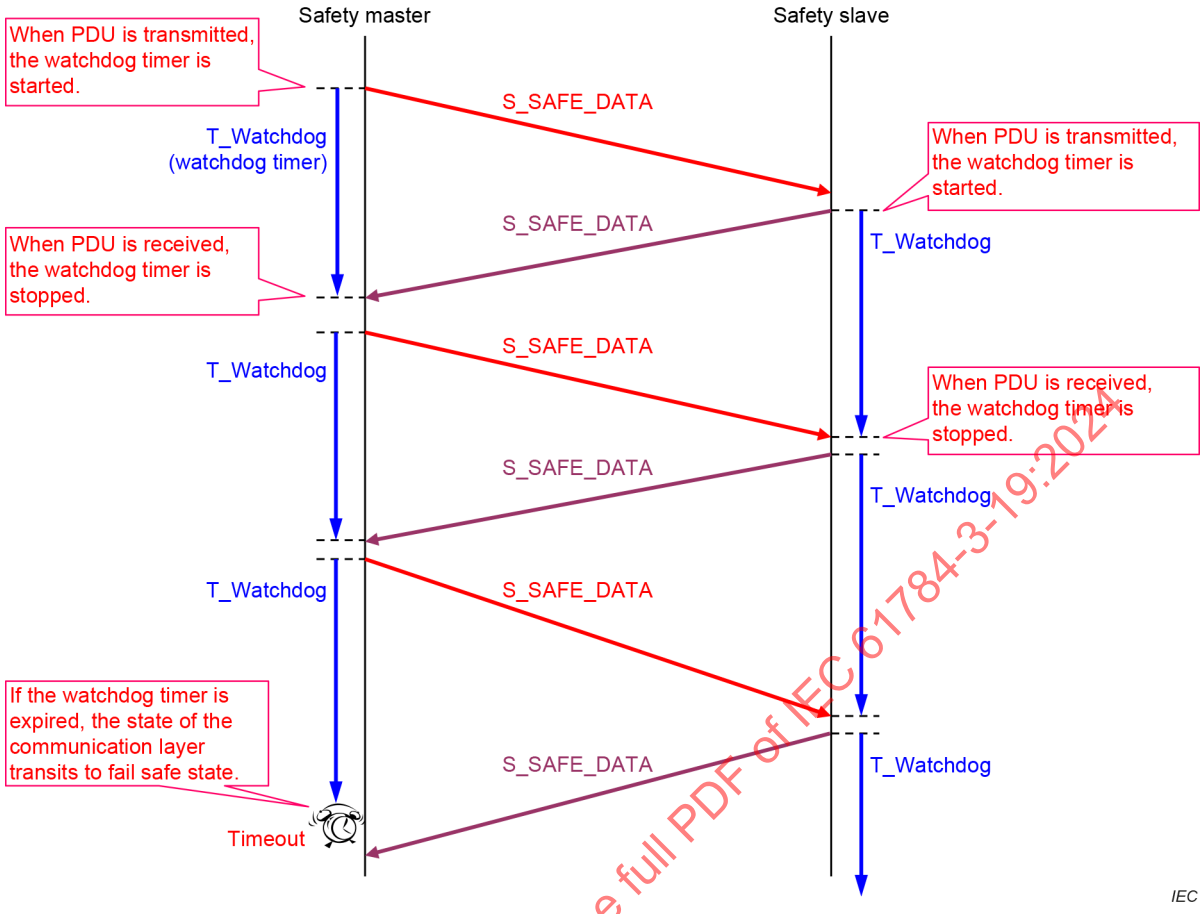


Figure 5 – Time expectation with watchdog timer

The expiration of this watchdog timer triggers an SCL transition to the fail safe state, and the SCL shall indicate this event to the safety application. The value of T_Watchdog is configured by the user with a configuration tool.

The transmitting timing of the S_SAFE_DATA command shall be synchronized between a safety master and safety slaves and the relative time between them shall be fixed, as shown Figure 6.

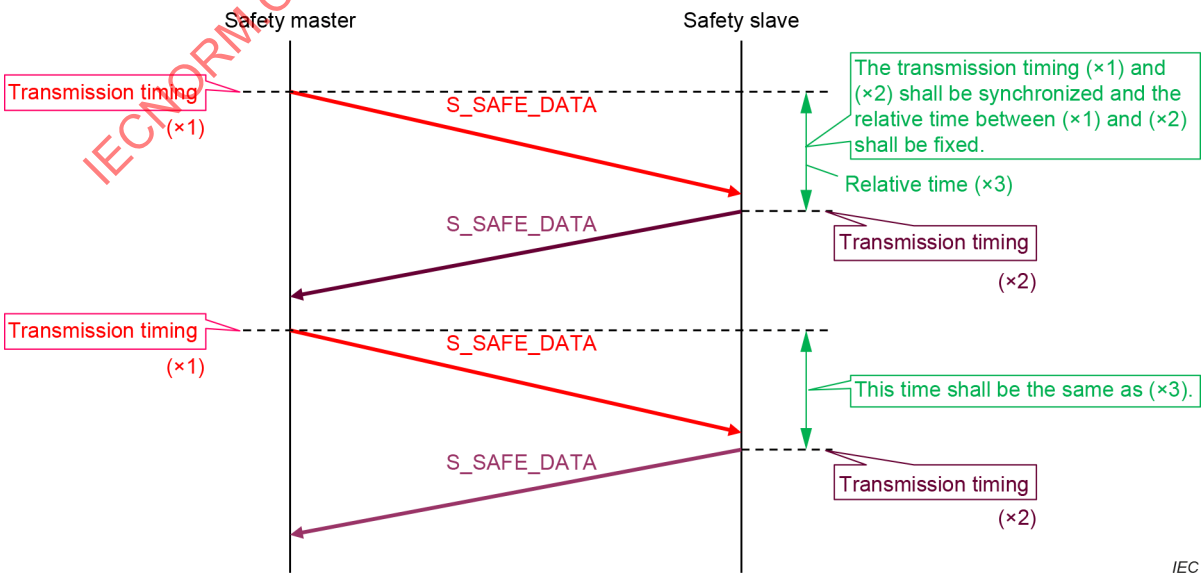


Figure 6 – Synchronization of transmission timing

In command other than S_SAFE_DATA, the safety master shall start a response timer upon the transmission of an SPDU, and monitor the time until receiving the response from the safety slave as shown in Figure 7.

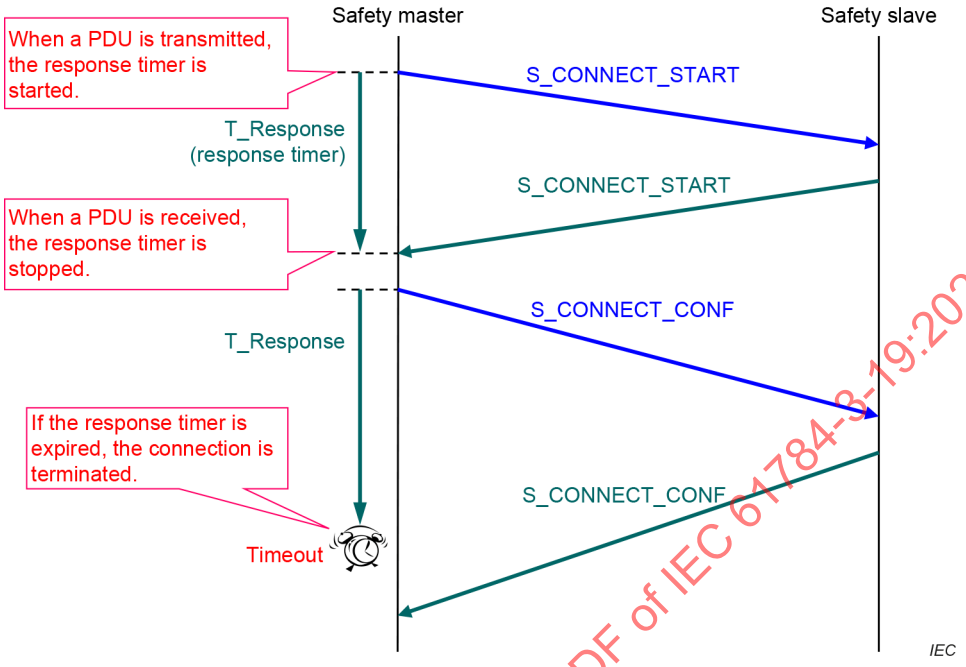


Figure 7 – Time expectation with response timer

The safety master does not monitor S_NOP or S_FAIL_SAFE commands. The safety slave only monitors S_SAFE_DATA commands.

The expiration of this response timer shall trigger the safety master SCL to terminate connection. The value of T_Response is configured by the user with a configuration tool.

5.3.4 Connection ID

During connection negotiation, the safety master and safety slave exchange connections keys from which the safety master generates a unique connection ID. This connection ID is transmitted to the safety slave and is then used by both for connection authentication. Refer to 7.3.1.2 for details.

5.3.5 CRC calculation

A CRC value shall be calculated for each block in an SPDU with unique generator polynomials used for each block. In the CRC calculation for Block 1, the polynomial of $g_1(x)$ = 0x90022004 shall be used as shown in Formula (1).

$$g_1(x) = x^{32} + x^{29} + x^{18} + x^{14} + x^3 + 1 \quad (1)$$

In the CRC calculation for Block 2, the polynomial of $g_2(x)$ = 0x992C1A4C shall be used as shown in Formula (2).

$$g_2(x) = x^{32} + x^{29} + x^{28} + x^{25} + x^{22} + x^{20} + x^{19} + x^{13} + x^{12} + x^{10} + x^7 + x^4 + x^3 + 1 \quad (2)$$

In the polynomials $g_1\ (x)$ and $g_2\ (x)$, the minimum Hamming distance is 6 when the data size (including CRC) is less than or equal to 4 092 bytes. If the data size is greater than 4 092 bytes, the minimum Hamming distance is less than 6, and residual error rate shall be calculated.

CRC generator seed values are unique for each block as shown in Table 3.

Table 3 – CRC seed values

Block	Seed
1	0x FFFF FFFF
2	0x FFFF FFFE

The CRC shall be calculated from the LSB, in the following order:

1) Command
2) Connection ID
3) Sequence number
4) State number
5) Data

5.3.6 Redundancy with cross checking

5.3.6.1 Redundant data generation

The redundant data shall be generated according to the following procedure with the comparison checks shown in **bold**:

1) Load data in Block 1 for each channel;
2) Copy data from Block 1 to Block 2 for each channel;
3) **Compare data in Block 1 with Block 2**;
4) Calculate CRC 1 and CRC 2 values for Block 1 and Block 2 as described in 5.3.5;
5) For channel B, calculate CRC 3 with CRC 1 and CRC 2 and transfer CRC 1, CRC 2, and CRC 3 from channel B to channel A;
6) For channel A, calculate CRC 3 with the CRC 1 and CRC 2 transferred from channel B; **compare the calculated CRC 3 in channel A with transferred CRC 3 from channel B**;
7) **Compare CRC 1 and CRC 2 calculated in channel A with that of channel B**;
8) Notify check result from channel A to channel B with its bit inversion data; check result has 8 bits where 0x01 indicates the result is OK and 0x00 indicates the result is NG;
9) Verify check result notified from channel A; take NOT of the bit inversion data notified from channel A and check whether it is the same as the check result notified from channel A or not;
10) Complete generation process in channel B;
11) Notify check result from channel B to channel A with its bit inversion data; check result has 8 bits where 0x01 indicates the result is OK and 0x00 indicates the result is NG;
12) Verify check result notified from channel B; take NOT of the bit inversion data notified from channel B and check whether it is the same as the check result notified from channel B or not;
13) Complete generation process in channel A.

If all comparison checks pass, the CRC 1 and CRC 2 are used as the CRCs in the transmission SPDU.

The redundant data generation processing is shown in Figure 8.

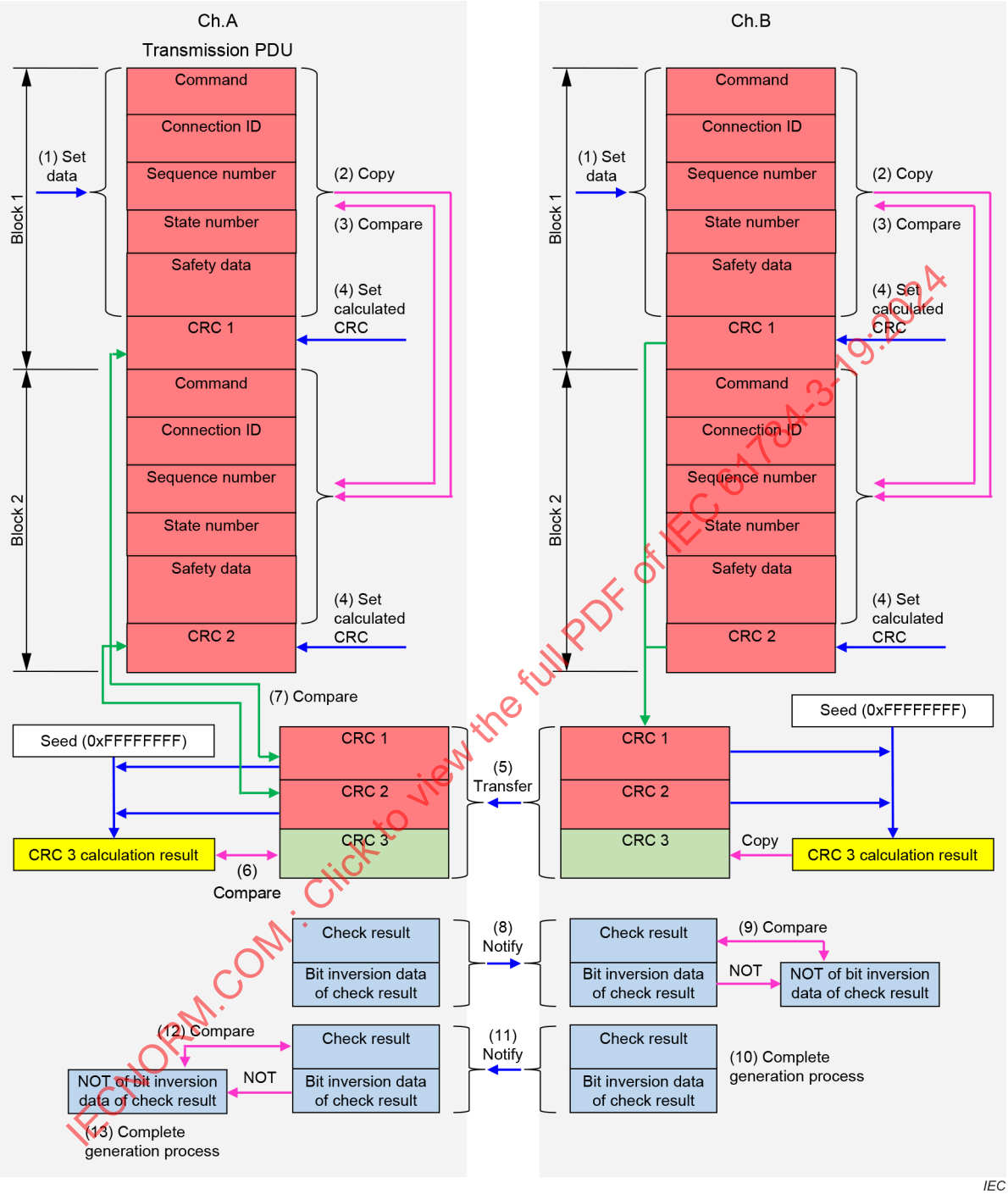


Figure 8 – Redundant data generation processing

5.3.6.2 Redundant data verification

Upon receipt, the redundant data shall be verified according to the following procedure with the comparison checks shown in **bold**:

1) Verify Block 1 using CRC 1 and Block 2 using CRC 2;
2) Compare redundant data in Block 1 with that of Block 2;
3) For channel B, calculate CRC 3 with CRC 1 and CRC 2 and transfer CRC 1, CRC 2, and CRC 3 from channel B to channel A;

4) For channel A, calculate CRC 3 with the CRC 1 and CRC 2 transferred from channel B; **compare the calculated CRC 3 in channel A with transferred CRC 3 from channel B**;
5) **Compare CRC 1 and CRC 2 calculated in channel A with that of channel B**;
6) Notify check result from channel A to channel B with its bit inversion data; check result has 8 bits where 0x01 indicates the result is OK and 0x00 indicates the result is NG;
7) Verify check result notified from channel A; take NOT of the bit inversion data notified from channel A and check whether it is the same as the check result notified from channel A or not;
8) Complete verification process in channel B;
9) Notify check result from channel B to channel A with its bit inversion data; check result has 8 bits where 0x01 indicates the result is OK and 0x00 indicates the result is NG;
10) Verify check result notified from channel B; take NOT of the bit inversion data notified from channel B and check whether it is the same as the check result notified from channel B or not;
11) Complete verification process in channel A.

If all verifications and comparison checks pass, the SPDU is verified as valid.

The redundant data verification process is shown in Figure 9.

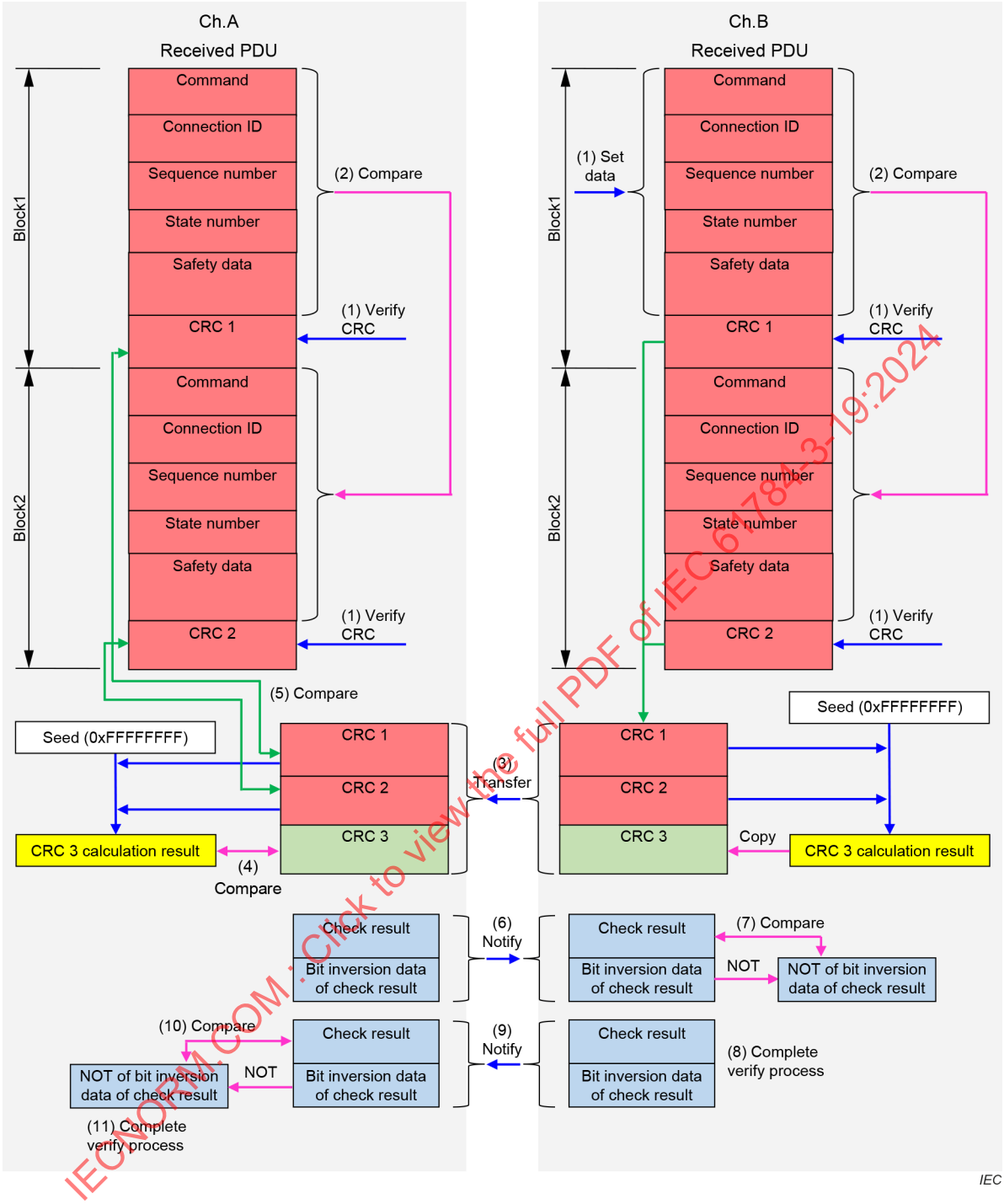


Figure 9 – Redundant data verification process

5.4 Safety communication layer structure

The FSCP 19 SCL is specified on top of the CPF 19 FAL, as shown in Figure 10.

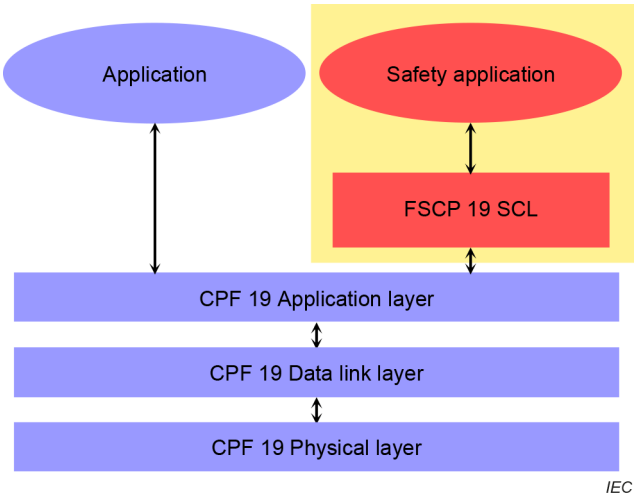


Figure 10 – SCL structure

The FSCP 19 SCL requires that the CPF 19 FAL is configured with a minimum user data size of 32 bytes.

5.5 Relationships with FAL (and DLL, PhL)

5.5.1 General

There are no FAL requirements other than those stated in this document. FSCP 19 can be implemented on the FAL of any CPF 19 profile.

5.5.2 Data types

For information about data types, see 7.1, IEC 61158-6-24, and IEC 61158-6-27.

6 Safety communication layer services

6.1 Service description

6.1.1 S_CONNECT_START

The safety master and the safety slave shall exchange their connection key for generating the connection ID. The safety master shall generate a random number, and set it in an S_CONNECT_START command SPDU as the master connection key. The safety master then transmits to the safety slave, and the safety slave saves the master connection key and transmits the generated random number as slave connection key. The data of S_CONNECT_START command is shown in Table 4.

Table 4 – S_CONNECT_START command data

byte	Name	Description
0	Master connection key (bit 0-7)	The safety master sets generated random number. The safety slave sets the value received from the safety master. This value is used to generate the Connection_Id in the safety master.
1	Master connection key (bit 8-15)	
2	Slave node address (bit 0-7)	The safety master sets the slave node address registered in the configuration parameter. The safety slave set its own node address. This value is used to confirm conformity with the node address in the safety slave.
3	Slave node address (bit 8-15)	
4	Slave connection key (bit 0-7)	Safety master: set 0. Safety slave: set generated random number. This value is used to generate the Connection_Id in the safety master.
5	Slave connection key (bit 8-15)	

Table 5 and Table 6 show an example with SPDU size 16 bytes. In this example, the data is fragmented into 2 SPDUs. The data is packed starting with the least significant byte and space at the end shall be loaded with 0x00 (see bytes 10 and 11 in the example).

Table 5 – S_CONNECT_START command SPDU (1st SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_CONNECT_START = 0x01	S_CONNECT_START = 0x01
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7) = 0x00	Connection ID (bit 0-7) = 0x00
3	Connection ID (bit 8-15) = 0x00	Connection ID (bit 8-15) = 0x00
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Master connection key (bit 0-7) = random number	Master connection key (bit 0-7)
9	Master connection key (bit 8-15) = random number	Master connection key (bit 8-15)
10	Slave node address (bit 0-7)	Slave node address (bit 0-7)
11	Slave node address (bit 8-15)	Slave node address (bit 8-15)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

Table 6 – S_CONNECT_START command SPDU (2nd SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_CONNECT_START = 0x01	S_CONNECT_START = 0x01
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7) = 0x00	Connection ID (bit 0-7) = 0x00
3	Connection ID (bit 8-15) = 0x00	Connection ID (bit 8-15) = 0x00
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Slave connection key (bit 0-7) = 0x00	Slave connection key (bit 0-7) = random number
9	Slave connection key (bit 8-15) = 0x00	Slave connection key (bit 8-15) = random number
10	0x00	0x00
11	0x00	0x00
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

6.1.2 S_CONNECT_CONF

The safety master shall notify the safety slave of the connection ID generated by the master connection key and the slave connection key.

The safety master shall calculate the CRC for the exchanged master connection key and slave connection key with S_CONNECT_START command SPDU, and send this CRC as the connection ID to the safety slave with S_CONNECT_CONF command SPDU. The safety master shall calculate the CRC from the LSB in the order of master connection key and slave connection key using the CCITT-16 polynomial $g_3(x)$ shown in Formula (3). The initial value for this calculation is 0xFFFF.

$$g_3(x) = x^{16} + x^{12} + x^5 + x^1 \tag{3}$$

The safety slave, upon receipt of the S_CONNECT_CONF command, shall calculate the CRC for its master connection key and slave connection key, and compare it with the connection ID sent by S_CONNECT_CONF command PDU. If the calculated CRC is same as the connection ID, and the slave device information (vender ID, device code) received in the S_CONNECT_CONF command is the same as its own information, the safety slave shall send the safety slave SPDU to the safety master. If these values are not the same, the safety slave shall send the S_DISCONNECT command to the safety master.

The data of the S_CONNECT_CONF command is shown in Table 7. If the SPDU length is too short for the complete data set, the safety master shall fragment the data into SPDU size and transmit the fragmented SPDUs.

Table 7 – S_CONNECT_CONF command data

byte	Name	Description
0	Connection ID (bit 0-7)	The safety master sets the Connection_Id. The safety slave sets the Connection ID received from the safety master. This value is used to authenticate the safety connection in the safety master and the safety slave.
1	Connection ID (bit 8-15)	
2	Slave node address (bit 0-7)	The safety master sets the slave node address registered in the configuration parameter. The safety slave set its own node address. This value is used to confirm the conformity with the node address in the safety slave.
3	Slave node address (bit8-15)	
4	Slave vendor ID (bit 0-7)	The safety master sets the slave vendor ID registered in the configuration parameter. The safety slave sets the value received from the safety master. This value is used to verify the slave information in the safety slave.
5	Slave vendor ID (bit 8-15)	
6	Slave vendor ID (bit 16-23)	
7	Slave vendor ID (bit 24-31)	
8	Slave device code (bit 0-7)	The safety master sets the slave device code registered in the configuration parameter. The safety slave sets the value received from the safety master. This value is used to verify the slave information in the safety slave.
9	Slave device code (bit 8-15)	
10	Slave device code (bit 16-23)	
11	Slave device code (bit 24-31)	

An example where the SPDU size is 16 bytes and data is fragmented into 3 SPDUs is shown in Table 8, Table 9, and Table 10.

Table 8 – S_CONNECT_CONF command SPDU (1st SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_CONNECT_CONF = 0x02	S_CONNECT_CONF = 0x02
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Connection ID (bit 0-7)	Connection ID (bit 0-7)
9	Connection ID (bit 8-15)	Connection ID (bit 8-15)
10	Slave node address (bit 0-7)	Slave node address (bit 0-7)
11	Slave node address (bit 8-15)	Slave node address (bit 8-15)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

Table 9 – S_CONNECT_CONF command SPDU (2nd SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_CONNECT_CONF = 0x02	S_CONNECT_CONF = 0x02
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Slave vendor ID (bit 0-7)	Slave vendor ID (bit 0-7)
9	Slave vendor ID (bit 8-15)	Slave vendor ID (bit 8-15)
10	Slave vendor ID (bit 16-23)	Slave vendor ID (bit 16-23)
11	Slave vendor ID (bit 24-31)	Slave vendor ID (bit 24-31)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

Table 10 – S_CONNECT_CONF command SPDU (3rd SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_CONNECT_CONF = 0x02	S_CONNECT_CONF = 0x02
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Slave device code (bit 0-7)	Slave device code (bit 0-7)
9	Slave device code (bit 8-15)	Slave device code (bit 8-15)
10	Slave device code (bit 16-23)	Slave device code (bit 16-23)
11	Slave device code (bit 24-31)	Slave device code (bit 24-31)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

6.1.3 S_PRM_SET

S_PRM_SET is used by the safety master to send the parameters to the safety slave. The data of this command is shown in Table 11. If the SPDU length is too short for the complete data set, the safety master shall fragment the data into SPDU size and transmit the fragmented SPDUs.

Table 11 – S_PRM_SET command data

byte	Name	Description
0	Communication parameter size (bit 0-7)	The safety master sets the size of communication parameter. The safety slave sets the value received from the safety master.
1	Communication parameter size (bit 8-15)	
2	Reserved = 0x00	
3	Reserved = 0x00	
4	Watchdog time (bit 0-7)	The safety master sets the watchdog time registered in the configuration parameter. The safety slave sets the value received from the safety master.
5	Watchdog time (bit 8-15)	
6	Watchdog time (bit 16-23)	
7	Watchdog time (bit 24-31)	
8	Output user data length (bit 0-7)	The safety master sets the output user data length calculated in the following formula. (output user data length) = {(output data length registered in the configuration parameter) − (non-safety output user data length)} / 2 − 12 The safety slave sets the value received from the safety master.
9	Output user data length (bit 8-15)	
10	Input user data length (bit 0-7)	The safety master sets the input user data length calculated in the following formula. (input user data length) = {(input data length registered in the configuration parameter) − (non-safety input user data length)} / 2 − 12 The safety slave sets the value received from the safety master.
11	Input user data length (bit 8-15)	

Table 12, Table 13 and Table 14 show the example for which the SPDU size is 16 bytes. In this example, the data is fragmented into 3 SPDUs. The data is packed starting with the least significant byte and space at the end shall be loaded with 0x00 (see bytes 10 and 11 in the example).

Table 12 – S_PRM_SET command SPDU (1st SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_PRM_SET = 0x04	S_PRM_SET = 0x04
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Communication parameter size (bit 0-7)	Communication parameter size (bit 0-7)
9	Communication parameter size (bit 8-15)	Communication parameter size (bit 8-15)
10	Reserved = 0x00	Reserved = 0x00
11	Reserved = 0x00	Reserved = 0x00
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

Table 13 – S_PRM_SET command SPDU (2nd SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_PRM_SET = 0x04	S_PRM_SET = 0x04
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Watchdog time (bit 0-7)	Watchdog time (bit 0-7)
9	Watchdog time (bit 8-15)	Watchdog time (bit 8-15)
10	Watchdog time (bit 16-23)	Watchdog time (bit 16-23)
11	Watchdog time (bit 24-31)	Watchdog time (bit 24-31)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

Table 14 – S_PRM_SET command SPDU (3rd SPDU)

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_PRM_SET = 0x04	S_PRM_SET = 0x04
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Output user data length (bit 0-7)	Output user data length (bit 0-7)
9	Output user data length (bit 8-15)	Output user data length (bit 8-15)
10	Input user data length (bit 0-7)	Input user data length (bit 0-7)
11	Input user data length (bit 8-15)	Input user data length (bit 8-15)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

6.1.4 S_PRM_APPLY

The S_PRM_APPLY command is used by the safety master to send the parameter CRC to the safety slave and to cause the safety slave to apply the parameter.

The safety slave shall calculate the CRC for the received S_PRM_SET command and shall not apply the parameter if the calculated CRC and the parameter CRC of this command are not the same.

The data of this command is shown in Table 15. If the SPDU length is too short for the complete data set, the safety master shall fragment the data into SPDU size and transmit the fragmented SPDUs.

Table 15 – S_PRM_APPLY command data

byte	Name	Description
0	Parameter CRC (bit 0-7)	The polynomial of 0x90022004 and the seed of 0xFFFFFFFF is used for CRC calculation, and calculated from LSB in the following order. (1) Watchdog time (2) Output user data length (3) Input user data length The safety master calculates the CRC with the data used in the generation of S_PRM_SET command shown in Table 11. The safety slave calculates the CRC with the data in S_PRM_SET command received from the safety master shown in Table 11.
1	Parameter CRC (bit 8-15)	
2	Parameter CRC (bit 16-23)	
3	Parameter CRC (bit 24-31)	

Table 16 shows the example for which SPDU size is 16 bytes.

Table 16 – S_PRM_APPLY command SPDU

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_PRM_APPLY = 0x05	S_PRM_APPLY = 0x05
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Parameter CRC (bit 0-7)	Parameter CRC (bit 0-7)
9	Parameter CRC (bit 8-15)	Parameter CRC (bit 8-15)
10	Parameter CRC (bit 16-23)	Parameter CRC (bit 16-23)
11	Parameter CRC (bit 24-31)	Parameter CRC (bit 24-31)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

6.1.5 S_SAFE_DATA

The S_SAFE_DATA command is used by the safety master to send the safety output data to the safety slave, and by the safety slave to send the safety input data to the safety master while in the safety data sending/receiving state. Table 17 shows an example of the S_SAFE_DATA Command SPDU.

The safety output data is the safety data that is sent from the safety master to the safety slave. The safety input data is the safety data that is sent from the safety slave to the safety master.

Table 17 – S_SAFE_DATA command SPDU

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_SAFE_DATA = 0x06	S_SAFE_DATA = 0x06
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
7	Reserved = 0x00	Reserved = 0x00
8	Safety output data 1	Safety input data 1
9	Safety output data 2	Safety input data 2
10	Safety output data 3	Safety input data 3
11	Safety output data 4	Safety input data 4
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

6.1.6 S_DISCONNECT

The S_DISCONNECT command is sent by the safety master or the safety slave in order to disconnect the safety connection and reinitialize the connection parameters. A safety master reinitializes the connection parameters in the safety salve by transmitting this command.

Table 18 shows an example for which the SPDU size is 16 bytes. Table 19 shows the list of values used in the S_DISCONNECT Command SPDU.

Table 18 – S_DISCONNECT command SPDU

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_DISCONNECT (L) = 0x03	S_DISCONNECT (L) = 0x03
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7) = 0x00	Connection ID (bit 0-7) = 0x00
3	Connection ID (bit 8-15) = 0x00	Connection ID (bit 8-15) = 0x00
4	Master sequence number (bit 0-7) = 0x01	Slave sequence number (bit 0-7) = 0x01
5	Master sequence number (bit 8-15) = 0x00	Slave sequence number (bit 8-15) = 0x00
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	Factor (Refer to Table 19)	Factor (Refer to Table 19)
9	0x00	0x00
10	0x00	0x00
11	0x00	0x00
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

Table 19 – Factor in S_DISCONNECT command

Value	Symbol	Description
0x00	(Reserved)	For future use.
0x01	Request received	A DISCONNECT request was received from the application.
0x02	Request received from master	S_DISCONNECT command was received from the safety master.
0x03-0x0F	(Reserved)	For future use
0x10	Address mismatch	The slave address in the S_CONNECT_START, S_CONNECT_CONF does not match the own address.
0x11	Connection key mismatch	The master connection key in the S_CONNECT_START sent from a safety slave does not match the key which the safety master generated. This factor is stored only in the SPDU sent by a safety slave.
0x12	Connection ID mismatch	The connection ID in the S_CONNECT_CONF sent from a safety slave does not match the ID which the safety master generated. This factor is stored only in the SPDU sent by a safety slave.
0x13	Device information Mismatch	The device information (slave node address, slave vendor ID, slave device code) in the S_PRM_SET does not match the own device information.
0x14	Parameter mismatch	The safety parameter in the S_PRM_SET does not match the own safety parameter.
0x15	Parameter CRC mismatch	The parameter CRC in the S_PRM_APPLY does not match the parameter CRC calculated by own.
0x16-0x1F	(Reserved)	For future use.
0x20	Illegal command	The command of the received SPDU is illegal.
0x21	Illegal connection ID	The connection ID of the received SPDU is illegal.
0x22	Illegal sequence number	The sequence number of the received SPDU is illegal.
0x23	Illegal state	The state of the received SPDU is illegal.
0x24	CRC error	The CRC of the received SPDU does not match the calculation result.
0x25	Data mismatch	The data of SPDU received other than CRC does not match between the Block1 and the Block2.
0x26	Cross-check error	The result of the cross-check is NG.
0x27	Response timer error	A normal SPDU was not received within the response monitoring time.
0x28-0xFF	(Reserved)	For future use.

For the CRC calculation of this command, the initial value 0x0001 shall be used for the master sequence number and the slave sequence number, and the initial value 0x00000000 shall be used for the extended master sequence number and the extended slave sequence number.

6.1.7 S_FAIL_SAFE

Upon transition to the fail safe state, the safety master and the safety slave shall send the S_FAIL_SAFE command. The safety master and the safety slave shall not execute any processing for this command. The Connection ID, Master sequence number, Slave sequence number and State number keep the value from just before the transition to the fail safe state. Table 20 shows an example for an SPDU size of 16 bytes.

Table 20 – S_FAIL_SAFE command SPDU

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_FAIL_SAFE = 0x07	S_FAIL_SAFE = 0x07
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7)	Connection ID (bit 0-7)
3	Connection ID (bit 8-15)	Connection ID (bit 8-15)
4	Master sequence number (bit 0-7)	Slave sequence number (bit 0-7)
5	Master sequence number (bit 8-15)	Slave sequence number (bit 8-15)
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	0x00	0x00
9	0x00	0x00
10	0x00	0x00
11	0x00	0x00
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

6.1.8 S_NOP

The safety master and the safety slave shall not process this command. Table 21 shows an example for which SPDU size is 16 bytes.

Table 21 – S_NOP command SPDU

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
0	S_NOP = 0x00	S_NOP = 0x00
1	Reserved = 0x00	Reserved = 0x00
2	Connection ID (bit 0-7) = 0x00	Connection ID (bit 0-7) = 0x00
3	Connection ID (bit 8-15) = 0x00	Connection ID (bit 8-15) = 0x00
4	Master sequence number (bit 0-7) = 0x00	Slave sequence number (bit 0-7) = 0x00
5	Master sequence number (bit 8-15) = 0x00	Slave sequence number (bit 8-15) = 0x00
6	State number	State number
7	Reserved = 0x00	Reserved = 0x00
8	0x00	0x00
9	0x00	0x00
10	0x00	0x00
11	0x00	0x00

byte	Safety master SPDU (safety master → safety slave)	Safety slave SPDU (safety slave → safety master)
12	CRC (bit 0-7)	CRC (bit 0-7)
13	CRC (bit 8-15)	CRC (bit 8-15)
14	CRC (bit 16-23)	CRC (bit 16-23)
15	CRC (bit 24-31)	CRC (bit 24-31)

7 SCL protocol

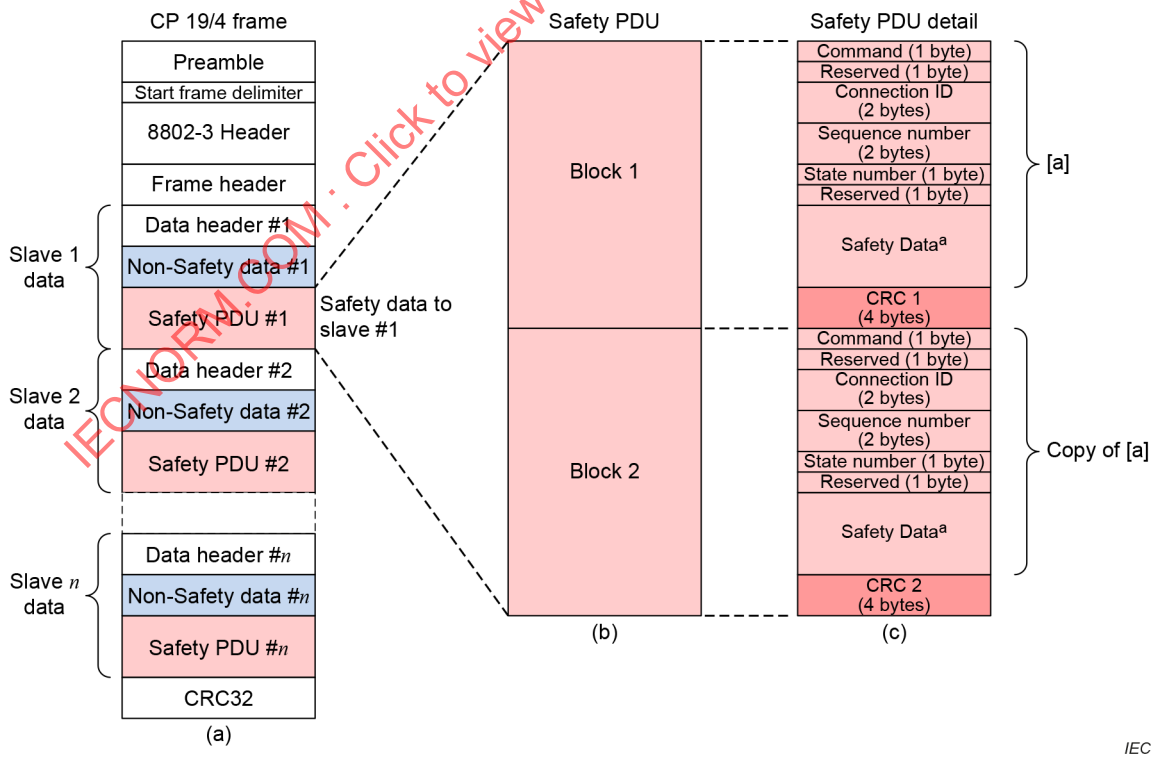
7.1 SPDU format

7.1.1 SPDU structure

Figure 11 a) shows an example of the CP 19/4 PDU with the FSCP 19 SPDUs in the payload. The length of the CP 19/4 PDU is fixed but the length of the non-safety data can be configured as a percentage of the SPDU length. A value of zero percent is permitted.

Figure 11 b), c) shows the SPDU format. An SPDU consists of two blocks (Block 1 and Block 2) and the data for the two blocks shall be the same except for the CRC value. CRC generator polynomials shall be unique among: Block 1, Block 2, and the CP 19/4 PDU (see 7.1.6 for a description of CRC generator polynomials).

The SPDU size, in bytes, is configurable in increments of 8 bytes, in the range of 32 to 1488. Therefore, each block size, in byte, is configurable in increment of 4 bytes, in the range of 16 to 744.



[a] The size of the safety data is limited by the FAL. CP19/4 uses ISO/IEC/IEEE 8802-3 which limits the safety data to 732 bytes.

Figure 11 – Safety PDU format

The relation between the user data size to be accepted by the service access point of the FAL and the maximum size of the safety user data in bytes is calculated by Formula (4).

$$\text{max\_safety\_user\_data\_length} = \frac{\text{user\_data\_length\_accpeted\_by\_FAL}}{2} - 12 \quad (4)$$

7.1.2 Connection ID

The connection ID is used for authentication by the safety master and the safety slave. When establishing the safety connection, the safety master and the safety slave exchange each other's connection key. The safety master generates the connection ID from the connection keys, and confirms this ID is unique. After this confirmation, the safety master sends the connection ID to the safety slave. Refer to 7.3.1.2 for details.

7.1.3 Sequence number

When an SPDU is sent, the safety master and the safety slave add a different sequence number to the SPDU. Refer to 5.3.2 for details.

7.1.4 Command

Table 22 shows the list of commands and their respective command codes. Refer to 6.1 for details.

Table 22 – List of commands

Code	Command	Description
0x0000	S_NOP	No operation. The safety master and the safety slaves perform no operation upon receipt of this command.
0x0001	S_CONNECT_START	Request to start connection establishment. The safety master sends the master connection key to the safety slave. And the safety slave replies with the slave connection key to the safety master. The safety master generates the connection ID from the master connection key and the slave connection key.
0x0002	S_CONNECT_CONF	Request to confirm connection establishment. If the connection ID generated by the safety master is valid, the safety master sends this command to the safety slave and establishes a connection.
0x0003	S_DISCONNECT	Request to disconnect connection. Disconnect the established connection.
0x0004	S_PRM_SET	Request to set parameters. The safety master sends communication parameters and application parameters to the safety slave.
0x0005	S_PRM_APPLY	Request to apply parameters. Apply parameters sent by S_PRM_SET command from the safety master to the safety slave.
0x0006	S_SAFE_DATA	Request to send safety data. The safety master sends safety output data to the safety slave. And the safety slave that receives it sends safety input data to the safety master.
0x0007	S_FAIL_SAFE	Notification of the fail safe state. This command is a notification that an SCL has transitioned to the fail safe state. The safety master and the safety slaves perform no operation in response to receiving this command.

7.1.5 State number

This number indicates the current state of the SCL. In the fail safe state, the State Number keeps the value from just before the transition to the fail safe state. See 7.2 for details of the state number.

7.1.6 CRC

There is a 32-bit CRC for each block. Refer to 5.3.5 for details.

7.1.7 Redundant data

When transmitting an SPDU, the complete data set (except for the CRC) is duplicated for redundancy. Each copy of the data is verified with a different CRC in the SPDU.

Refer to 5.3.6.1 and 5.3.6.2 for details.

7.2 Safety FAL service protocol machine

7.2.1 State transition of safety master

7.2.1.1 Safety communication layer

For the SCL in the safety master: the state transition diagram is shown in Figure 12, the states are described in Table 23, and the state transition matrix is shown in Table 24.

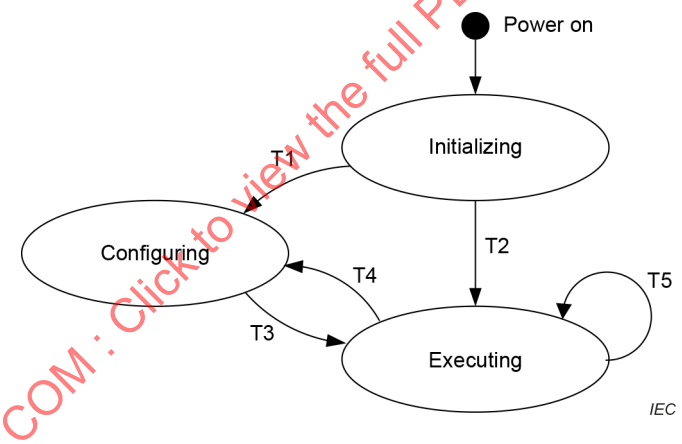


Figure 12 – Safety master SCL – state transition diagram

Table 23 – Safety master SCL – state description

State	Description
Initializing	Initial processing is executing.
Configuring	Configuration is executing. After configuration is completed, the SCL transitions to Executing state.
Executing	Connections between safety slaves can be established. For safety connections: the state transition diagram is shown in Figure 13.

Table 24 – Safety master SCL − state transition matrix

State	Transition	Condition	Action	Destination state
Initializing	T1	Initializing completed. && Configuration request received.	Initialize the variables related to the connection.	Configuring
Initializing	T2	Initializing completed. && Configuration request NOT received.	Initialize the variables related to the connection.	Executing
Configuring	T3	Configuration completed.	-	Executing
Executing	T4	Configuration started. && All modules are in Idle state.	-	Configuring
Executing	T5	Configuration started. && Some modules are NOT in Idle state.	-	Executing

7.2.1.2 Safety connection

For the safety connection in the safety master: the state transition diagram is shown in Figure 13, the states are described in Table 25, and the state transition matrix is shown in Table 26.

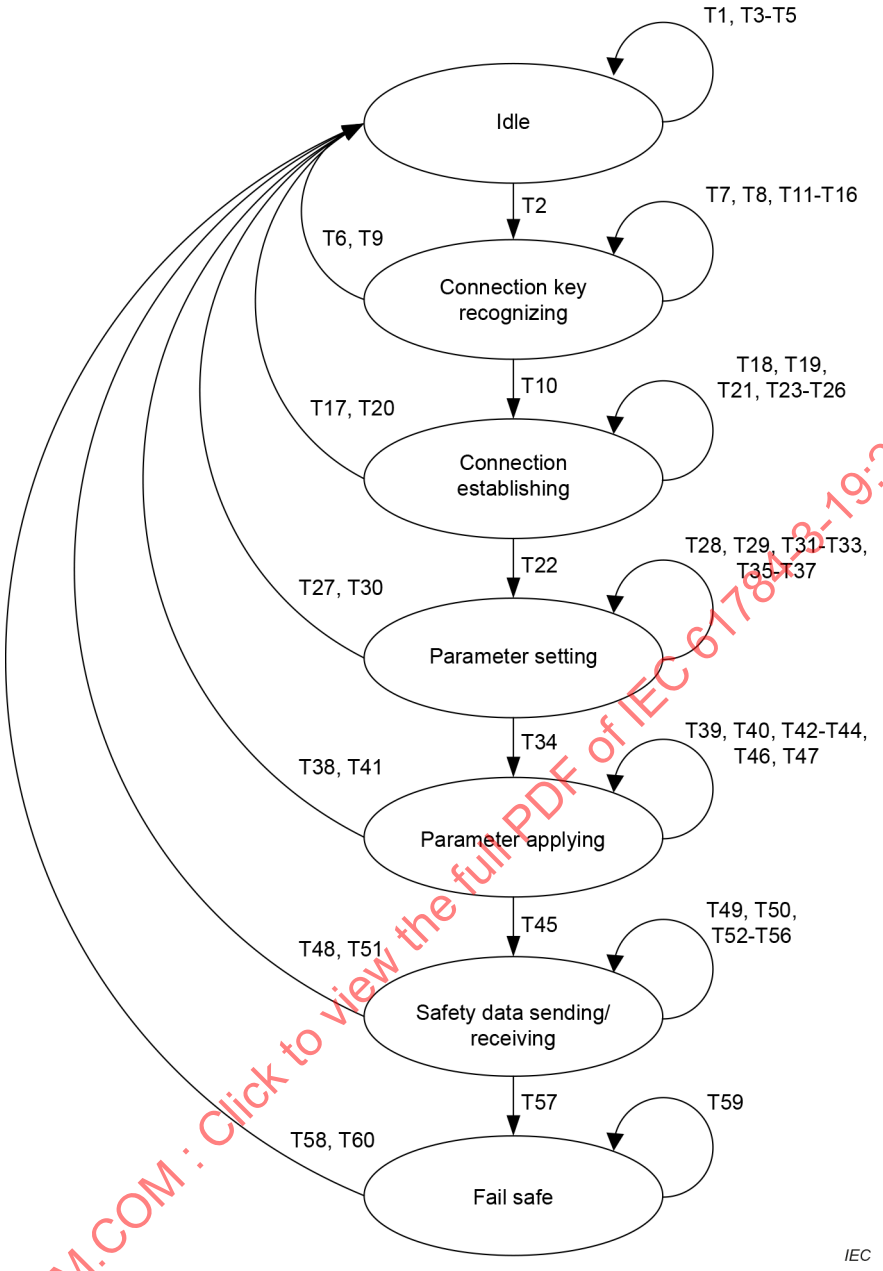


Figure 13 – Safety master safety connection − state transition diagram

Table 25 – Safety master safety connection − state description

State	Description
Idle	In this state, safety communication is not executed and there are no safety connections. A connection instance is waiting for the request to start/restart safety communication from user. Gate to protect unintended automatic recovery and to ensure the safe "start" in the case of normal operation or initial setting completion, or "restart" after the fault detection and/or removal of fault(s) checked by the user.
Connection key recognizing	Connection instance exchanges the connection key with the safety slave.
Connection establishing	Connection instance establishes the connection with the safety slave.
Parameter setting	Connection instance sets the slave parameter to the safety slave.
Parameter applying	Connection instance requests application of the slave parameter to the safety slave.
Safety data sending/receiving	Connection instance sends and receives the safety data with the safety slave.
Fail safe	Connection instance sends the S_FAIL_SAFE command. The S_DISCONNECT command from the safety slave or user intervention shall be done to transit to Idle state.
NOTE The safety master has these state for each connection with safety slaves individually.	

Table 26 – Safety master safety connection − state transition matrix

State	Transition	Condition	Action	Destination state
Idle	T1	Reset request received.	Initialize the variables related to the connection.	Idle
Idle	T2	Start safety communication request received.	Initialize the variables related to the connection. Send S_CONNECT_START command. Start response timer.	Connection key recognizing
Idle	T3	Stop safety communication request received.	Send S_DISCONNECT command. Start response timer.	Idle
Idle	T4	S_DISCONNECT command received.	Initialize the variables related to the connection. Stop response timer.	Idle
Idle	T5	Response timer expired.	Send S_DISCONNECT command. Start response timer.	Idle
Connection key recognizing	T6	Reset request received.	Initialize the variables related to the connection.	Idle
Connection key recognizing	T7	Stop safety communication request received.	Send S_DISCONNECT command. Start response timer.	Connection key recognizing

State	Transition	Condition	Action	Destination state
Connection key recognizing	T8	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Connection key recognizing
Connection key recognizing	T9	S_DISCONNECT command received.	Initialize the variables related to the connection. Stop response timer.	Idle
Connection key recognizing	T10	S_CONNECT_START command received. && Generated connection ID is NOT used.	Send S_CONNECT_CONF command. Start response timer.	Connection establishing
Connection key recognizing	T11	S_CONNECT_START command received. && Generated connection ID is already used.	Send S_CONNECT_START command. Start response timer.	Connection key recognizing
Connection key recognizing	T12	S_CONECT_CONF command received.	Notify error to application.	Connection key recognizing
Connection key recognizing	T13	S_PRM_SET command received.	Notify error to application.	Connection key recognizing
Connection key recognizing	T14	S_PRM_APPLY command received.	Notify error to application.	Connection key recognizing
Connection key recognizing	T15	S_SAFE_DATA command received.	Notify error to application.	Connection key recognizing
Connection key recognizing	T16	Response timer expired.	Send S_DISCONNECT command. Start response timer.	Connection key recognizing
Connection establishing	T17	Reset request received.	Initialize the variables related to the connection.	Idle
Connection establishing	T18	Stop safety communication request received.	Send S_DISCONNECT command. Start response timer.	Connection establishing
Connection establishing	T19	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Connection establishing
Connection establishing	T20	S_DISCONNECT command received.	Initialize the variables related to the connection. Stop response timer.	Idle

State	Transition	Condition	Action	Destination state
Connection establishing	T21	S_CONNECT_START command received.	Notify error to application.	Connection establishing
Connection establishing	T22	S_CONECT_CONF command received.	Send S_PRM_SET command. Start response timer.	Parameter setting
Connection establishing	T23	S_PRM_SET command received.	Notify error to application.	Connection establishing
Connection establishing	T24	S_PRM_APPLY command received.	Notify error to application.	Connection establishing
Connection establishing	T25	S_SAFE_DATA command received.	Notify error to application.	Connection establishing
Connection establishing	T26	Response timer expired.	Send S_DISCONNECT command. Start response timer.	Connection establishing
Parameter setting	T27	Reset request received.	Initialize the variables related to the connection.	Idle
Parameter setting	T28	Stop safety communication request received.	Send S_DISCONNECT command. Start response timer.	Parameter setting
Parameter setting	T29	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Parameter setting
Parameter setting	T30	S_DISCONNECT command received.	Initialize the variables related to the connection. Stop response timer.	Idle
Parameter setting	T31	S_CONNECT_START command received.	Notify error to application.	Parameter setting
Parameter setting	T32	S_CONECT_CONF command received.	Notify error to application.	Parameter setting
Parameter setting	T33	S_PRM_SET command received. && All parameter data has NOT been transmitted.	Send S_PRM_SET command. Start response timer.	Parameter setting
Parameter setting	T34	S_PRM_SET command received. && All parameter data has been transmitted.	Send S_PRM_APPLY command. Start response timer.	Parameter applying
Parameter setting	T35	S_PRM_APPLY command received.	Notify error to application.	Parameter setting
Parameter setting	T36	S_SAFE_DATA command received.	Notify error to application.	Parameter setting

State	Transition	Condition	Action	Destination state
Parameter setting	T37	Response timer expired.	Send S_DISCONNECT command. Start response timer.	Parameter setting
Parameter applying	T38	Reset request received.	Initialize the variables related to the connection.	Idle
Parameter applying	T39	Stop safety communication request received.	Send S_DISCONNECT command. Start response timer.	Parameter applying
Parameter applying	T40	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Parameter applying
Parameter applying	T41	S_DISCONNECT command received.	Initialize the variables related to the connection. Stop response timer.	Idle
Parameter applying	T42	S_CONNECT_START command received.	Notify error to application.	Parameter applying
Parameter applying	T43	S_CONECT_CONF command received.	Notify error to application.	Parameter applying
Parameter applying	T44	S_PRM_SET command received.	Notify error to application.	Parameter applying
Parameter applying	T45	S_PRM_APPLY command received.	Send S_SAFE_DATA command. Start watchdog timer.	Safety data sending/ receiving
Parameter applying	T46	S_SAFE_DATA command received.	Notify error to application.	Parameter applying
Parameter applying	T47	Response timer expired.	Send S_DISCONNECT command. Start response timer.	Parameter applying
Safety data sending/ receiving	T48	Reset request received.	Initialize the variables related to the connection.	Idle
Safety data sending/ receiving	T49	Stop safety communication request received.	Stop watchdog timer. Send S_DISCONNECT command. Start response timer.	Safety data sending/ receiving
Safety data sending/ receiving	T50	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Safety data sending/ receiving

State	Transition	Condition	Action	Destination state
Safety data sending/ receiving	T51	S_DISCONNECT command received.	Stop watchdog timer. Notify error to application. Initialize the variables related to the connection.	Idle
Safety data sending/ receiving	T52	S_CONNECT_START command received.	Notify error to application.	Safety data sending/ receiving
Safety data sending/ receiving	T53	S_CONECT_CONF command received.	Notify error to application.	Safety data sending/ receiving
Safety data sending/ receiving	T54	S_PRM_SET command received.	Notify error to application.	Safety data sending/ receiving
Safety data sending/ receiving	T55	S_PRM_APPLY command received.	Notify error to application.	Safety data sending/ receiving
Safety data sending/ receiving	T56	S_SAFE_DATA command received.	Stop watchdog timer. Deliver safety input data to application. Send S_SAFE_DATA command. Start watchdog timer.	Safety data sending/ receiving
Safety data sending/ receiving	T57	Watchdog timer expired.	Stop watchdog timer. Notify error to application. Send S_FAIL_SAFE command.	Fail safe
Fail safe	T58	Reset request received.	Initialize the variables related to the connection.	Idle
Fail safe	T59	Stop safety communication request received.	Send S_DISCONNECT command. Start response timer.	Fail safe
Fail safe	T60	S_DISCONNECT command received.	Initialize the variables related to the connection.	Idle

7.2.2 State transition of safety slave

7.2.2.1 Safety communication layer

For the SCL in the safety slave: the state transition diagram is shown in Figure 14, the states are described in Table 27, and the state transition matrix is shown in Table 28.

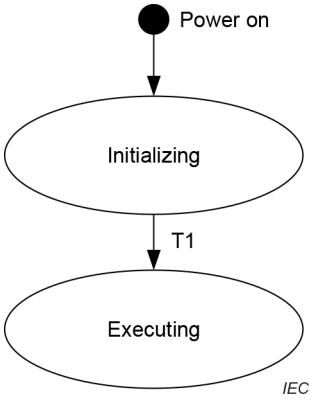


Figure 14 – Safety slave SCL − state transition diagram

Table 27 – Safety slave SCL − state description

State	Description
Initializing	Initial processing is executing.
Executing	Connections between safety masters can be established. For the safety connection: the state transition diagram is shown in Figure 15

Table 28 – Safety slave SCL – state transition matrix

State	Transition	Condition	Action	Destination state
Initializing	T1	Initializing completed.	Initialize the variables related to the connection.	Executing

7.2.2.2 Safety connection

For the safety connection in the safety slave: the state transition diagram is shown in Figure 15, the states are described in Table 29, and the state transition matrix is shown in Table 30.

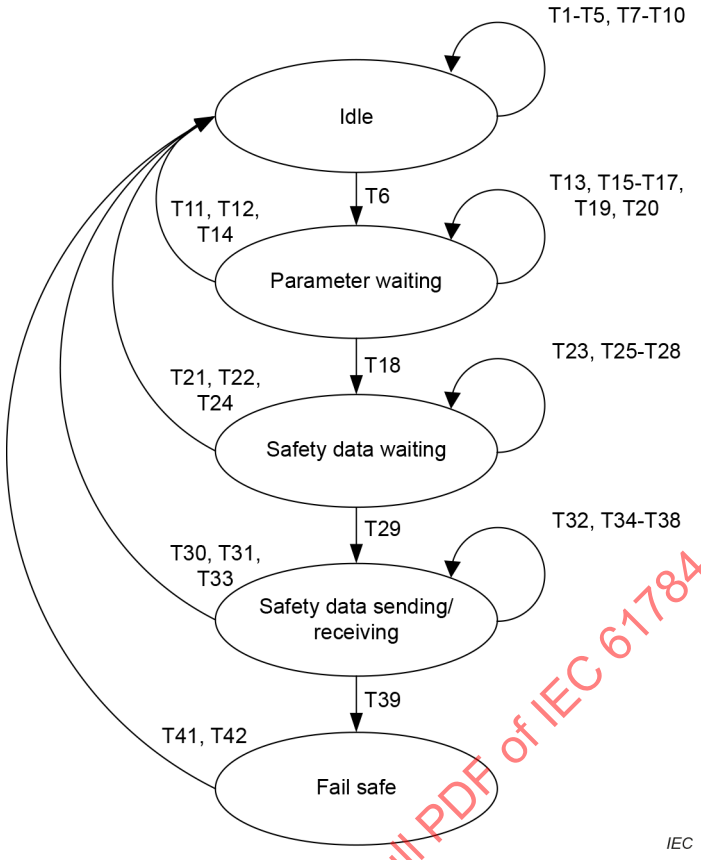


Figure 15 – Safety slave safety connection − state transition diagram

Table 29 – Safety slave safety connection − state description

State	Description
Idle	Connection instance is waiting for the connection establishment from the safety master.
Parameter waiting	Connection instance is waiting for the parameter setting from the safety master.
Safety data waiting	Connection instance is waiting for a reception of the first safety data from the safety master.
Safety data sending/receiving	Connection instance sends and receives the safety data with the safety master.
Fail safe	Connection instance sends the S_FAIL_SAFE command; The S_DISCONNECT command from the safety slave or user intervention shall be done to transit to Idle state.
NOTE The safety slave has these state for each connection with safety masters individually.	

Table 30 – Safety slave safety connection − state transition matrix

State	Transition	Condition	Action	Destination state
Idle	T1	Reset request received.	Initialize the variables related to the connection.	Idle
Idle	T2	Stop safety communication request received.	Initialize the variables related to the connection. Send S_DISCONNECT command.	Idle
Idle	T3	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Idle
Idle	T4	S_DISCONNECT command received.	Initialize the variables related to the connection. Notify error to application.	Idle
Idle	T5	S_CONNECT_START command received.	Save the received master connection key. Generate a slave connection key. Send S_CONNECT_START command.	Idle
Idle	T6	S_CONNECT_CONF command received. && Compare check of connection ID passed.	Save the connection ID. S_CONNECT_CONF command.	Parameter waiting
Idle	T7	S_CONNECT_CONF command received. && Compare check of connection ID NOT passed.	Initialize the variables related to the connection. Send S_DISCONNECT command.	Idle
Idle	T8	S_PRM_SET command received.	Notify error to application.	Idle
Idle	T9	S_PRM_APPLY command received.	Notify error to application.	Idle
Idle	T10	S_SAFE_DATA command received.	Notify error to application.	Idle
Parameter waiting	T11	Reset request received.	Initialize the variables related to the connection.	Idle
Parameter waiting	T12	Stop safety communication request received.	Initialize the variables related to the connection. Send S_DISCONNECT command.	Idle

State	Transition	Condition	Action	Destination state
Parameter waiting	T13	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Parameter waiting
Parameter waiting	T14	S_DISCONNECT command received.	Initialize the variables related to the connection. Notify error to application.	Idle
Parameter waiting	T15	S_CONNECT_START command received.	Notify error to application.	Parameter waiting
Parameter waiting	T16	S_CONNECT_CONF command received.	Notify error to application.	Parameter waiting
Parameter waiting	T17	S_PRM_SET command received.	Store the received parameter. Send S_PRM_SET command.	Parameter waiting
Parameter waiting	T18	S_PRM_APPLY command received. && Compare check of parameter CRC passed.	Apply the received parameter. Send S_PRM_APPLY command.	Safety data waiting
Parameter waiting	T19	S_PRM_APPLY command received. && Compare check of parameter CRC NOT passed.	Notify error to application.	Parameter waiting
Parameter waiting	T20	S_SAFE_DATA command received.	Notify error to application.	Parameter waiting
Safety data waiting	T21	Reset request received.	Initialize the variables related to the connection.	Idle
Safety data waiting	T22	Stop safety communication request received.	Initialize the variables related to the connection. Send S_DISCONNECT command.	Idle
Safety data waiting	T23	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application.	Safety data waiting
Safety data waiting	T24	S_DISCONNECT command received.	Initialize the variables related to the connection. Notify error to application.	Idle
Safety data waiting	T25	S_CONNECT_START command received.	Notify error to application.	Safety data waiting
Safety data waiting	T26	S_CONNECT_CONF command received.	Notify error to application.	Safety data waiting

State	Transition	Condition	Action	Destination state
Safety data waiting	T27	S_PRM_SET command received.	Notify error to application.	Safety data waiting
Safety data waiting	T28	S_PRM_APPLY command received.	Notify error to application.	Safety data waiting
Safety data waiting	T29	S_SAFE_DATA command received.	Deliver safety output data to application. Send S_SAFE_DATA command. Start watchdog timer.	Safety data sending/ receiving
Safety data sending/ receiving	T30	Reset request received.	Initialize the variables related to the connection.	Idle
Safety data sending/ receiving	T31	Stop safety communication request received.	Stop watchdog timer. Initialize the variables related to the connection. Send S_DISCONNECT command.	Idle
Safety data sending/ receiving	T32	Illegal SPDU received. Illegal connection ID. Illegal sequence number. Illegal CRC. Illegal command. Illegal data.	Notify error to application. Send S_SAFE_DATA command.	Safety data sending/ receiving
Safety data sending/ receiving	T33	S_DISCONNECT command received.	Stop watchdog timer. Notify error to application. Initialize the variables related to the connection.	Idle
Safety data sending/ receiving	T34	S_CONNECT_START command received.	Notify error to application. Send S_SAFE_DATA command.	Safety data sending/ receiving
Safety data sending/ receiving	T35	S_CONNECT_CONF command received.	Notify error to application. Send S_SAFE_DATA command.	Safety data sending/ receiving
Safety data sending/ receiving	T36	S_PRM_SET command received.	Notify error to application. Send S_SAFE_DATA command.	Safety data sending/ receiving
Safety data sending/ receiving	T37	S_PRM_APPLY command received.	Notify error to application. Send S_SAFE_DATA command.	Safety data sending/ receiving
Safety data sending/ receiving	T38	S_SAFE_DATA command received.	Stop watchdog timer. Deliver safety output data to application. Send S_SAFE_DATA command. Start watchdog timer.	Safety data sending/ receiving

State	Transition	Condition	Action	Destination state
Safety data sending/ receiving	T39	Watchdog timer expired.	Stop watchdog timer. Notify error to application. Send S_FAIL_SAFE command.	Fail safe
Fail safe	T40	Reset request received.	Initialize the variables related to the connection.	Idle
Fail safe	T41	Stop safety communication request received.	Initialize the variables related to the connection Send S_DISCONNECT command.	Idle
Fail safe	T42	S_DISCONNECT command received.	Initialize the variables related to the connection. Notify error to application.	Idle

7.3 Behaviour description

7.3.1 Connection establishment

7.3.1.1 General

Subclause 7.3.1 explains the procedure of the establishment of safety connections.

All SPDUs exchanged between safety master and safety slave are verified with CRC data integrity. The CCITT-16 generator polynomial shown in Formula (3) shall be used for these CRC calculations which are calculated from LSB to MSB.

7.3.1.2 Connection ID generation sequence

7.3.1.2.1 General

The connection ID shall be generated by the safety master and the safety connection shall be established using the following sequence:

1) the safety master sends the master connection key to the safety slave with a S_CONNECT_START command. The safety slave replies by sending the slave connection key to the safety master;
2) the safety master generates a connection ID (see 5.3.4) and sends it to the safety slave using the S_CONNECT_CONF command.

7.3.1.2.2 S_CONNECT_START command sent to safety slave

The safety master shall send the S_CONNECT_START command to the safety slave with the following data:

1) master connection key (the random number that the SCL in the safety master generated, see 7.3.1.2.8);
2) safety slave node address (one of the configuration parameters registered in the safety master);
3) slave connection key (set 0x0000).

7.3.1.2.3 S_CONNECT_START command sent to safety master

The safety slave upon receipt of a S_CONNECT_START command shall authenticate the slave node address in the SPDU, then reply by sending a S_CONNECT_START command back to the safety master with the following data:

1) master connection key (the value sent from the safety master);
2) safety slave node address;
3) slave connection key (the random number that the SCL in the safety slave generated).

7.3.1.2.4 Connection ID generated by safety master

Upon receipt of a S_CONNECT_START command from a safety slave, the safety master shall generate a connection ID value using the following sequence:

1) verify the value of master connection key matches what was sent to the safety slave;
2) generate a connection ID value;
3) verify the uniqueness of the connection ID value among all safety channels;
4) if the generated connection ID value is not unique, the safety master shall discard the connection keys and start the process over by sending a S_CONNECT_START command to the safety salve with a new master connection key.

7.3.1.2.5 S_CONNECT_CONF command sent to safety slave

Upon verification of the connection ID value, the safety master shall send a S_CONNECT_CONF command to the safety slave with the following data:

1) connection ID;
2) safety slave node address;
3) safety slave vendor ID;
4) safety slave device code.

NOTE The values for the safety slave node address, vendor ID, and device code are all found in the configuration parameter registered in the safety master.

7.3.1.2.6 Connection ID confirmed by safety slave

The safety slave upon receipt of a S_CONNECT_CONF command from the safety master with verified data integrity shall store the received connection ID and send a S_CONNECT_CONF command to the safety master.

7.3.1.2.7 Connection established by safety master

Upon receipt of a S_CONNECT_CONF command from the safety slave, the safety master shall verify the received connection ID and if verified, the safety connection is established and the safety master transitions to the next state (see 7.2.1).

7.3.1.2.8 Generation of master connection key and slave connection key

A random number generated in the SCL shall be used for the master connection key and the slave connection key. The size of the random number shall be 16 bits.

For random number generation, proper randomness shall be guaranteed by using algorithms such as Mersenne Twister, or the Xorshift. The Linear Congruential Generators shall not be used because their random numbers are not evenly distributed in multi-dimensional space.

To guarantee randomness at each power cycle, an initial value of seed for the random numbers shall be different from the previous power cycle. Moreover, to guarantee the uniqueness among all connections, the seed for the random numbers shall be different among all modules. (E.g. the module number is added to the seed.)

EXAMPLE Seed values can be generated by combining: a current time, an elapse time after production, and the number of power cycles stored in NVS.

7.3.1.3 Management of node address in safety slave

7.3.1.3.1 General

When starting up, a safety slave shall obtain its node address from the non-safety subsystem. With each receipt of a S_CONNECT_START command or a S_CONNECT_CONF command, the safety slave shall verify the node address.

7.3.1.3.2 Variables for node address and device information

The SCL of a safety slave shall maintain the variables for the node address and device information as shown in Table 31.

Table 31 – Safety salve node and device variables

Variable	Description
Node address (Node_Address)	16 bits

Device information (Device_Info)	{
	Vendor ID (Vendor_ID): 32 bits
	Device code (Device_Code): 32 bits
	}

7.3.1.3.3 Processing of node address and device information

The node address and device information shall be processed as follows:

- the initial value of Node_Address and the initial value of each member of Device_Info shall be 0xFFFF;
- when starting up, the node address obtained from the non-safety subsystem shall be copied to Node_Address (see Figure 16);
- device information obtained from the non-safety subsystem shall be copied to each member of Device_Info (see Figure 16);
- when the S_CONNECT_START command is received, the slave node address in the SPDU shall be compared to the Node_Address (see Figure 17);
- when the S_CONNECT_CONF command is received, the slave node address in the SPDU shall be compared to the Node_Address and the slave device information in the SPDU shall be compared to each member of the Device_Info (see Figure 18).

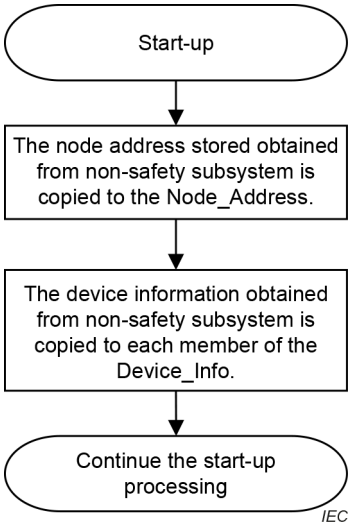


Figure 16 – Node address and device information processing flow at start-up

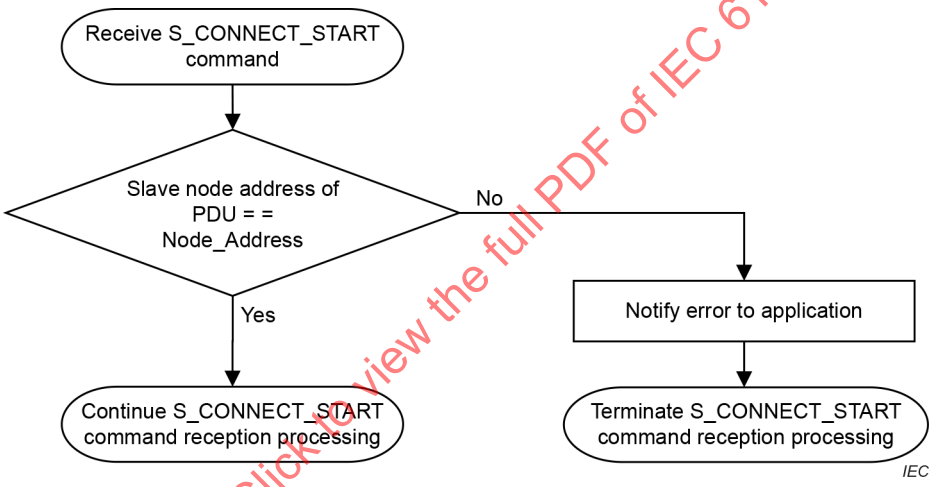


Figure 17 – S_CONNECT_START command reception processing flow

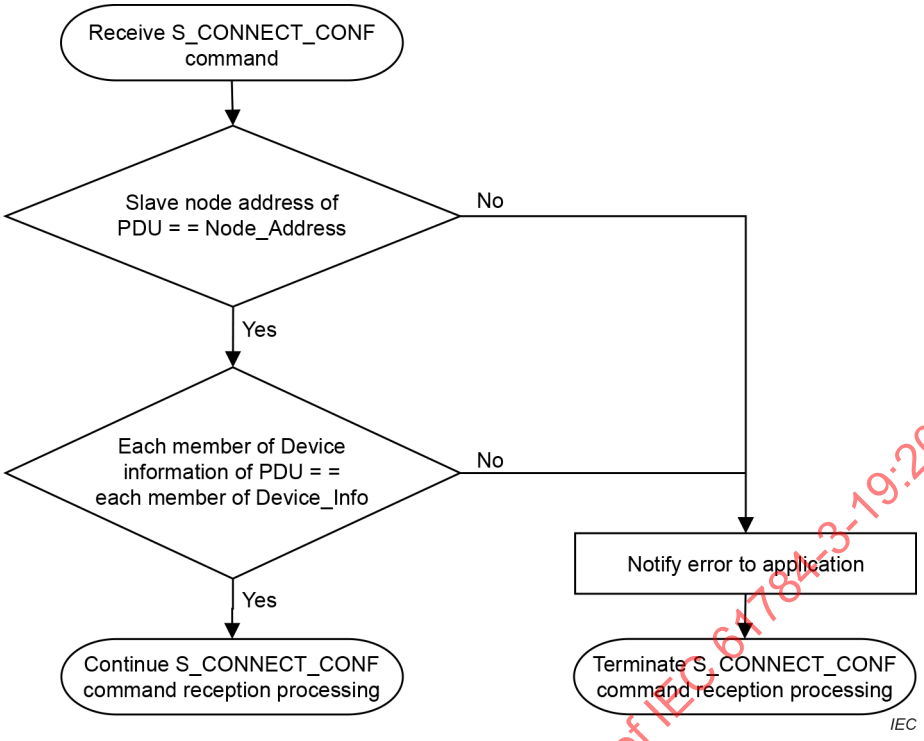


Figure 18 – S_CONNECT_CONF command reception processing flow

7.3.1.4 Sequence from connection establishment to safety data transmission/reception

Figure 19 and Figure 20 show the sequence that the safety master and the safety slave shall use to establish a safety connection and start communication as follows:

1) the safety master shall send a S_CONNECT_START command containing the master connection key to the safety slave;
2) the safety slave, upon receipt of a S_CONNECT_START command, shall generate a slave connection key and send a S_CONNECT_START command response containing this key to the safety master;
3) the safety master, upon receipt of the S_CONNECT_START command response, shall generate a connection ID, and if verified unique, the safety master shall send a S_CONNECT_CONF command containing the connection ID to the safety slave, but if the connection ID is not unique, the safety master shall send a new S_CONNECT_START command to the safety slave;
4) the safety slave, upon receipt of the S_CONNECT_CONF command, shall store the connection ID and send a S_CONNECT_CONF command response to the safety master;
5) upon receipt of the S_CONNECT_CONF command response by the safety master, the safety connection is established;
6) the safety master shall send a S_PRM_SET command with the slave parameter to the safety slave (the S_PRM_SET command shall be fragmented across multiple SPDUs if the slave parameter is too long for a single SPDU);
7) the safety slave, upon receipt of a S_PRM_SET command, shall store the slave parameter and send a S_PRM_SET command response to the safety master;
8) the safety master, upon receipt of the S_PRM_SET command response, shall send a S_PRM_APPLY command to request the safety slave to apply the slave parameter;
9) the safety slave, upon receipt of a S_PRM_APPLY command, shall apply the slave parameter and send a S_PRM_APPLY command response to the safety master;

10) the safety master, upon receipt of the S_PRM_APPLY command response, shall send a S_SAFE_DATA command containing safety output data to the safety slave and start the watchdog timer;

11) the safety slave, upon receipt of a S_SAFE_DATA command, shall send a S_SAFE_DATA command containing safety input data to the safety master and start the watchdog timer.

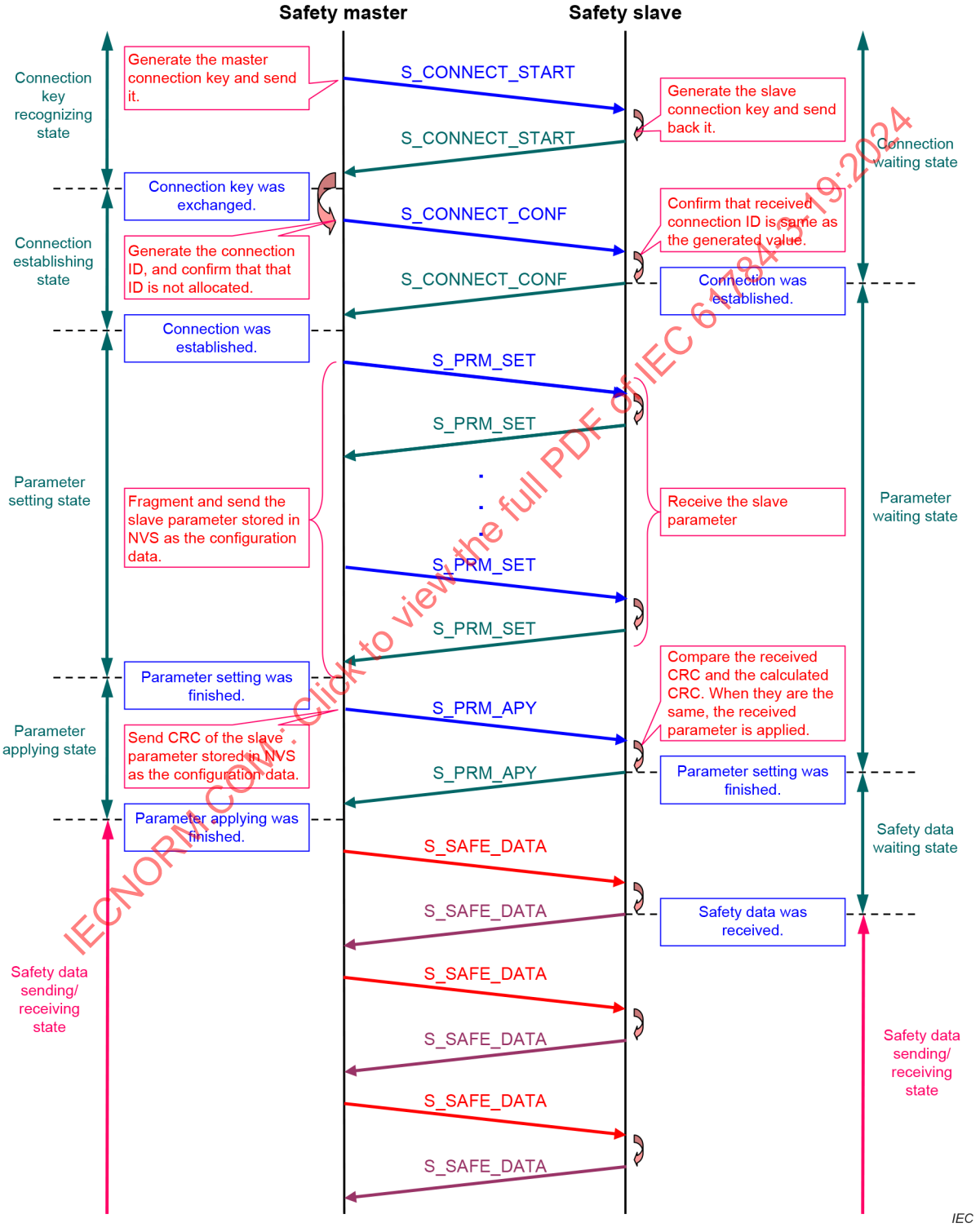


Figure 19 – Sequence example 1 from connection establishment to safety data transmission/reception

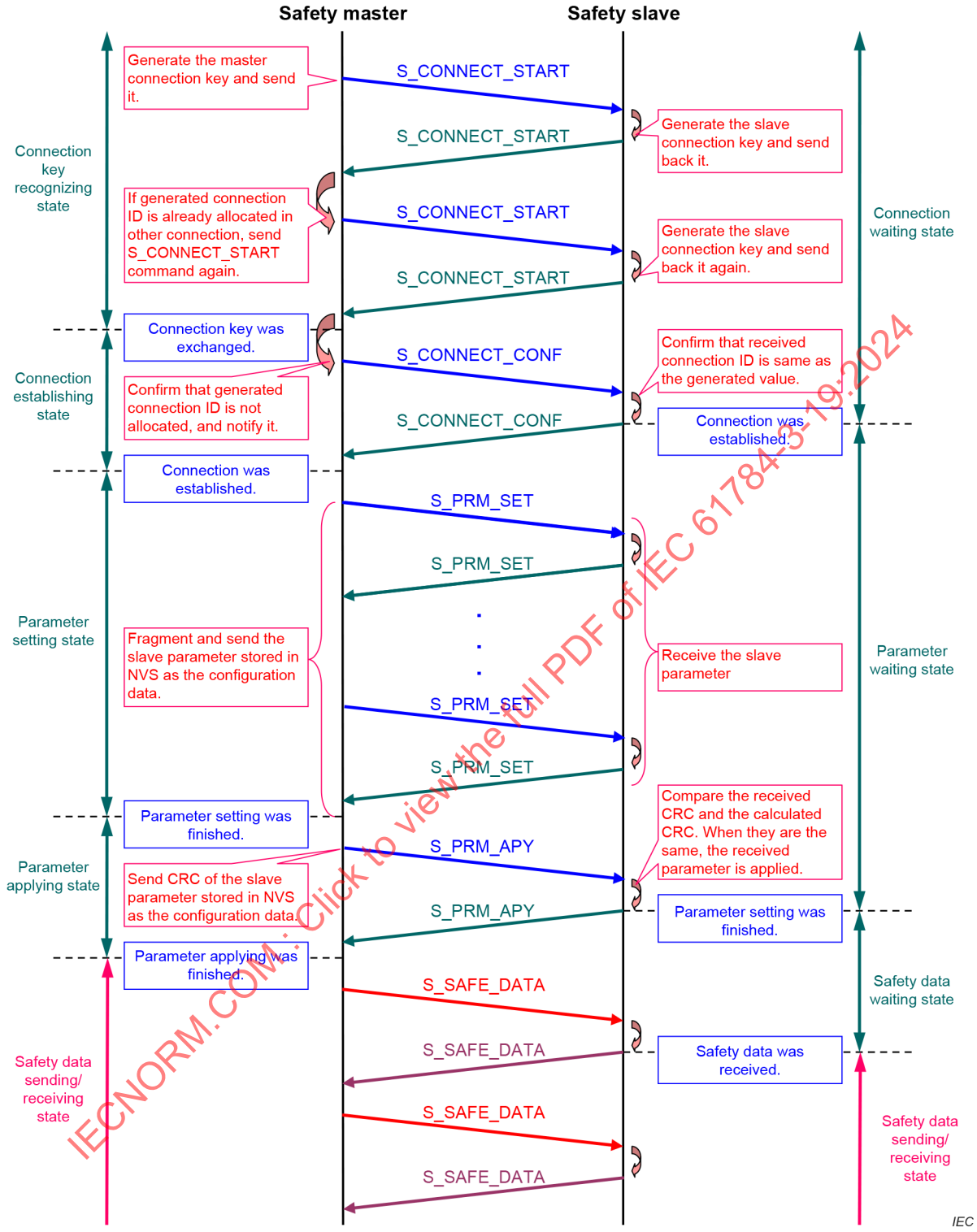


Figure 20 – Sequence example 2 from connection establishment to safety data transmission/reception

7.3.2 Safety data sending/receiving sequence

7.3.2.1 General

Subclause 7.3.2 explains the S_SAFE_DATA command sequence without error and with error.

7.3.2.2 Sequence without error

Figure 21 shows the sequence and the processing of S_SAFE_DATA without error in the safety master and the safety slave.

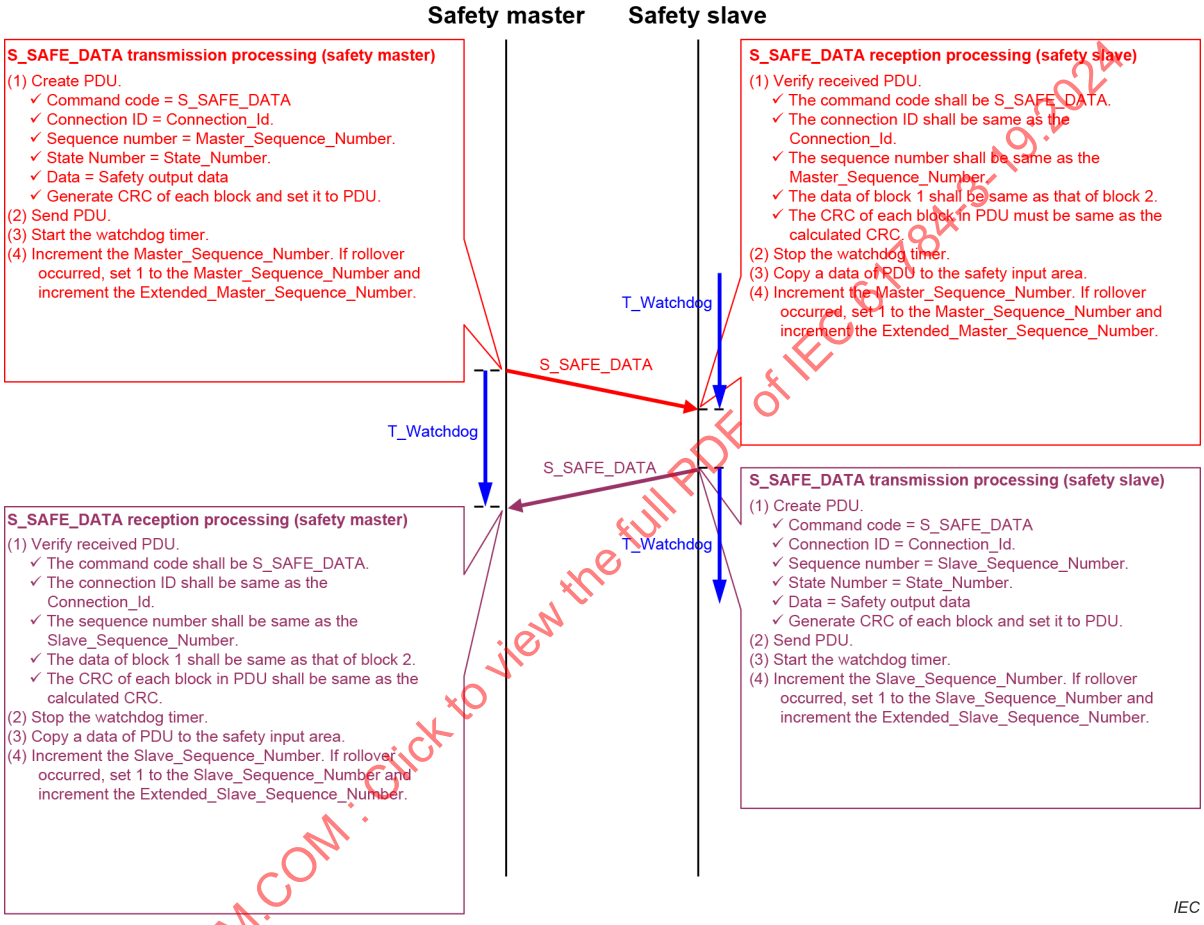


Figure 21 – S_SAFE_DATA command sequence

7.3.2.3 Loss or delay of message from safety master

Figure 22 shows the safety measure for loss of a S_SAFE_DATA command message from the safety master. Figure 23 shows the safety measure for delay of a S_SAFE_DATA command message from the safety master.

If a valid S_SAFE_DATA command is not received by a safety slave within the time expectation, the watchdog timer in the safety slave expires. As a result, the safety slave SCL transitions to the fail safe state, whereupon the safety communication is stopped and the safety output is set to its safe state.

When a valid S_SAFE_DATA command is not received by the safety master within the time expectation, the watchdog timer in the safety master expires. As a result, the safety master SCL transitions to the fail safe state, whereupon the safety communication is stopped.

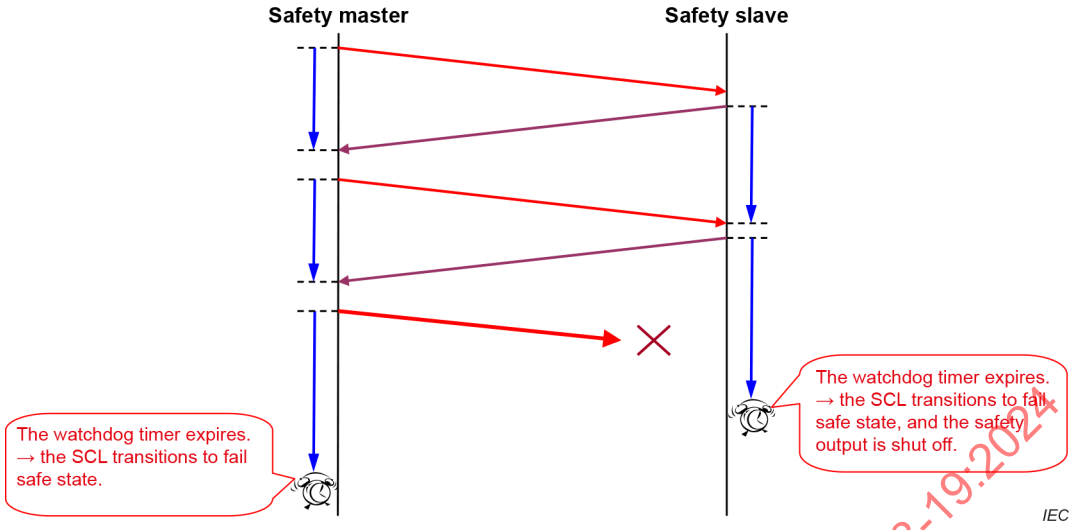


Figure 22 – Loss of S_SAFE_DATA command from safety master

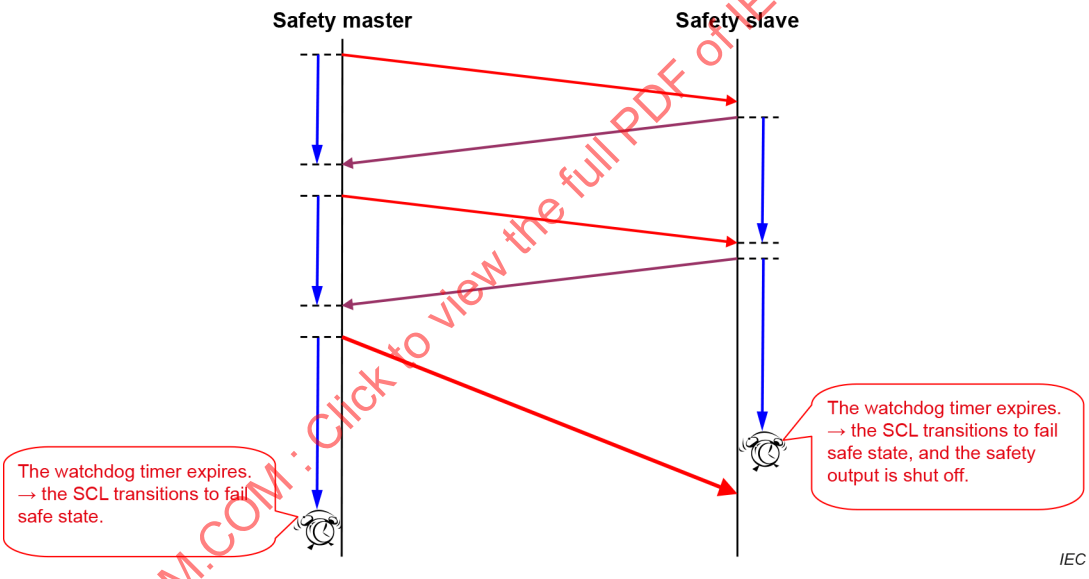


Figure 23 – Delay of S_SAFE_DATA command from safety master

7.3.2.4 Loss or delay of message from safety slave

Figure 24 shows the safety measure for loss of a S_SAFE_DATA command message from the safety slave. Figure 25 shows the safety measure for delay of a S_SAFE_DATA command message from the safety slave.

If a valid S_SAFE_DATA command is not received by a safety master within the time expectation, the watchdog timer in the safety master expires. As a result, the safety master SCL transitions to the fail safe state, whereupon the safety communication is stopped.

When a valid S_SAFE_DATA command is not received by the safety slave within the time expectation, the watchdog timer in the safety slave expires. As a result, the safety slave SCL transitions to the fail safe state, whereupon the safety communication is stopped and the safety output is set to its safe state.

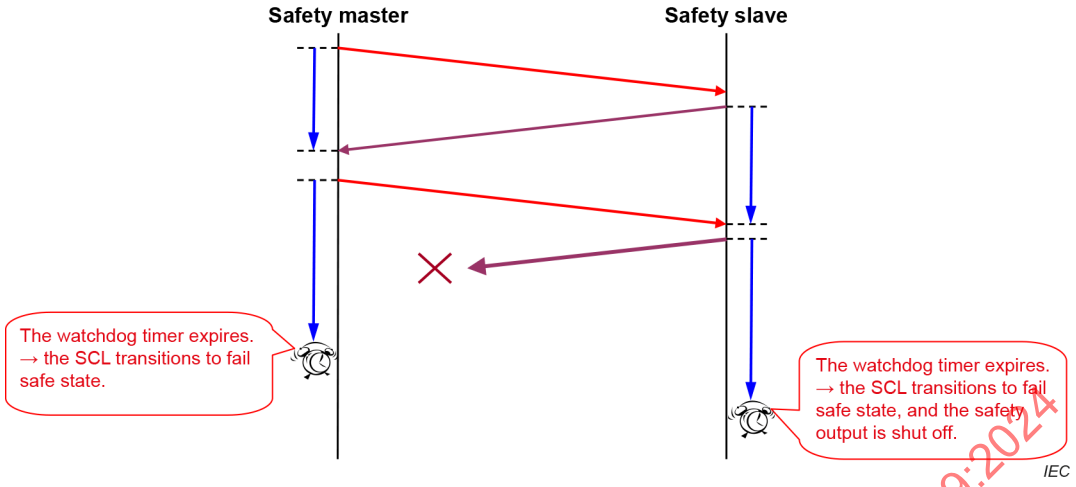


Figure 24 – Loss of S_SAFE_DATA command from safety slave

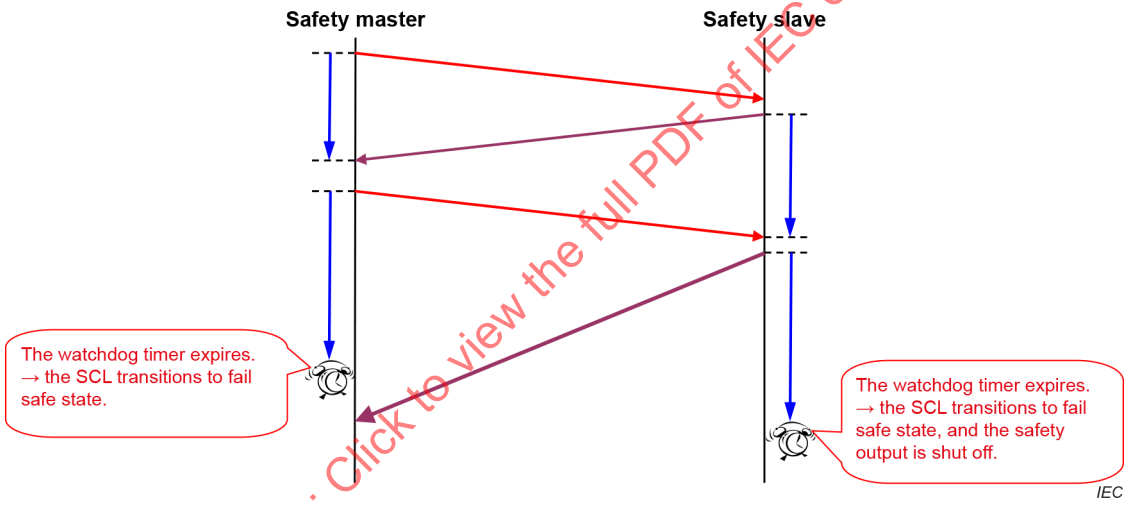


Figure 25 – Delay of S_SAFE_DATA command from safety slave

7.3.2.5 Insertion of message to safety slave

Figure 26 shows the safety measure for S_SAFE_DATA command message insertion in a safety slave.

Misdirected and out-of-sequence S_SAFE_DATA command messages are detected in the safety slave SCL by verifying the connection ID and the sequence number. Invalid messages are discarded.

If a valid S_SAFE_DATA command is not received by a safety slave within the time expectation, the watchdog timer in the safety slave expires. As a result, the safety slave SCL transitions to the fail safe state, whereupon the safety communication is stopped and the safety output is set to its safe state.

When a valid S_SAFE_DATA command is not received by the safety master within the time expectation, the watchdog timer in the safety master expires. As a result, the safety master SCL transitions to the fail safe state, whereupon the safety communication is stopped.

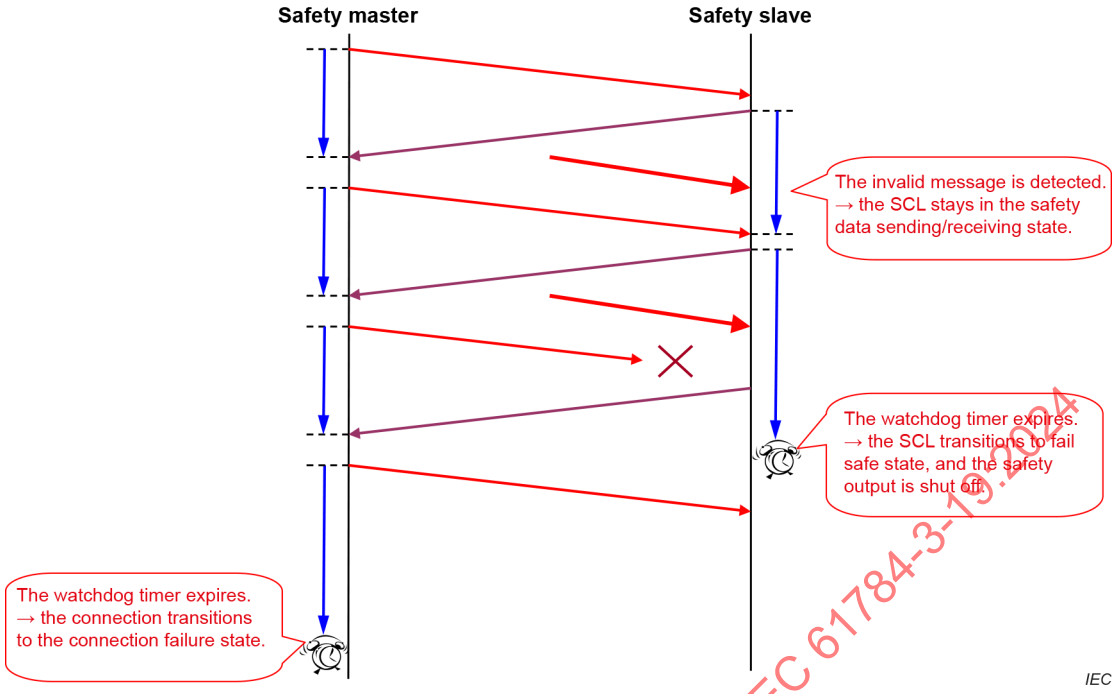


Figure 26 – Insertion of message to safety slave

7.3.2.6 Insertion of message to safety master

Figure 27 shows the safety measure for S_SAFE_DATA command message insertion in a safety master.

Misdirected and out-of-sequence S_SAFE_DATA command messages are detected in the safety master SCL by verifying the connection ID and the sequence number. Invalid messages are discarded.

If a valid S_SAFE_DATA command is not received by a safety master within the time expectation, the watchdog timer in the safety master expires. As a result, the safety master SCL transitions to the fail safe state, whereupon the safety communication is stopped.

When a valid S_SAFE_DATA command is not received by the safety slave within the time expectation, the watchdog timer in the safety slave expires. As a result, the safety slave SCL transitions to the fail safe state, whereupon the safety communication is stopped and the safety output is set to its safe state.

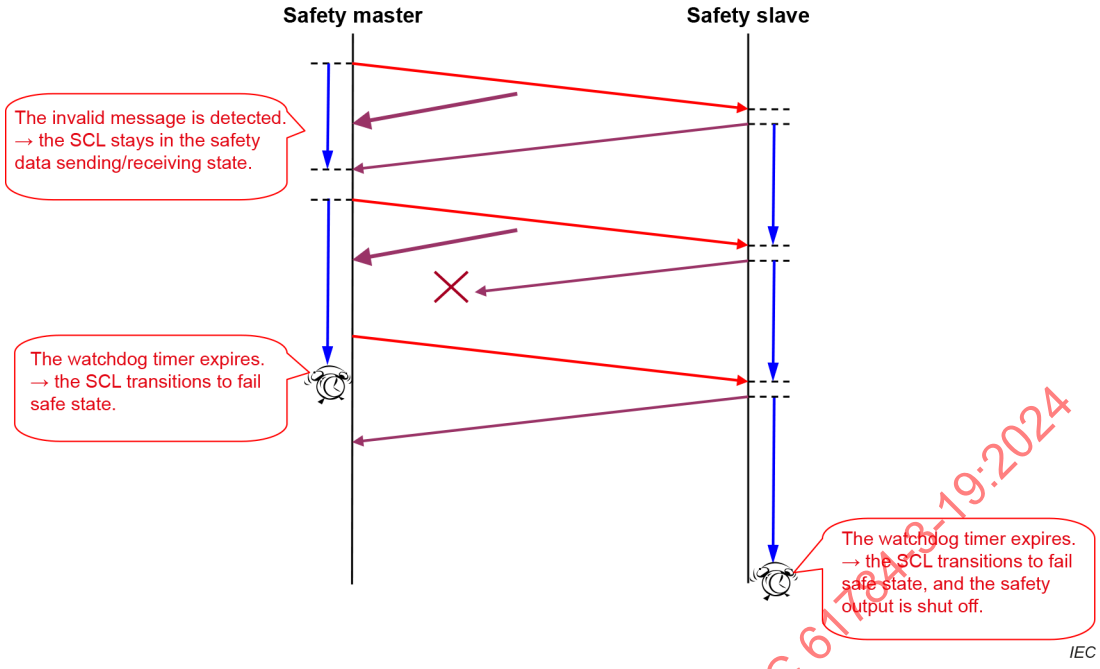


Figure 27 – Insertion of message to safety master

7.3.3 Disconnect safety channel

7.3.3.1 General

Subclause 7.3.3 explains disconnection of the safety channel in the safety master and the safety slave.

7.3.3.2 Disconnect safety channel in safety master

A safety master shall reinitialize the safety connection state and send a S_DISCONNECT command to the safety slave upon detection of:

- response timer expired in: idle state, connection key recognizing state, connection establishing state, parameter setting state, or parameter applying state;
- stop safety communication requested from application in: idle state, connection key recognizing state, connection establishing state, parameter setting state, parameter applying state, safety data sending/receiving state or fail safe state.

A safety slave that receives a S_DISCONNECT command shall reinitialize the safety connection state and configuration data, and send a S_DISCONNECT command to the safety master.

7.3.3.3 Disconnect safety channel in safety slave

A safety slave shall reinitialize the connection state and configuration data, and send a S_DISCONNECT command to the safety master upon detection of:

- stop safety communication requested from application in: idle state, parameter waiting state, safety data waiting state, safety data sending/receiving state or fail safe state.

A safety master that receives a S_DISCONNECT command shall reinitialize the safety connection state.

8 SCL management

8.1 Parameter definitions

8.1.1 General

Subclause 8.1 describes the parameter variables used in this protocol. The comment in brackets in the title of each subclause shows the name of the variable. Table 32 shows the list of the parameter variables.

Table 32 – List of parameter variables

Variable name	Type of variable	Data size / Variable range	Reference
T_Watchdog	Unsigned32	32 bits / 1 to $2^{32}-1$	8.1.2
T_Response	Unsigned32	32 bits / 1 to $2^{32}-1$	8.1.3
Master_Connection_Key	Unsigned16	16 bits / 1 to $2^{16}-1$	8.1.4
Slave_Connection_Key	Unsigned16	16 bits / 1 to $2^{16}-1$	8.1.5
Connection_Id	Unsigned16	16 bits / 1 to $2^{16}-1$	8.1.6
Master_Sequence_Number	Unsigned16	16 bits / 1 to $2^{16}-1$	8.1.7
Extended_Master_Sequence_Number	Unsigned32	32 bits / 0 to $2^{32}-1$	8.1.8
Slave_Sequence_Number	Unsigned16	16 bits / 1 to $2^{16}-1$	8.1.9
Extended_Slave_Sequence_Number	Unsigned32	32 bits / 0 to $2^{32}-1$	8.1.10
Node_Address	Unsigned16	16 bits / 0 to $2^{16}-1$	8.1.11
Device_Info (structure)	Unsigned32	32 bits / 0 to $2^{32}-1$	8.1.12
	Unsigned32	32 bits / 0 to $2^{32}-1$	
Output_Data_Length	Unsigned16	16 bits / 32 to 8 140	8.1.13
Input_Data_Length	Unsigned16	16 bits / 32 to 8 140	8.1.14
Output_User_Data_Length	Unsigned16	16 bits / 4 to 4 058	8.1.15
Input_User_Data_Length	Unsigned16	16 bits / 4 to 4 058	8.1.16
Stop_Safety_Loop	Unsigned8	8 bits / (set per bit)	8.1.17
Stop_Safety_Loop_Oth	Unsigned8	8 bits / (set per bit)	8.1.18

8.1.2 T_Watchdog

T_Watchdog is the watchdog time for monitoring from the transmission of an SPDU to the reception of the next SPDU. This variable shall be used in the safety master and the safety slave when their SCL is in the safety data sending/receiving state.

The safety master and the safety slave shall prepare this variable for each safety connection. The unit for this variable is nanosecond, and the range is 1 to $2^{32}-1$. This value is configured by the user and shall be greater than the minimum value defined in the implementation document.

8.1.3 T_Response

T_Response is the response time for monitoring from the transmission of an SPDU to the reception of the next SPDU. This variable shall be used by the safety master SCL in all states except the safety data sending/receiving state.

The safety master shall prepare this variable for each safety connection. The unit for this variable is nanosecond, and the range is 1 to $2^{32}-1$. This value is configured by the user and shall be greater than the minimum value defined in the implementation document.

8.1.4 Master_Connection_Key

The master connection key is a random 16-bit number that the safety master generates for the connection ID. Value 0 shall not be used.

8.1.5 Slave_Connection_Key

The slave connection key is a random 16-bit number that the safety slave generates for the connection ID. Value 0 shall not be used.

8.1.6 Connection_Id

The connection ID is a 16-bit authentication code that the safety master generates (from the master connection key and the slave connection key) which is used for message authentication by both the safety master and safety slave.

8.1.7 Master_Sequence_Number

The master sequence number is a 16-bit number that the safety master increments after sending an SPDU. The range for this variable is 1 to $2^{16}-1$, and when the value reaches $2^{16}-1$, the next increment is to the value 1.

8.1.8 Extended_Master_Sequence_Number

The extended master sequence number is a 32-bit number that the safety master increments whenever the master sequence number rolls over (changing from $2^{16}-1$ to 1). The range for this variable is 0 to $2^{32}-1$, and when the value reaches $2^{32}-1$, the next increment is to the value 0.

8.1.9 Slave_Sequence_Number

The slave sequence number is a 16-bit number that the safety slave increments after sending an SPDU. The range for this variable is 1 to $2^{16}-1$, and when the value reaches $2^{16}-1$, the next increment is to the value 1.

8.1.10 Extended_Slave_Sequence_Number

The extended slave sequence number is a 32-bit number that the safety slave increments whenever the slave sequence number rolls over (changing from $2^{16}-1$ to 1). The range for this variable is 0 to $2^{32}-1$, and when the value reaches $2^{32}-1$, the next increment is to the value 0.

8.1.11 Node_Address

The node address is a 16-bit variable that holds the node address of the safety slave. The initial value shall be 0xFFFF. When the safety slave starts, the node address obtained from non-safety subsystem shall be stored in this variable.

8.1.12 Device_Info (structure)

Device information is a structured variable that holds the device information of the safety slave. The initial value for each member shall be 0xFFFFFFFF. When the safety slave starts, the device information obtained from the non-safety subsystem shall be stored in this structure. This variable has the following structure:

```
Device information (Device_Info) {
Vendor ID (Vendor_ID): 32 bits
Device code (Device_Code): 32 bits
}
```

8.1.13 Output_Data_Length

The output data length is the length, in bytes, of the SPDU sent from a safety master to a safety slave. The safety master and the safety slave shall configure this variable for each safety connection. The length in each safety connection shall be equal between the safety master and the safety slave. The range for this variable is 32 to 1488.

8.1.14 Input_Data_Length

The input data length is the length, in bytes, of the SPDU sent from a safety slave to a safety master. The safety master and the safety slave shall configure this variable for each safety connection. The length in each safety connection shall be equal between the safety master and the safety slave. The range for this variable is 32 to 1488.

8.1.15 Output_User_Data_Length

The output user data length is the length, in bytes, of the safety user data sent from a safety master to a safety slave. The safety master and the safety slave shall configure this variable for each safety connection. The length for each safety connection shall be equal between the safety master and the safety slave. The range for this variable is 4 to 732. This value shall be calculated by Formula (5).

$$\text{output\_user\_data\_length} = \frac{\text{output\_data\_length}}{2} - 12 \tag{5}$$

8.1.16 Input_User_Data_Length

The input user data length is the length, in bytes, of the safety user data sent from a safety slave to a safety master. The safety master and the safety slave shall configure this variable for each safety connection. The length for each connection shall be equal between the safety master and the safety slave. The range for this variable is 4 to 732. This value shall be calculated by Formula (6).

$$\text{input\_user\_data\_length} = \frac{\text{input\_data\_length}}{2} - 12 \tag{6}$$

8.1.17 Stop_Safety_Loop

The stop safety loop is the configuration value for setting other domains (safety functions) in which the safe data communication is halted when an error has been detected in a domain. Only the safety master shall prepare this variable. The value shall be set per bit according to the specification shown in Table 33 and downloaded from configuration tool.

Table 33 – Specification of stop safety loop setting

Domain to be set [a]	Bit	Description	Remarks
#0	0	Make the domain #0 stop safe data communication when the domain #0 has detected an error.	This bit always shall be 1. [b]
	1	Make the domain #1 stop safe data communication when the domain #0 has detected an error.	
	2	Make the domain #2 stop safe data communication when the domain #0 has detected an error.	
	3	Make the domain #3 stop safe data communication when the domain #0 has detected an error.	
	4-7	(Reserved)	
#1	0	Make the domain #0 stop safe data communication when the domain #1 has detected an error.	
	1	Make the domain #1 stop safe data communication when the domain #1 has detected an error.	This bit always shall be 1. [b]
	2	Make the domain #2 stop safe data communication when the domain #1 has detected an error.	
	3	Make the domain #3 stop safe data communication when the domain #1 has detected an error.	
	4-7	(Reserved)	
#2	0	Make the domain #0 stop safe data communication when the domain #2 has detected an error.	
	1	Make the domain #1 stop safe data communication when the domain #2 has detected an error.	
	2	Make the domain #2 stop safe data communication when the domain #2 has detected an error.	This bit always shall be 1. [b]
	3	Make the domain #3 stop safe data communication when the domain #2 has detected an error.	
	4-7	(Reserved)	
#3	0	Make the domain #0 stop safe data communication when the domain #3 has detected an error.	
	1	Make the domain #1 stop safe data communication when the domain #3 has detected an error.	
	2	Make the domain #2 stop safe data communication when the domain #3 has detected an error.	
	3	Make the domain #3 stop safe data communication when the domain #3 has detected an error.	This bit always shall be 1. [b]
	4-7	(Reserved)	

[a] The index of domains is equal to the serial number of the domains which the safety master controls.

[b] This bit always shall be 1 because the domain which detects an error always shall stop the safe data communication.

8.1.18 Stop_Safety_Loop_Oth

The stop safety loop other is the configuration value for setting other domains (safety functions) for which the safe data communication in the domain is halted when an error has been detected there. Only the safety master shall prepare this variable. The value shall be set per bit according to the specification shown in Table 34 and downloaded from configuration tool.

Table 34 – Specification of stop safety loop other setting

Domain to be set [a]	Bit	Description	Remarks
#0	0	Make the domain #0 stop safe data communication when the domain #0 has detected an error.	= 1 [b]
	1	Make the domain #0 stop safe data communication when the domain #1 has detected an error.	
	2	Make the domain #0 stop safe data communication when the domain #2 has detected an error.	
	3	Make the domain #0 stop safe data communication when the domain #3 has detected an error.	
	4-7	(Reserved)	
#1	0	Make the domain #1 stop safe data communication when the domain #0 has detected an error.	
	1	Make the domain #1 stop safe data communication when the domain #1 has detected an error.	= 1 [b]
	2	Make the domain #1 stop safe data communication when the domain #2 has detected an error.	
	3	Make the domain #1 stop safe data communication when the domain #3 has detected an error.	
	4-7	(Reserved)	
#2	0	Make the domain #2 stop safe data communication when the domain #0 has detected an error.	
	1	Make the domain #2 stop safe data communication when the domain #1 has detected an error.	
	2	Make the domain #2 stop safe data communication when the domain #2 has detected an error.	= 1 [b]
	3	Make the domain #2 stop safe data communication when the domain #3 has detected an error.	
	4-7	(Reserved)	
#3	0	Make the domain #3 stop safe data communication when the domain #0 has detected an error.	
	1	Make the domain #3 stop safe data communication when the domain #1 has detected an error.	
	2	Make the domain #3 stop safe data communication when the domain #2 has detected an error.	
	3	Make the domain #3 stop safe data communication when the domain #3 has detected an error.	= 1 [b]
	4-7	(Reserved)	

[a] The index of domains is equal to the serial number of the domains which the safety master controls.

[b] This bit shall be 1 because the domain which detects an error shall always stop the safe data communication.

9 System requirements

9.1 Indicators and switches

9.1.1 General

Table 35 shows the LEDs specified for a device implementing FSCP 19.

Table 35 – LED specifications

LED name	Safety master product required/ optional	Safety slave product required/ optional
Safety connection LED	Optional	Optional

9.1.2 Safety connection LED

The safety connection LED shows the state of the safety connection. The specification for this LED is shown Table 36.

For devices with multiple safety connections:

- if all safety connections are in the safety data sending/receiving state, this LED shall be on;
- if at least one safety connection is being established and at least one safety connection is not in the safety data sending/receiving state, this LED shall blink.

The following characters shall be used for the label of LED. (x is port number (e.g. 1, 2). When the device has only one port, this number may be omitted.)

- Safety connection x
- SACOx
- SCx

Table 36 – Safety connection LED specification

Color	LED status	Description
Green or yellow	Off	All connection has been disconnected.
	Blinking (ON: 500 ms, OFF: 500 ms)	The safety connection has been established, but the safety data is not being transmitted or received.
	On	The safety connection has been established, and the safety data is sending and receiving.

9.2 Installation guidelines

This document specifies protocol and services for a safety communication system implementing FSCP 19 which is based on the IEC 61158 series Type 24 and 27. Usage of safety devices implementing FSCP 19 requires proper installation. All devices connected to a safety communication system defined in FSCP 19 shall follow the recommendations and comply with the specifications given in IEC 61784-5-19.

9.3 Safety function response time

9.3.1 System response time

The system response time is the worst case time from occurrence of a failure or safety input until the system transitions to the safe state. The system response time is the sum of response times for each element on a safety function. A safety function in the system is shown in Figure 28.

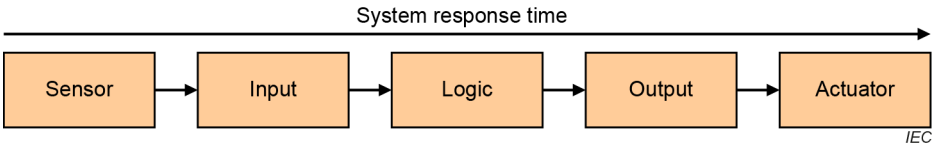


Figure 28 – Elements of safety function

These elements for FSCP 19 are shown in Figure 29.

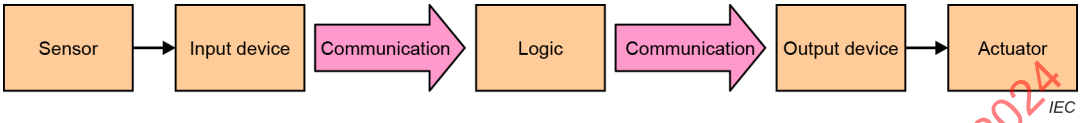


Figure 29 – Safety function of FSCP 19 system

Therefore, the system response time is calculated as sum of:

- sensor response time;
- input device response time;
- FSCP 19 input response time;
- safety controller response time;
- FSCP 19 output response time;
- output device response time;
- actuator response time.

9.3.2 FSCP 19 response time

9.3.2.1 Input response time

FSCP 19 input response time t_{ir} is measured from the input slave safety application endpoint to the master safety application endpoint.

$$t_{ir} = t_{ir1} + t_{ir2} \tag{7}$$

where

t_{ir} is the input response time;

t_{ir1} is T_Watchdog (the maximum time for the slave SCL to send the input data);

t_{ir2} is the master SCL processing time.

9.3.2.2 Output reaction time

FSCP 19 output reaction time t_{or} is measured from the master safety application endpoint to the output slave safety application endpoint.

$$t_{or} = t_{or1} + t_{or2} \tag{8}$$

where

t_{or} is the output reaction time;

t_{or1} is T_Watchdog (the maximum time for the master SCL to send the output data);

t_{or2} is the slave SCL processing time.

9.4 Duration of demands

The duration of demand by the safety-related application to the SCL shall be sufficient in duration such that demand is detected within the longest safety function response time by the application.

9.5 Constraints for calculation of system characteristics

9.5.1 Number of stations

The number of safety slaves that can be connected to the safety master shall be 63 or less.

9.5.2 Probability considerations

The calculated result of the maximum residual error rate per hour is shown in Table 37 and Figure 30. Under the condition of $P_e \leq 0{,}01$, the maximum residual error rate for typical data length is less than 10^{-9}.

Table 37 – Residual error rate

One block (incl. CRC)		SPDU (incl. CRC)		HD		Maximum of λ_{SC}		λ_{SC}				
octets	n bits	octets	n_0 bits	d_1	d_2	P_e	λ_{SC}	$P_e = 2/n_0$	$P_e = 4/n_0$	$P_e = 8/n_0$	$P_e = 16/n_0$	$P_e = 0{,}01$
16	128	32	256	6	8	0,054 7	1,23E-14	2,33E-16	2,38E-16	2,03E-15	1,08E-14	2,33E-16
76	608	152	1 216	6	6	0,009 9	2,39E-16	2,33E-16	2,33E-16	2,36E-16	2,37E-16	2,39E-16
140	1 120	280	2 240	6	6	0,005 4	2,33E-16	2,33E-16	2,33E-16	2,33E-16	2,33E-16	2,33E-16
372	2 976	744	5 952	6	6	0,002	2,33E-16	2,33E-16	2,33E-16	2,33E-16	2,33E-16	2,33E-16
744	5 952	1 488	11 904	6	6	0,001	2,33E-16	2,33E-16	2,33E-16	2,33E-16	2,33E-16	2,33E-16

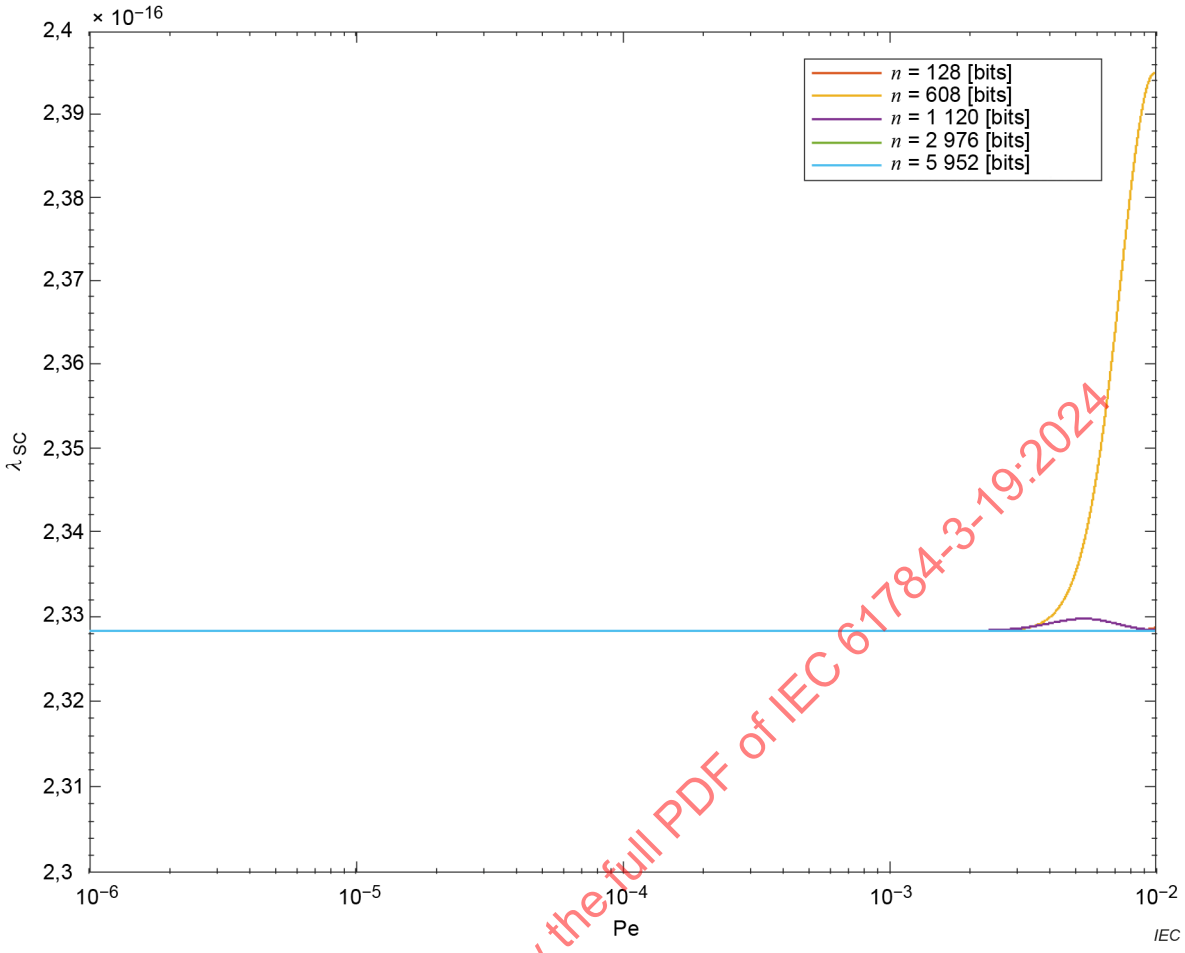


Figure 30 – Residual error rate

9.6 Maintenance

System behaviour specifications for device repair and replacement are outside the scope of this document. Usually this will be part of a functional safety management plan. Repair, replacement, maintenance, overall safety validation, overall operation, modifications, retrofits, decommissioning, and disposal according to IEC 61508 shall be considered.

It is recommended that users contact the device or system supplier for further information and for guidance in setting parameters.

Additional requirements for maintenance, as well as other requirements, are specified in IEC 61508 series, IEC 61511 series and/or IEC 62061.

9.7 Safety manual

The supplier of safety slaves that incorporate the SCL according to FSCP 19 shall prepare an appropriate safety manual according to IEC 61508. This safety manual shall also include the installation requirements as specified in 9.2.

10 Assessment

It is the manufacturer's responsibility to develop safety devices in accordance with safety standards in IEC 61508 series, IEC 61511 series, IEC 62061, and others as appropriate.

Bibliography

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/)

IEC 60050-191:1990, *International Electrotechnical Vocabulary – Chapter 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*

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*

IEC 61010-2-201, *Safety requirements for electrical equipment for measurement, control and laboratory use – Part 2-201: Particular requirements for control equipment*

IEC 61131-6, *Programmable controllers – Part 6: Functional safety*

IEC 61158-1, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-5 (all parts), *Industrial communication networks – Fieldbus specifications – Part 5: Application layer service definition*

IEC 61158-5-24, *Industrial communication networks – Fieldbus specifications – Part 5-24: Application layer service definition – Type 24 elements*

IEC 61158-5-27, *Industrial communication networks – Fieldbus specifications – Part 5-27: Application layer service definition – Type 27 elements*

IEC 61496 (all parts), *Safety of machinery – Electro-sensitive protective equipment*

IEC 61508-1, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements*

IEC 61508-4:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

IEC 61508-5, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5: Examples of methods for the determination of safety integrity*

IEC 61784-1 (all parts), *Industrial networks – Profiles – Part 1: Fieldbus profiles*

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

[3] Withdrawn

IEC 61784-3 (all parts), *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses*

IEC 61784-5 (all parts), *Industrial networks – Profiles – Part 5: Installation of fieldbuses – Installation profiles for CPF x*

IEC 61800-5-2, *Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional*

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

IEC 62443 (all parts), *Security for industrial automation and control systems*

ISO 10218-1:2011, *Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*

ISO 13849 (all parts), *Safety of machinery – Safety-related parts of control systems*

ISO 13849-1:2023, *Safety of machinery – Safety-related parts of control systems -- Part 1: General principles for design*

ISO/IEC 2382:2015, *Information technology – Vocabulary*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC/IEEE 8802-3, *Telecommunications and exchange between information technology systems – Requirements for local and metropolitan area networks – Part 3: Standard for Ethernet*

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

RÉSEAUX DE COMMUNICATION INDUSTRIELS – PROFILS –

Partie 3-19: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 19

AVANT-PROPOS

1) La Commission Électrotechnique 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. À 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'IEC attire l'attention sur le fait que la mise en application du présent document peut entraîner l'utilisation d'un brevet. L'IEC ne prend pas position quant à la preuve, à la validité et à l'applicabilité de tout droit de brevet revendiqué à cet égard. À la date de publication du présent document, l'IEC avait reçu notification qu'un brevet pouvait être nécessaire à sa mise en application. Toutefois, il y a lieu d'avertir les responsables de la mise en application du présent document que des informations plus récentes sont susceptibles de figurer dans la base de données de brevets, disponible à l'adresse https://patents.iec.ch. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

L'IEC 61784-3-19 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.

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
65C/1276/CDV	65C/1298/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce 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/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 de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé, ou
- révisé.

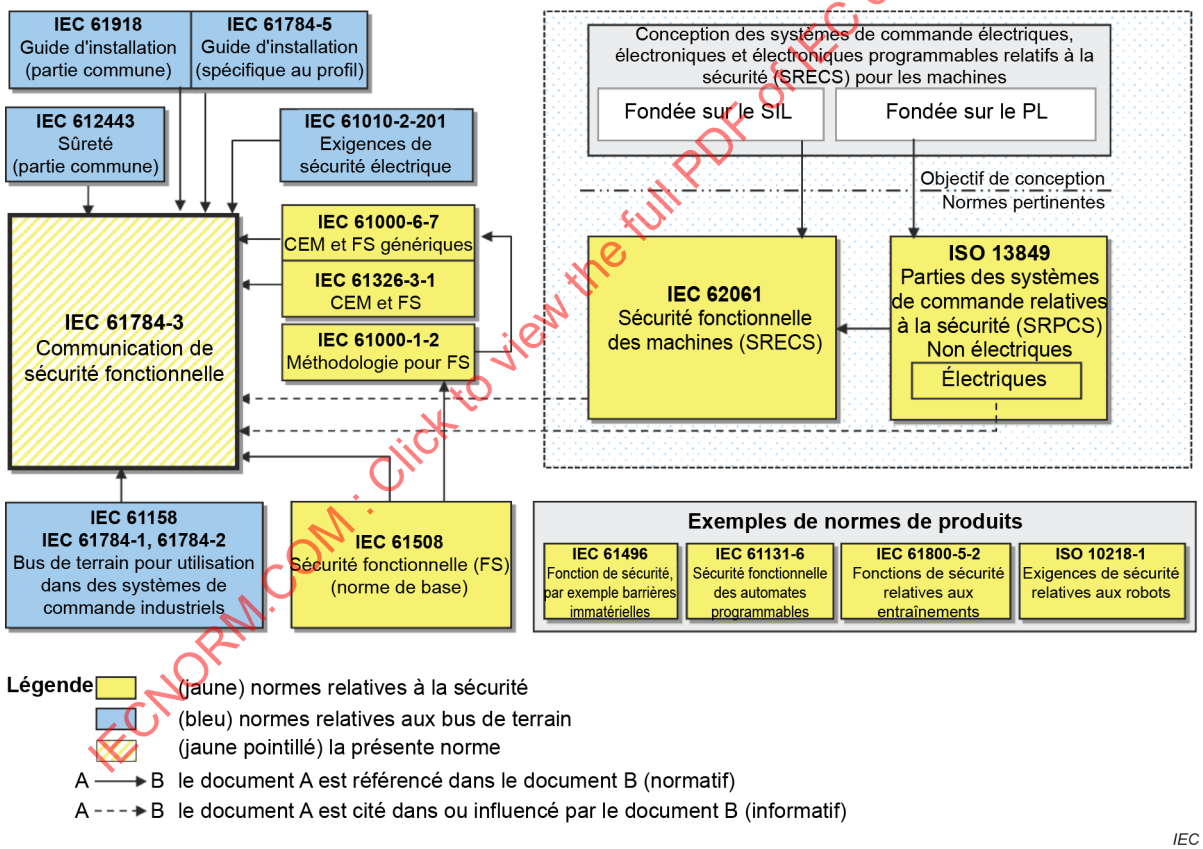
IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer ce document en utilisant une imprimante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

INTRODUCTION

La série de normes IEC 61158 relatives 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 et celles relatives à la sécurité.

La série IEC 61784-3 définit les principes qui s'appliquent aux communications de sécurité fonctionnelle par référence à la série IEC 61508; elle spécifie plusieurs couches de communication de sécurité (profils et protocoles correspondants) fondées sur les profils de communication et les couches de protocole de l'IEC 61784-1, de l'IEC 61784-2 et de la série IEC 61158. 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.

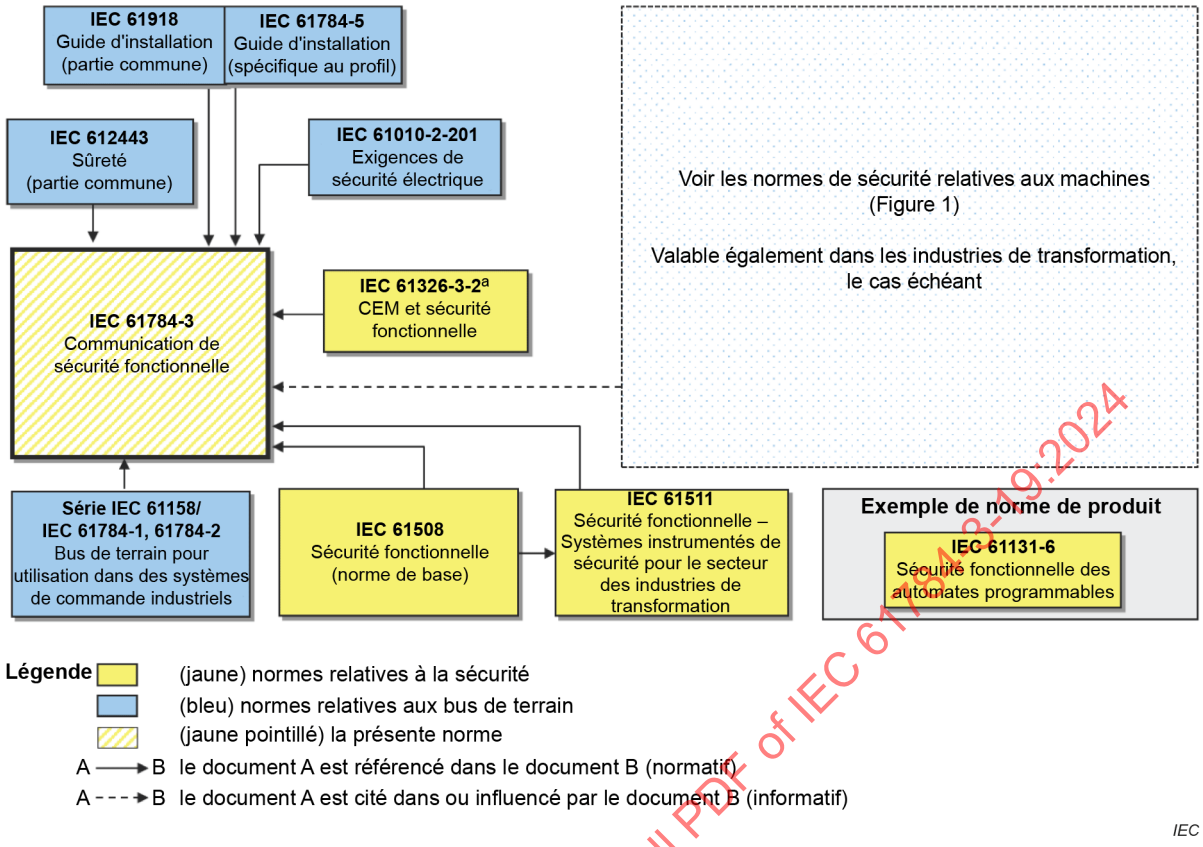
La Figure 1 représente les relations entre la série IEC 61784-3 et les normes pertinentes relatives à la sécurité et aux bus de terrain dans un environnement de machines.



NOTE L'IEC 62061 spécifie la relation entre le PL (Catégorie) et le SIL.

Figure 1 – Relations entre l'IEC 61784-3 et d'autres normes (machines)

La Figure 2 représente les relations entre la série IEC 61784-3 et les normes pertinentes relatives à la sécurité et aux bus de terrain dans un environnement de transformation.



[a] Pour des environnements électromagnétiques spécifiés; sinon, l'IEC 61326-3-1 ou l'IEC 61000-6-7.

Figure 2 – Relations entre l'IEC 61784-3 et d'autres normes (transformation)

Les couches de communication de sécurité mises en œuvre dans le cadre de systèmes relatifs à la sécurité conformément à la série IEC 61508 procurent la confiance nécessaire pour la transmission de messages (informations) entre plusieurs participants sur un bus de terrain dans un système relatif à la sécurité ou procurent une confiance 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 la série l'IEC 61784-3 permettent ainsi de pouvoir utiliser un bus de terrain avec les applications qui exigent une sécurité fonctionnelle jusqu'au niveau d'intégrité de sécurité (SIL) spécifié par son profil de communication de sécurité fonctionnelle correspondant.

Le SIL ainsi revendiqué 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).

La série IEC 61784-3 décrit:

- les principes de base de la mise en œuvre des exigences de la série IEC 61508 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 la série IEC 61784-2, y compris les extensions de la couche de sécurité aux sections relatives au service et aux protocoles de communication de la série IEC 61158.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – PROFILS –

Partie 3-19: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 19

1 Domaine d'application

La présente partie de l'IEC 61784-3 spécifie une couche de communication de sécurité (services et protocole) fondée sur l'IEC 61784-1-19, l'IEC 61784-2-19 et la série IEC 61158 (Types 24 et 27). Elle identifie les principes qui s'appliquent aux communications de sécurité fonctionnelle définies dans l'IEC 61784-3, associées à cette couche de communication de sécurité qui est destinée à être mise en œuvre sur les appareils de sécurité uniquement.

NOTE 1 Elle ne couvre pas les aspects relatifs à la sécurité électrique et à la sécurité intrinsèque. La sécurité électrique concerne les dangers comme 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[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 Le SIL ainsi revendiqué 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 61131-2, *Mesurage et contrôle des processus industriels – Automates programmables – Partie 2: Exigences et essais des équipements*

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

IEC 61158-6-24, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-24: Spécification du protocole de la couche application – Éléments de type 24*

IEC 61158-6-27, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-27: Spécification du protocole de la couche application – Éléments de type 27*

[1] Dans les pages suivantes du présent document, "IEC 61508" est utilisé en lieu et place de l'expression "la série IEC 61508".

IEC 61326-3-1, *Matériel électrique de mesure, de commande et de laboratoire – Exigences relatives à la CEM – Partie 3-1: Exigences d'immunité pour les systèmes relatifs à la sécurité et pour les matériels destinés à réaliser des fonctions relatives à la sécurité (sécurité fonctionnelle) – Applications industrielles générales*

IEC 61326-3-2, *Matériel électrique de mesure, de commande et de laboratoire – Exigences relatives à la CEM – Partie 3-2: Exigences d'immunité pour les systèmes relatifs à la sécurité et pour les matériels destinés à réaliser des fonctions relatives à la sécurité (sécurité fonctionnelle) – Applications industrielles dont l'environnement électromagnétique est spécifié*

IEC 61508 (toutes les parties), *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité*

IEC 61511 (toutes les parties), *Sécurité fonctionnelle – Systèmes instrumentés de sécurité pour le secteur des industries de transformation*

IEC 61784-1-19:2023, *Réseaux industriels – Profils – Partie 1-19: Profils de bus de terrain – Famille de profils de communication 19*

IEC 61784-2-19:2023, *Réseaux industriels – Profils – Partie 2-19: Profils de bus de terrain supplémentaires pour les réseaux en temps réel fondés sur l'ISO/IEC/IEEE 8802-3 – CPF 19*

IEC 61784-3, *Réseaux de communication industriels – Profils – Partie 3: Bus de terrain de sécurité fonctionnelle – Règles générales et définitions de profils*

IEC 61784-5-19, *Réseaux industriels – Profils – Partie 5-19: Installation des bus de terrain – Profils d'installation pour CPF 19*

IEC 62061, *Sécurité des machines – Sécurité fonctionnelle des systèmes de commande relatifs à la sécurité*

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 https://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse https://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
canal de communication
connexion logique entre deux points limites d'un *système de communication*

3.1.1.2
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

3.1.1.3
connexion
liaison logique entre objets applicatifs au sein du même appareil ou d'appareils différents

3.1.1.4
CRC
contrôle de redondance cyclique
<valeur> donnée redondante déduite, et enregistrée ou transmise simultanément, d'un bloc de données afin de détecter toute corruption des données

3.1.1.5
CRC
contrôle de redondance cyclique
<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.

3.1.1.6
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, modifiée – Des Notes à l'article ont été ajoutées.]

3.1.1.7
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, modifiée – Les notes et figures ont été remplacées.]

3.1.1.8
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, modifiée – La référence à la figure a été supprimée.]

3.1.1.9
trame
synonyme déconseillé de DLPDU

3.1.1.10
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*

3.1.1.11
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, modifiée – "(généralement des octets)" a été rajouté, les notes et la source ont été supprimées.]

3.1.1.12
redondance
existence de plusieurs moyens pour accomplir une fonction requise ou pour représenter des informations

[SOURCE: IEC 61508-4:2010, 3.4.6, modifiée – L'exemple et les notes ont été supprimés.]

3.1.1.13
canal de communication de sécurité
SC
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*.

Note 2 à l'article: L'abréviation "SC" est dérivée du terme anglais développé correspondant "safety communication channel".

3.1.1.14
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".

3.1.1.15
connexion de sécurité
connexion qui utilise le protocole de sécurité pour des transactions de communications

3.1.1.16
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.

3.1.1.17
appareil de sécurité
appareil conçu conformément à l'IEC 61508 et qui met en œuvre le profil de communication de sécurité fonctionnelle

3.1.1.18
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, modifiée – Les références et l'exemple ont été supprimés.]

3.1.1.19
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, adopté dans l'IEC 61784-3:2021, 5.2.4 est traité par les profils de communication de sécurité fonctionnelle définis dans le présent document.

3.1.1.20
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 l'IEC 61508-1:2010, Tableaux 2 et 3.

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 modifiée – La note 4 à l'article a été ajoutée dans le français.]

3.1.1.21
mesure de sécurité
mesure permettant 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.

3.1.1.22
PDU de sécurité
SPDU
PDU transférée par le biais du *canal de communication de sécurité*

Note 1 à l'article: La 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 de la SPDU en vue de son insertion dans des champs séparés de la *trame de bus de terrain*.

Note 3 à l'article: L'abréviation "SPDU" est dérivée du terme anglais développé correspondant "safety PDU".

3.1.1.23
application relative à la sécurité
programmes conçus conformément à l'IEC 61508 pour satisfaire aux exigences SIL de l'application

3.1.1.24
système relatif à la sécurité
système qui exécute les *fonctions de sécurité* conformément à l'IEC 61508

3.1.1.25
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

3.1.2 CPF 19: Termes et définitions supplémentaires

3.1.2.1
maître de sécurité
entité de communication active qui transmet les *données de sortie de sécurité* à un esclave de sécurité et qui reçoit les données d'entrée de sécurité envoyées par un esclave de sécurité

3.1.2.2
esclave de sécurité
entité de communication active qui reçoit les données de sortie de sécurité envoyées par un maître de sécurité et qui transmet des données de sortie de sécurité à un maître de sécurité

3.1.2.3
maître non relatif à la sécurité
entité de communication active qui transmet les données de sortie non relatives à la sécurité à un esclave non relatif à la sécurité et qui reçoit les données d'entrée non relatives à la sécurité envoyées par un esclave non relatif à la sécurité

3.1.2.4
esclave non relatif à la sécurité
entité de communication active qui reçoit les données de sortie non relatives à la sécurité envoyées par un maître non relatif à la sécurité et qui transmet des données de sortie non relatives à la sécurité à un maître non relatif à la sécurité

3.1.2.5
adresse de nœud
identificateur de chaque entité de communication

3.1.2.6
données de sortie de sécurité
SPDU transmise par un maître de sécurité à un esclave de sécurité

3.1.2.7
données d'entrée de sécurité
SPDU transmise par un esclave de sécurité à un maître de sécurité

3.1.2.8
données utilisateur de sortie de sécurité
données définies par l'utilisateur dans les données de sortie de sécurité

3.1.2.9
données utilisateur d'entrée de sécurité
données définies par l'utilisateur dans les données d'entrée de sécurité

3.1.2.10
état de sécurité intrinsèque
état dans lequel passent les maîtres de sécurité et les esclaves de sécurité en cas d'erreur

Note 1 à l'article: Dans cet état, les données utilisateur de sortie de sécurité et les données utilisateur d'entrée de sécurité sont toutes définies sur "0".

3.2 Symboles et termes abrégés

3.2.1 Symboles et abréviations communs

CP (Communication Profile)	profil de communication	[IEC 61784-1 (toutes les parties)]
CPF (Communication Profile Family)	famille de profils de communication	[IEC 61784-1 (toutes les parties)]
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	
EUC (Equipment Under Control)	équipement commandé	[IEC 61508-4:2010]
E/E/PE (Electrical/Electronic/Programmable Electronic)	électrique/électronique/électronique programmable	[IEC 61508-4:2010]
FAL (Fieldbus Application Layer)	couche application de bus de terrain	[IEC 61158-5 (toutes les parties)]
FSCP (Functional Safety Communication Profile)	profil de communication de sécurité fonctionnelle	
PDU (Protocol Data Unit)	unité de données de protocole	[ISO/IEC 7498-1]
Pe	probabilité d'erreurs sur les éléments binaires	
PhL (Physical Layer)	couche physique	[ISO/IEC 7498-1]
PL (Performance Level)	niveau de performance	[ISO 13849-1]
SC (Safety Communication Channel)	canal de communication de sécurité	
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 19: Symboles et abréviations supplémentaires

NVS (Non-Volatile Storage) mémoire rémanente

LSB (Least Significant Bit) bit de poids faible

3.3 Conventions

Les conventions utilisées dans le présent document sont définies dans l'IEC 61784-1-19, l'IEC 61784-2-19 et la série IEC 61158 Types 24 et 27.

4 Vue d'ensemble du FSCP 19 (MECHATROLINK Safety)

La CPF 19 (couramment appelée MECHATROLINK™[2]) est définie dans l'IEC 61784-1-19 et l'IEC 61784-2-19, avec les couches de protocole de bus de terrain définies dans la série IEC 61158 Types 24 et 27. Le profil de communication de sécurité fonctionnelle FSCP 19 (MECHATROLINK Safety) est fondé sur la CPF 19 et la SCL définie dans la présente partie.

[2] MECHATROLINK™ est une appellation commerciale de Yaskawa Electric Corporation. 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 l'appellation commerciale MECHATROLINK™. L'utilisation de l'appellation commerciale MECHATROLINK™ exige l'autorisation de Yaskawa Electric Corporation et la conformité aux conditions de son utilisation (essais et validation).

La Figure 3 donne une vue d'ensemble de la structure du système pour le FSCP 19. Dans le domaine de contrôle 1, un maître non relatif à la sécurité est chargé d'établir et de gérer les canaux de communication non relatifs à la sécurité dans le cadre d'une relation maître-esclave avec les esclaves non relatifs à la sécurité utilisant la CPF 19.

Dans les domaines de contrôle 2 et 3, les maîtres de sécurité sont chargés d'établir et de gérer les canaux de communication de sécurité dans une relation maître-esclave avec les esclaves de sécurité utilisant le FSCP 19. Chaque maître de sécurité doit être limité à un seul domaine de contrôle de sécurité.

Le FSCP 19 permet aux communications de sécurité et aux communications non relatives à la sécurité de coexister. Un appareil peut mettre en œuvre la fonction de communication non relative à la sécurité ou la fonction de communication relative à la sécurité, ou les deux.

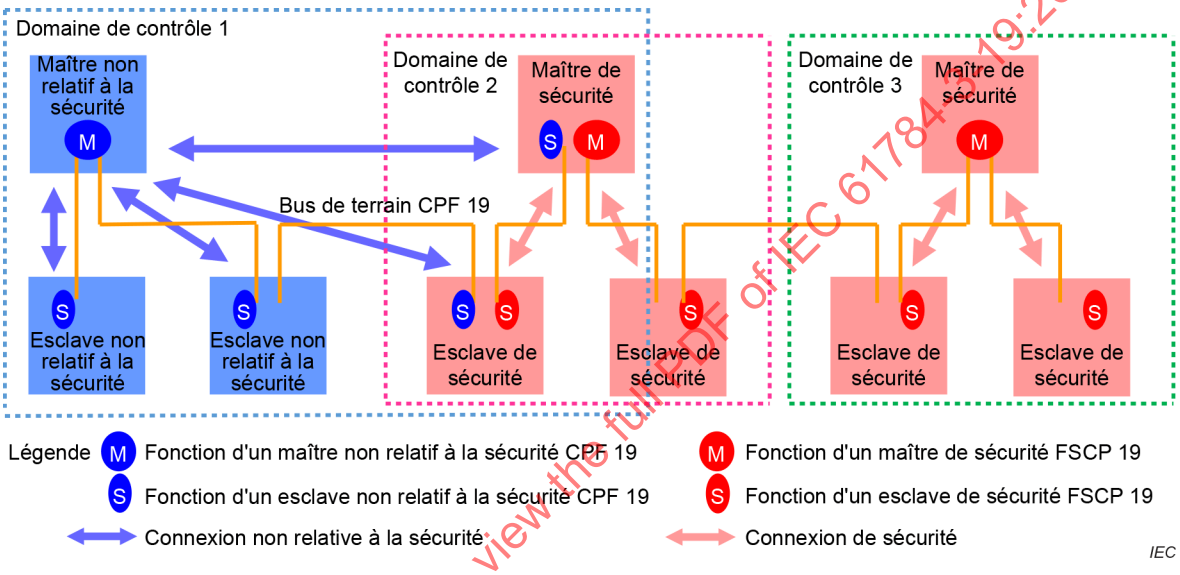


Figure 3 – Système FSCP 19 de base

5 Généralités

5.1 Documents externes de spécifications applicables au profil

D'autres documents fournissant des informations complémentaires sont en cours d'élaboration.

5.2 Exigences fonctionnelles de sécurité

Le présent document spécifie les services et protocoles d'un système de communication de sécurité fonctionnelle fondé sur le bus de terrain CPF 19. Les technologies de communication spécifiées dans le présent document doivent être mises en œuvre uniquement dans les appareils conçus conformément aux exigences de l'IEC 61508.

Les exigences suivantes doivent s'appliquer au développement des appareils qui mettent en œuvre le protocole FSCP 19. Les mêmes exigences ont été utilisées dans le développement de FSCP 19.

- Le protocole FSCP 19 est conçu de manière à prendre en charge le niveau d'intégrité de sécurité 3 (SIL 3) (voir l'IEC 61508).
- Les mises en œuvre de FSCP 19 doivent être conformes à l'IEC 61508.
- Les exigences de base qui s'appliquent au développement du protocole FSCP 19 sont spécifiées dans l'IEC 61784-3.

- Les conditions d'environnement doivent satisfaire à l'IEC 61131-2 pour les niveaux de base et à l'IEC 61326-3-1 et l'IEC 61326-3-2 pour les essais de marge de sécurité, sauf si des normes de produits spécifiques existent.
- Sauf spécification contraire dans la présente partie, les exigences relatives à la CPF 19 ne doivent pas être modifiées pour la sécurité.

5.3 Mesures de sécurité

5.3.1 Généralités

Le Tableau 1 répertorie les mesures de sécurité utilisées dans le FSCP 19.

Tableau 1 – Erreurs de communication et mesures de sécurité

Erreurs de communication	**Mesures de sécurité**				
	Numéro de séquence (voir le 5.3.2)	Délai (voir le 5.3.3)	ID de connexion (voir le 5.3.4)	CRC (voir le 5.3.5)	Redondance avec contre-vérification (voir le 5.3.6)
Corruption				X	X
Répétition non prévue	X				
Séquence incorrecte	X				
Perte	X	X			
Retard inacceptable		X			
Insertion	X		X		
Déguisement	X		X	X	X
Adressage			X		
Répétition causée par des défaillances de la mémoire dans des produits réseaux non relatifs à la sécurité	X				

5.3.2 Numéro de séquence

Pour détecter une défaillance du réseau, un maître de sécurité et des esclaves de sécurité doivent posséder quatre types de numéros de séquence pour chaque instance de connexion, conformément au Tableau 2.

Tableau 2 – Liste des numéros de séquence

Numéro de séquence	Taille	Description
Numéro de séquence du maître	16 bits	Inclus dans la SPDU Transmis par un maître de sécurité à un esclave de sécurité Vérifié par l'esclave de sécurité La valeur initiale est 0x 0001
Numéro de séquence étendu du maître	32 bits	Non inclus dans la SPDU Vérifié par l'esclave de sécurité La valeur initiale est 0x 0000 0000
Numéro de séquence de l'esclave	16 bits	Inclus dans la SPDU Transmis par un esclave de sécurité à un maître de sécurité Vérifié par le maître de sécurité La valeur initiale est 0x 0001
Numéro de séquence étendu de l'esclave	32 bits	Non inclus dans la SPDU Vérifié par le maître de sécurité La valeur initiale est 0x 0000 0000

Le numéro de séquence du maître doit être incrémenté de 1 lorsque le maître de sécurité transmet une SPDU. Le numéro de séquence de l'esclave doit être incrémenté de 1 lorsque l'esclave de sécurité transmet une SPDU. Chaque numéro de séquence doit être incrémenté à partir d'une valeur initiale 1, et cette valeur de 16 bits doit se réinitialiser; la valeur après $2^{16}-1$ est 1. La valeur 0 ne doit pas être utilisée.

Lorsque le numéro de séquence du maître passe de $2^{16}-1$ à 1, le numéro de séquence étendu du maître doit être incrémenté de 1. Lorsque le numéro de séquence de l'esclave passe de $2^{16}-1$ à 1, le numéro de séquence étendu de l'esclave doit être incrémenté de 1. Chaque numéro de séquence étendu doit être incrémenté à partir d'une valeur initiale 0, et cette valeur de 32 bits doit se réinitialiser; la valeur après $2^{32}-1$ est 0.

Le maître de sécurité et l'esclave de sécurité doivent initialiser ces variables avant d'établir une connexion. La Figure 4 représente la procédure d'incrémentation du numéro de séquence.

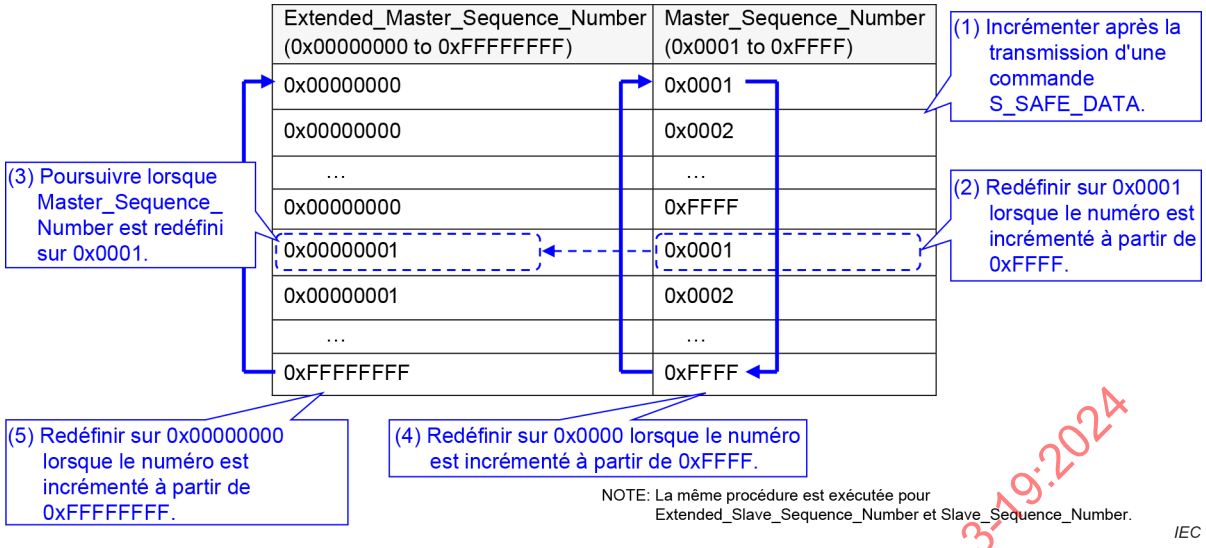


Figure 4 – Procédure d'incrémentation du numéro de séquence

Chaque numéro de séquence doit être incrémenté et vérifié de la manière décrite dans le processus de transmission et de réception pour toutes les commandes, avec les exceptions suivantes:

- le numéro de séquence indiqué dans la commande S_NOP doit toujours être zéro;
- le numéro de séquence indiqué dans la commande S_FAIL_SAFE doit conserver la dernière valeur qu'il avait immédiatement avant le passage de la SCL à l'état de sécurité intrinsèque.

5.3.3 Délai

Après une réception normale suivie de la transmission d'une commande S_SAFE_DATA, la SCL doit démarrer un temporisateur de chien de garde et le réinitialiser en cas de réception normale d'une commande S_SAFE_DATA ultérieure, comme cela est représenté à la Figure 5.

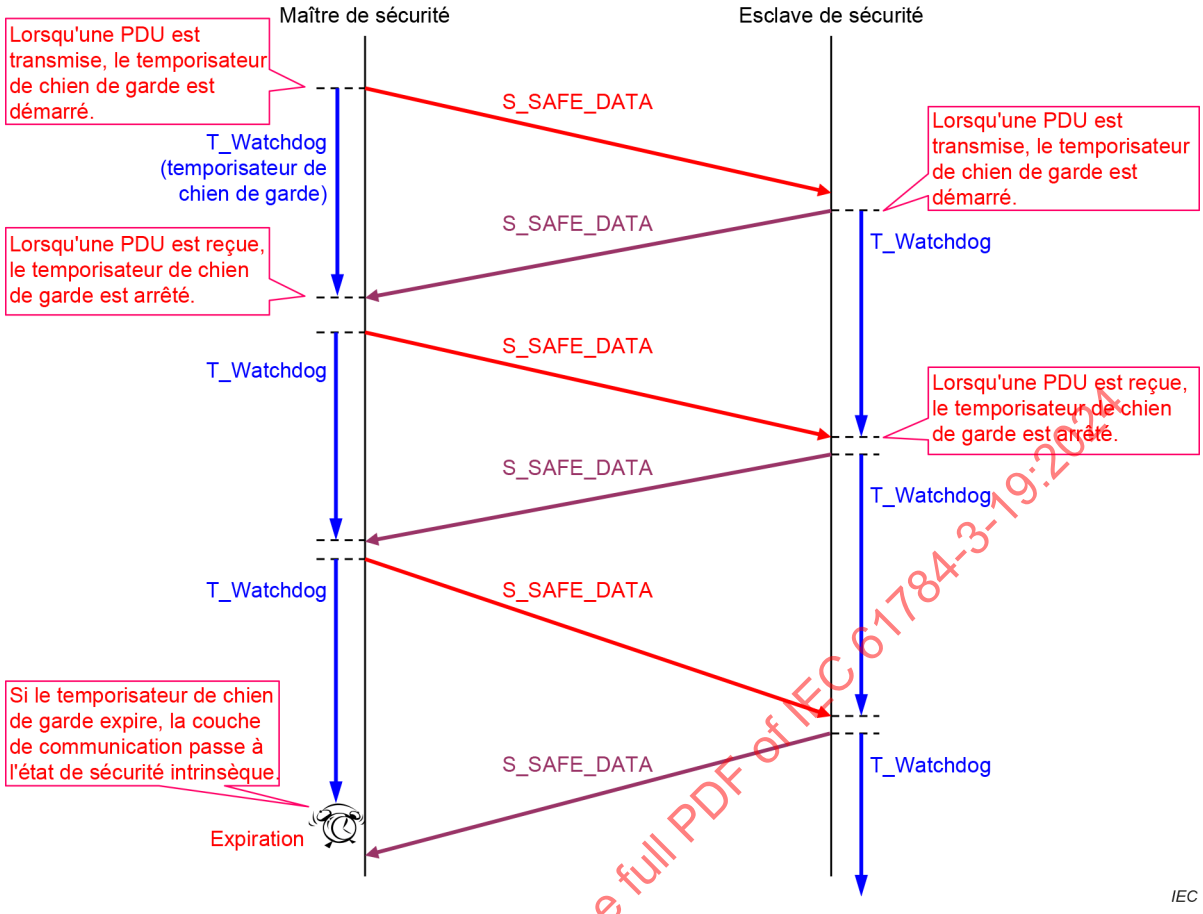


Figure 5 – Délai avec temporisateur de chien de garde

L'expiration de ce temporisateur de chien de garde déclenche le passage de la SCL à l'état de sécurité intrinsèque; la SCL doit indiquer cet événement à l'application de sécurité. La valeur de T_Watchdog est configurée par l'utilisateur au moyen d'un outil de configuration.

La temporisation de transmission de la commande S_SAFE_DATA doit être synchronisée entre un maître de sécurité et les esclaves de sécurité. Le temps relatif entre le maître et les esclaves doit être fixe, conformément à la Figure 6.

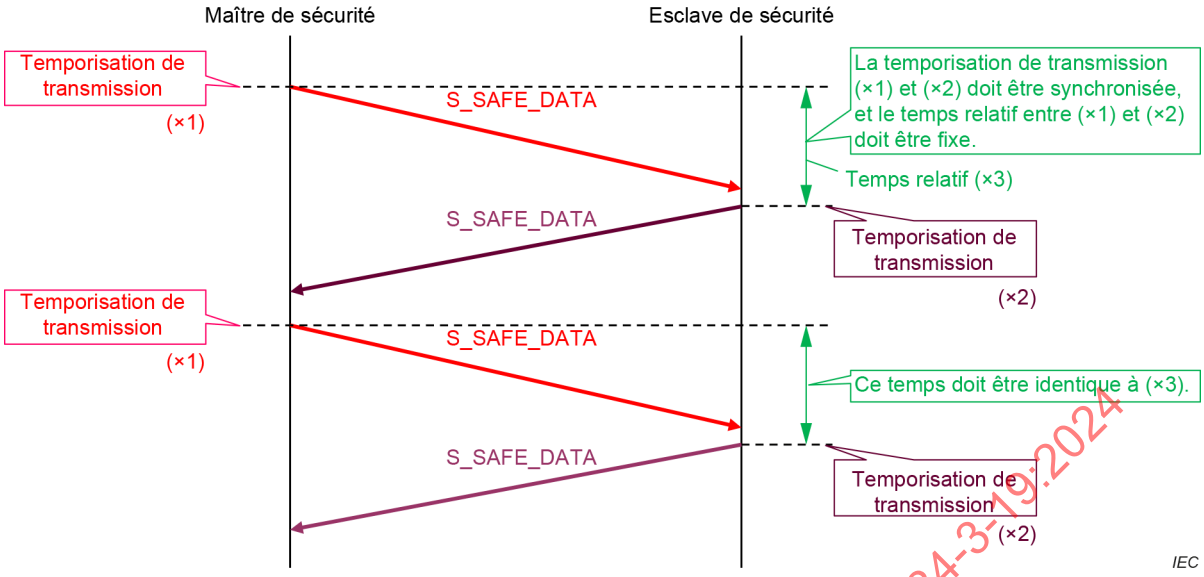


Figure 6 – Synchronisation de la temporisation de transmission

Pour les autres commandes que S_SAFE_DATA, le maître de sécurité doit démarrer un temporisateur de réponse dès qu'il reçoit une SPDU et surveiller le temps jusqu'à ce que l'esclave de sécurité reçoive la réponse, conformément à la Figure 7.

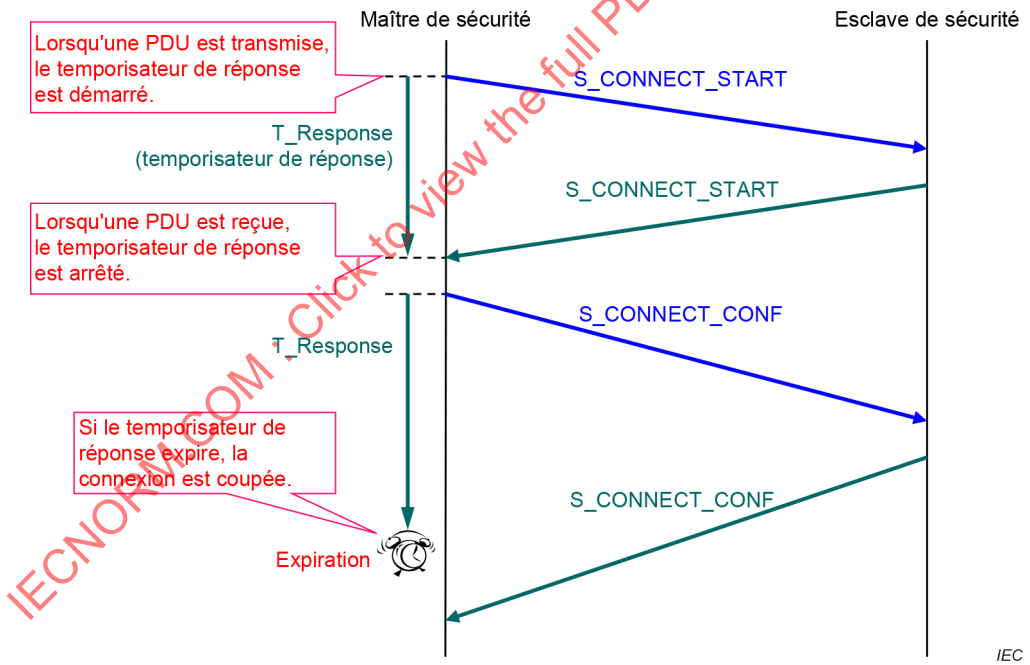


Figure 7 – Délai avec temporisateur de réponse

Le maître de sécurité ne surveille pas les commandes S_NOP ou S_FAIL_SAFE. L'esclave de sécurité surveille uniquement les commandes S_SAFE_DATA.

L'expiration de ce temporisateur de réponse doit déclencher la fin de la connexion par la SCL du maître de sécurité. La valeur de T_Response est configurée par l'utilisateur au moyen d'un outil de configuration.

5.3.4 ID de connexion

Lors de la négociation de la connexion, le maître de sécurité et l'esclave de sécurité échangent leurs clés de connexion à partir desquelles le maître de sécurité génère un ID de connexion unique. Cet ID de connexion est transmis à l'esclave de sécurité, et est ensuite utilisé par le maître et l'esclave pour l'authentification de connexion. Pour plus d'informations, voir le 7.3.1.2.

5.3.5 Calcul du CRC

Une valeur de CRC doit être calculée pour chaque bloc d'une SPDU en utilisant des polynômes générateurs uniques pour chaque bloc. Dans le calcul du CRC pour le Bloc 1, le polynôme $g_1(x)$ = 0x90022004 doit être utilisé conformément à la Formule (1).

$$g_1(x) = x^{32} + x^{29} + x^{18} + x^{14} + x^3 + 1 \quad (1)$$

Dans le calcul du CRC pour le Bloc 2, le polynôme $g_2(x)$ = 0x992C1A4C doit être utilisé conformément à la Formule (2).

$$g_2(x) = x^{32} + x^{29} + x^{28} + x^{25} + x^{22} + x^{20} + x^{19} + x^{13} + x^{12} + x^{10} + x^7 + x^4 + x^3 + 1 \quad (2)$$

Dans les polynômes $g_1(x)$ et $g_2(x)$, la distance de Hamming minimale est de 6 lorsque la taille des données (en incluant le CRC) est inférieure ou égale à 4 092 octets. Si la taille des données est supérieure à 4 092 octets, la distance de Hamming minimale est inférieure à 6 et le taux d'erreurs résiduelles doit être calculé.

Les valeurs de départ du générateur de CRC sont uniques pour chaque bloc, conformément au Tableau 3.

Tableau 3 – Valeurs de départ du CRC

Bloc	Valeur de départ
1	0x FFFF FFFF
2	0x FFFF FFFE

Le CRC doit être calculé à partir du bit de poids faible (LSB, *Least Significant Bit*), dans l'ordre suivant:

1) Commande
2) ID de connexion
3) Numéro de séquence
4) Numéro d'état
5) Données

5.3.6 Redondance avec contre-vérification

5.3.6.1 Génération des données redondantes

Les données redondantes doivent être générées conformément à la procédure suivante, en procédant aux contrôles de comparaison indiqués en **gras**:

1) Charger les données dans le Bloc 1 pour chaque canal;
2) Copier les données du Bloc 1 dans le Bloc 2 pour chaque canal;
3) **Comparer les données du Bloc 1 à celles du Bloc 2**;
4) Calculer les valeurs CRC 1 et CRC 2 pour le Bloc 1 et le Bloc 2, conformément au 5.3.5;
5) Pour le canal B, calculer le CRC 3 à partir du CRC 1 et du CRC 2 et transférer le CRC 1, le CRC 2 et le CRC 3 du canal B au canal A;
6) Pour le canal A, calculer le CRC 3 à partir du CRC 1 et du CRC 2 transférés depuis le canal B; **comparer le CRC 3 calculé dans le canal A au CRC 3 transféré depuis le Canal B**;
7) **Comparer le CRC 1 et le CRC 2 calculés dans le Canal A à ceux du Canal B**;
8) Notifier le résultat du contrôle du canal A au canal B avec ses données d'inversion de bits; le résultat du contrôle comporte 8 bits, où 0x01 signifie que le résultat est OK et 0x00 signifie que le résultat n'est pas OK (Not Good / NG);
9) Vérifier le résultat du contrôle notifié depuis le Canal A; NE PAS prendre les données d'inversion de bits notifiées depuis le canal A et vérifier si elles sont identiques au résultat du contrôle notifié depuis le canal A;
10) Finaliser le processus de génération dans le Canal B;
11) Notifier le résultat du contrôle du canal B au canal A avec ses données d'inversion de bits; le résultat du contrôle comporte 8 bits, où 0x01 signifie que le résultat est OK et 0x00 signifie que le résultat n'est pas OK (Not Good / NG);
12) Vérifier le résultat du contrôle notifié depuis le canal B; NE PAS prendre les données d'inversion de bits notifiées depuis le canal B et vérifier si elles sont identiques au résultat du contrôle notifié depuis le canal B;
13) Finaliser le processus de génération dans le canal A.

Si tous les contrôles de comparaison sont réussis, le CRC 1 et le CRC 2 sont utilisés comme CRC dans la SPDU transmise.

La Figure 8 représente le processus de génération des données redondantes.

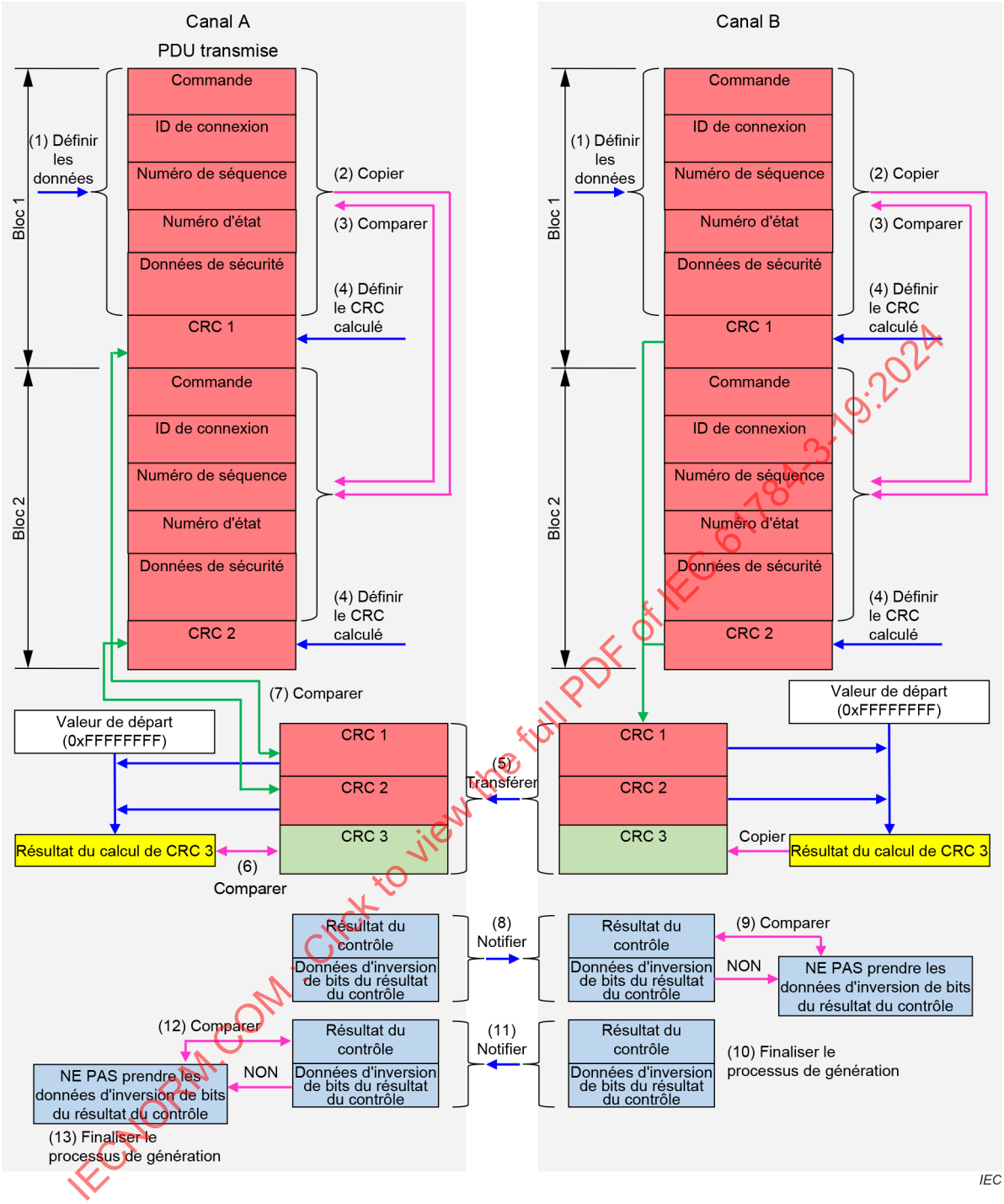


Figure 8 – Processus de génération des données redondantes

5.3.6.2 Vérification des données redondantes

Après leur réception, les données redondantes doivent être vérifiées conformément à la procédure suivante, en procédant aux contrôles de comparaison indiqués en **gras**:

1) Vérifier le Bloc 1 à l'aide du CRC 1 et le Bloc 2 à l'aide du CRC 2;
2) Comparer les données redondantes du Bloc 1 à celles du Bloc 2;
3) Pour le canal B, calculer le CRC 3 à partir du CRC 1 et du CRC 2 et transférer le CRC 1, le CRC 2 et le CRC 3 du canal B au canal A;

4) Pour le canal A, calculer le CRC 3 à partir du CRC 1 et du CRC 2 transférés depuis le canal B; **comparer le CRC 3 calculé dans le canal A au CRC 3 transféré depuis le canal B**;
5) **Comparer le CRC 1 et le CRC 2 calculés dans le canal A à ceux du canal B**;
6) Notifier le résultat du contrôle du canal A au canal B avec ses données d'inversion de bits; le résultat du contrôle comporte 8 bits, où 0x01 signifie que le résultat est OK et 0x00 signifie que le résultat n'est pas OK (Not Good / NG);
7) Vérifier le résultat du contrôle notifié depuis le canal A; NE PAS prendre les données d'inversion de bits notifiées depuis le canal A et vérifier si elles sont identiques au résultat du contrôle notifié depuis le canal A;
8) Finaliser le processus de vérification dans le canal B;
9) Notifier le résultat du contrôle du canal B au canal A avec ses données d'inversion de bits; le résultat du contrôle comporte 8 bits, où 0x01 signifie que le résultat est OK et 0x00 signifie que le résultat n'est pas OK (Not Good / NG);
10) Vérifier le résultat du contrôle notifié depuis le canal B; NE PAS prendre les données d'inversion de bits notifiées depuis le canal B et vérifier si elles sont identiques au résultat du contrôle notifié depuis le canal B;
11) Finaliser le processus de vérification dans le canal A.

Si tous les contrôles de vérification et de comparaison sont réussis, la SPDU est considérée comme valide.

La Figure 9 représente le processus de vérification des données redondantes.

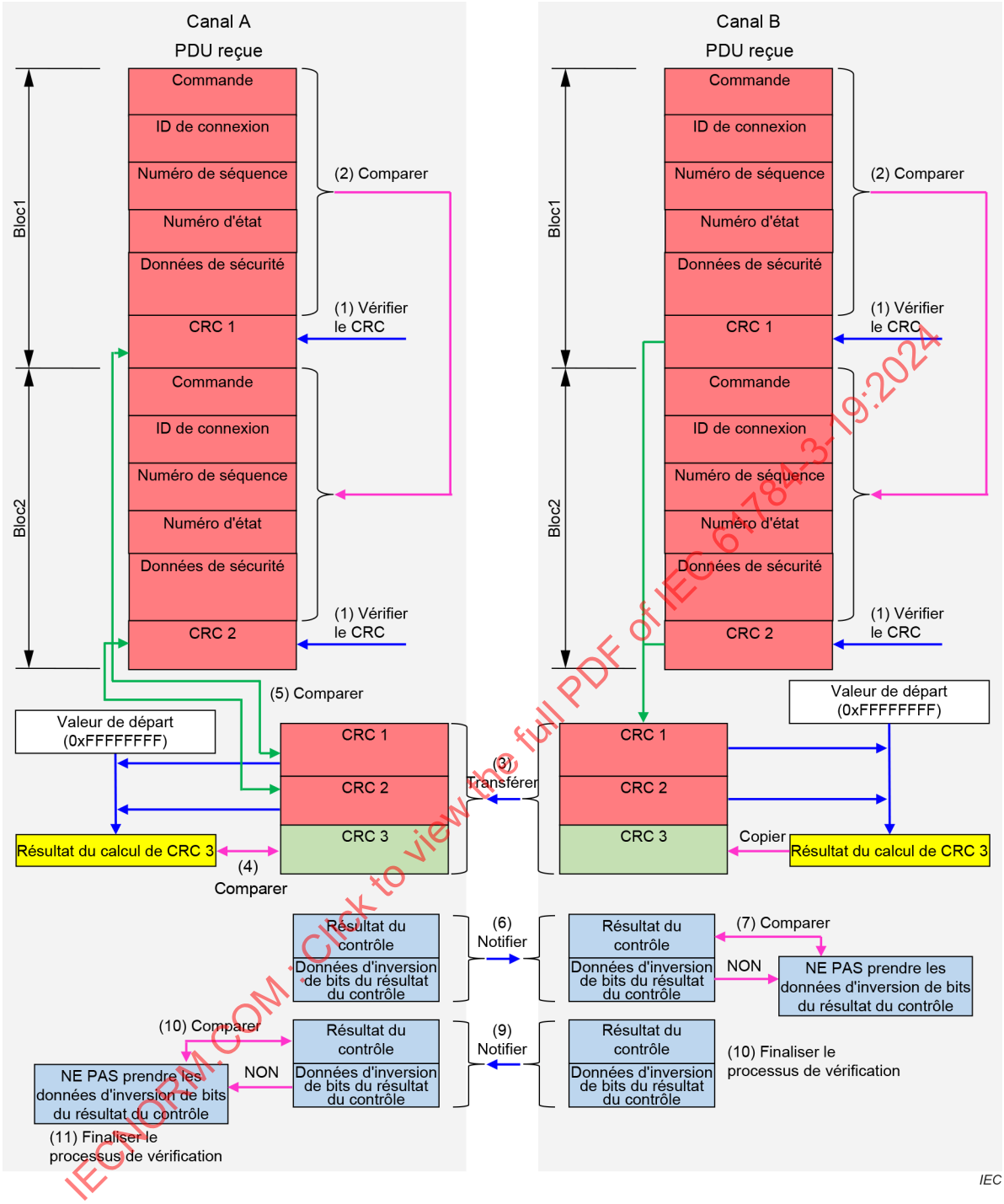


Figure 9 – Processus de vérification des données redondantes

5.4 Structure de la couche de communication de sécurité

La SCL FSCP 19 est spécifiée; elle se superpose à la FAL CPF 19, comme cela est représenté à la Figure 10.

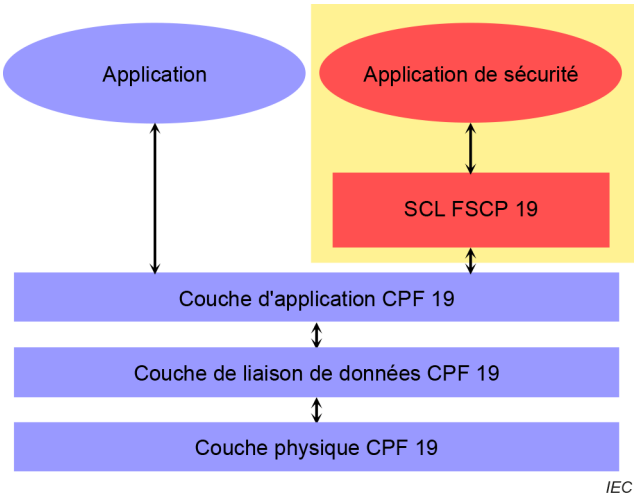


Figure 10 – Structure de la SCL

La SCL FSCP 19 exige que la FAL CPF 19 soit configurée avec une taille de données utilisateur minimale de 32 octets.

5.5 Relations avec la FAL (et avec la DLL et la PhL)

5.5.1 Généralités

Il n'existe aucune exigence FAL autre que celles énoncées dans le présent document. Le FSCP 19 peut être mis en œuvre par superposition sur la FAL de tout profil CPF 19.

5.5.2 Types de données

Pour plus d'informations sur les types de données, voir le 7.1, l'IEC 61158-6-24 et l'IEC 61158-6-27.

6 Services de la couche de communication de sécurité

6.1 Description des services

6.1.1 S_CONNECT_START

Le maître de sécurité et l'esclave de sécurité doivent échanger leurs clés de connexion pour générer l'ID de connexion. Le maître de sécurité doit générer un nombre aléatoire et le définir dans une SPDU de la commande S_CONNECT_START en tant que clé de connexion du maître. Le maître de sécurité transmet la SPDU à l'esclave de sécurité; l'esclave de sécurité sauvegarde la clé de connexion du maître et transmet le nombre aléatoire généré en tant que clé de connexion de l'esclave. Le Tableau 4 répertorie les données de la commande S_CONNECT_START.

Tableau 4 – Données de la commande S_CONNECT_START

octet	Nom	Description
0	Clé de connexion du maître (bits 0-7)	Le maître de sécurité génère un nombre aléatoire. L'esclave de sécurité définit la valeur envoyée par le maître de sécurité. Cette valeur est utilisée pour générer le Connection_Id dans le maître de sécurité.
1	Clé de connexion du maître (bits 8-15)	
2	Adresse de nœud de l'esclave (bits 0-7)	Le maître de sécurité définit l'adresse de nœud de l'esclave enregistrée dans le paramètre de configuration. L'esclave de sécurité définit sa propre adresse de nœud. Cette valeur est utilisée pour confirmer que l'adresse est conforme à l'adresse de nœud dans l'esclave de sécurité.
3	Adresse de nœud de l'esclave (bits 8-15)	
4	Clé de connexion de l'esclave (bits 0-7)	Maître de sécurité: défini sur 0. L'esclave de sécurité génère un nombre aléatoire. Cette valeur est utilisée pour générer le Connection_Id dans le maître de sécurité.
5	Clé de connexion de l'esclave (bits 8-15)	

Le Tableau 5 et le Tableau 6 correspondent à l'exemple d'une SPDU de 16 octets. Dans cet exemple, les données sont fragmentées en 2 SPDU. Les données sont encapsulées en commençant par l'octet de poids faible et l'espace à la fin doit être rempli avec la valeur 0x00 (voir les octets 10 et 11 de l'exemple).

Tableau 5 – SPDU de la commande S_CONNECT_START (1re SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_CONNECT_START = 0x01	S_CONNECT_START = 0x01
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7) = 0x00	ID de connexion (bits 0-7) = 0x00
3	ID de connexion (bits 8-15) = 0x00	ID de connexion (bits 8-15) = 0x00
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Clé de connexion du maître (bits 0-7) = nombre aléatoire	Clé de connexion du maître (bits 0-7)
9	Clé de connexion du maître (bits 8-15) = nombre aléatoire	Clé de connexion du maître (bits 8-15)
10	Adresse de nœud de l'esclave (bits 0-7)	Adresse de nœud de l'esclave (bits 0-7)
11	Adresse de nœud de l'esclave (bits 8-15)	Adresse de nœud de l'esclave (bits 8-15)
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

Tableau 6 – SPDU de la commande S_CONNECT_START (2nde SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_CONNECT_START = 0x01	S_CONNECT_START = 0x01
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7) = 0x00	ID de connexion (bits 0-7) = 0x00
3	ID de connexion (bits 8-15) = 0x00	ID de connexion (bits 8-15) = 0x00
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Clé de connexion de l'esclave (bits 0-7) = 0x00	Clé de connexion de l'esclave (bits 0-7) = nombre aléatoire
9	Clé de connexion de l'esclave (bits 8-15) = 0x00	Clé de connexion de l'esclave (bits 8-15) = nombre aléatoire
10	0x00	0x00
11	0x00	0x00
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

6.1.2 S_CONNECT_CONF

Le maître de sécurité doit indiquer à l'esclave de sécurité l'ID de connexion généré à partir des clés de connexion du maître et de l'esclave.

Le maître de sécurité doit calculer le CRC à partir des clés de connexion du maître et de l'esclave échangées dans la SPDU de la commande S_CONNECT_START, puis doit envoyer ce CRC en tant qu'ID de connexion à l'esclave de sécurité dans la SPDU de la commande S_CONNECT_CONF. Le maître de sécurité doit calculer le CRC à partir du bit de poids faible dans l'ordre de la clé de connexion du maître et de la clé de connexion de l'esclave en utilisant le polynôme CCITT-16 $g_3(x)$ conformément à la Formule (3). La valeur initiale pour ce calcul est 0xFFFF.

$$g_3(x) = x^{16} + x^{12} + x^5 + x^1 \tag{3}$$

Lorsqu'il reçoit la commande S_CONNECT_CONF, l'esclave de sécurité doit calculer le CRC pour sa clé de connexion de maître et sa clé de connexion d'esclave, puis doit le comparer à l'ID de connexion envoyé dans la PDU de la commande S_CONNECT_CONF. Si le CRC calculé est identique à l'ID de connexion, et si les informations d'appareil de l'esclave (ID de fournisseur, code d'appareil) reçues dans la commande S_CONNECT_CONF sont identiques à ses propres informations, l'esclave de sécurité doit envoyer au maître de sécurité la SPDU de l'esclave de sécurité. Si ces valeurs sont différentes, l'esclave de sécurité doit envoyer la commande S_DISCONNECT au maître de sécurité.

Le Tableau 7 répertorie les données de la commande S_CONNECT_CONF. Si la longueur de la SPDU n'est pas suffisante pour stocker l'ensemble de données complet, le maître de sécurité doit fragmenter les données en plusieurs SPDU et transmettre les SPDU fragmentées.

Tableau 7 – Données de la commande S_CONNECT_CONF

octet	Nom	Description
0	ID de connexion (bits 0-7)	Le maître de sécurité définit le Connection_Id. L'esclave de sécurité définit l'ID de connexion envoyé par le maître de sécurité. Cette valeur est utilisée pour authentifier la connexion de sécurité dans le maître de sécurité et l'esclave de sécurité.
1	ID de connexion (bits 8-15)	
2	Adresse de nœud de l'esclave (bits 0-7)	Le maître de sécurité définit l'adresse de nœud de l'esclave enregistrée dans le paramètre de configuration. L'esclave de sécurité définit sa propre adresse de nœud. Cette valeur est utilisée pour confirmer que l'adresse est conforme à l'adresse de nœud dans l'esclave de sécurité.
3	Adresse de nœud de l'esclave (bits 8-15)	
4	ID de fournisseur de l'esclave (bits 0-7)	Le maître de sécurité définit l'ID de fournisseur de l'esclave enregistré dans le paramètre de configuration. L'esclave de sécurité définit la valeur envoyée par le maître de sécurité. Cette valeur est utilisée pour vérifier les informations relatives à l'esclave dans l'esclave de sécurité.
5	ID de fournisseur de l'esclave (bits 8-15)	
6	ID de fournisseur de l'esclave (bits 16-23)	
7	ID de fournisseur de l'esclave (bits 24-31)	
8	Code d'appareil de l'esclave (bits 0-7)	Le maître de sécurité définit le code d'appareil de l'esclave enregistré dans le paramètre de configuration. L'esclave de sécurité définit la valeur envoyée par le maître de sécurité. Cette valeur est utilisée pour vérifier les informations relatives à l'esclave dans l'esclave de sécurité.
9	Code d'appareil de l'esclave (bits 8-15)	
10	Code d'appareil de l'esclave (bits 16-23)	
11	Code d'appareil de l'esclave (bits 24-31)	

Le Tableau 8, le Tableau 9 et le Tableau 10 correspondent à l'exemple d'une SPDU de 16 octets, avec des données fragmentées en 3 SPDU.

Tableau 8 – SPDU de la commande S_CONNECT_CONF (1re SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_CONNECT_CONF = 0x02	S_CONNECT_CONF = 0x02
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
9	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
10	Adresse de nœud de l'esclave (bits 0-7)	Adresse de nœud de l'esclave (bits 0-7)
11	Adresse de nœud de l'esclave (bits 8-15)	Adresse de nœud de l'esclave (bits 8-15)
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

Tableau 9 – SPDU de la commande S_CONNECT_CONF (2e SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_CONNECT_CONF = 0x02	S_CONNECT_CONF = 0x02
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	ID de fournisseur de l'esclave (bits 0-7)	ID de fournisseur de l'esclave (bits 0-7)
9	ID de fournisseur de l'esclave (bits 8-15)	ID de fournisseur de l'esclave (bits 8-15)
10	ID de fournisseur de l'esclave (bits 16-23)	ID de fournisseur de l'esclave (bits 16-23)
11	ID de fournisseur de l'esclave (bits 24-31)	ID de fournisseur de l'esclave (bits 24-31)
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

Tableau 10 – SPDU de la commande S_CONNECT_CONF (3e SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_CONNECT_CONF = 0x02	S_CONNECT_CONF = 0x02
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
8	Code d'appareil de l'esclave (bits 0-7)	Code d'appareil de l'esclave (bits 0-7)
9	Code d'appareil de l'esclave (bits 8-15)	Code d'appareil de l'esclave (bits 8-15)
10	Code d'appareil de l'esclave (bits 16-23)	Code d'appareil de l'esclave (bits 16-23)
11	Code d'appareil de l'esclave (bits 24-31)	Code d'appareil de l'esclave (bits 24-31)
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

6.1.3 S_PRM_SET

Le maître de sécurité utilise la commande S_PRM_SET pour envoyer les paramètres à l'esclave de sécurité. Le Tableau 11 répertorie les données de cette commande. Si la longueur de la SPDU n'est pas suffisante pour stocker l'ensemble de données complet, le maître de sécurité doit fragmenter les données en plusieurs SPDU et transmettre les SPDU fragmentées.

Tableau 11 – Données de la commande S_PRM_SET

octet	Nom	Description
0	Taille du paramètre de communication (bits 0-7)	Le maître de sécurité définit la taille du paramètre de communication. L'esclave de sécurité définit la valeur qu'il a reçue du maître de sécurité.
1	Taille du paramètre de communication (bits 8-15)	
2	Réservé = 0x00	
3	Réservé = 0x00	
4	Temps de fonctionnement du chien de garde (bits 0-7)	Le maître de sécurité définit le temps de fonctionnement du chien de garde enregistré dans le paramètre de configuration. L'esclave de sécurité définit la valeur qu'il a reçue du maître de sécurité.
5	Temps de fonctionnement du chien de garde (bits 8-15)	
6	Temps de fonctionnement du chien de garde (bits 16-23)	
7	Temps de fonctionnement du chien de garde (bits 24-31)	
8	Longueur des données utilisateur de sortie (bits 0-7)	Le maître de sécurité définit la longueur des données utilisateur de sortie calculée dans la formule suivante. (longueur des données utilisateur de sortie) = {(longueur des données de sortie enregistrée dans le paramètre de configuration) − (longueur des données utilisateur de sortie non relatives à la sécurité)} / 2 − 12 L'esclave de sécurité définit la valeur qu'il a reçue du maître de sécurité.
9	Longueur des données utilisateur de sortie (bits 8-15)	
10	Longueur des données utilisateur d'entrée (bits 0-7)	

octet	Nom	Description
11	Longueur des données utilisateur d'entrée (bits 8-15)	Le maître de sécurité définit la longueur des données utilisateur d'entrée calculée dans la formule suivante. (longueur des données utilisateur d'entrée) = {(longueur des données d'entrée enregistrée dans le paramètre de configuration) − (longueur des données utilisateur d'entrée non relatives à la sécurité)} / 2 − 12 L'esclave de sécurité définit la valeur qu'il a reçue du maître de sécurité.

Le Tableau 12, le Tableau 13 et le Tableau 14 correspondent à l'exemple d'une SPDU de 16 octets. Dans cet exemple, les données sont fragmentées en 3 SPDU. Les données sont encapsulées en commençant par l'octet de poids faible et l'espace à la fin doit être rempli avec la valeur 0x00 (voir les octets 10 et 11 de l'exemple).

Tableau 12 – SPDU de la commande S_PRM_SET (1re SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_PRM_SET = 0x04	S_PRM_SET = 0x04
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Taille du paramètre de communication (bits 0-7)	Taille du paramètre de communication (bits 0-7)
9	Taille du paramètre de communication (bits 8-15)	Taille du paramètre de communication (bits 8-15)
10	Réservé = 0x00	Réservé = 0x00
11	Réservé = 0x00	Réservé = 0x00
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

Tableau 13 – SPDU de la commande S_PRM_SET (2e SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_PRM_SET = 0x04	S_PRM_SET = 0x04
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Temps de fonctionnement du chien de garde (bits 0-7)	Temps de fonctionnement du chien de garde (bits 0-7)
9	Temps de fonctionnement du chien de garde (bits 8-15)	Temps de fonctionnement du chien de garde (bits 8-15)
10	Temps de fonctionnement du chien de garde (bits 16-23)	Temps de fonctionnement du chien de garde (bits 16-23)
11	Temps de fonctionnement du chien de garde (bits 24-31)	Temps de fonctionnement du chien de garde (bits 24-31)
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

Tableau 14 – SPDU de la commande S_PRM_SET (3e SPDU)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_PRM_SET = 0x04	S_PRM_SET = 0x04
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Longueur des données utilisateur de sortie (bits 0-7)	Longueur des données utilisateur de sortie (bits 0-7)
9	Longueur des données utilisateur de sortie (bits 8-15)	Longueur des données utilisateur de sortie (bits 8-15)
10	Longueur des données utilisateur d'entrée (bits 0-7)	Longueur des données utilisateur d'entrée (bits 0-7)
11	Longueur des données utilisateur d'entrée (bits 8-15)	Longueur des données utilisateur d'entrée (bits 8-15)
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

6.1.4 S_PRM_APPLY

Le maître de sécurité utilise la commande S_PRM_APPLY pour envoyer le CRC de paramètre à l'esclave de sécurité et pour que celui-ci applique le paramètre.

L'esclave de sécurité doit calculer le CRC pour la commande S_PRM_SET reçue et ne doit pas appliquer le paramètre si le CRC calculé et le CRC de paramètre de cette commande ne sont pas identiques.

Le Tableau 15 répertorie les données de cette commande. Si la longueur de la SPDU n'est pas suffisante pour stocker l'ensemble de données complet, le maître de sécurité doit fragmenter les données en plusieurs SPDU et transmettre les SPDU fragmentées.

Tableau 15 – Données de la commande S_PRM_APPLY

octet	Nom	Description
0	CRC de paramètre (bits 0-7)	Le polynôme 0x90022004 et la valeur de départ 0xFFFFFFFFFFFFFFFF sont utilisés pour calculer le CRC; le calcul est effectué à partir du bit de poids faible dans l'ordre suivant. (1) Temps de fonctionnement du chien de garde (2) Longueur des données utilisateur de sortie (3) Longueur des données utilisateur Le maître de sécurité calcule le CRC à partir des données utilisées pour la génération de la commande S_PRM_SET conformément au Tableau 11. L'esclave de sécurité calcule le CRC à partir des données de la commande S_PRM_SET envoyée par le maître de sécurité conformément au Tableau 11.
1	CRC de paramètre (bits 8-15)	
2	CRC de paramètre (bits 16-23)	
3	CRC de paramètre (bits 24-31)	

Le Tableau 16 correspond à l'exemple d'une SPDU de 16 octets.

Tableau 16 – SPDU de la commande S_PRM_APPLY

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_PRM_APPLY = 0x05	S_PRM_APPLY = 0x05
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	CRC de paramètre (bits 0-7)	CRC de paramètre (bits 0-7)
9	CRC de paramètre (bits 8-15)	CRC de paramètre (bits 8-15)
10	CRC de paramètre (bits 16-23)	CRC de paramètre (bits 16-23)
11	CRC de paramètre (bits 24-31)	CRC de paramètre (bits 24-31)
12	CRC (bits 0-7)	CRC (bits 0-7)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

6.1.5 S_SAFE_DATA

Le maître de sécurité utilise la commande S_SAFE_DATA pour envoyer les données de sortie de sécurité à l'esclave de sécurité. L'esclave de sécurité utilise la commande S_SAFE_DATA pour envoyer les données d'entrée de sécurité au maître de sécurité pendant qu'il est à l'état d'envoi/de réception des données de sécurité. La Tableau 17 représente un exemple de SPDU de la commande S_SAFE_DATA.

Les données de sortie de sécurité correspondent aux données de sécurité envoyées par le maître de sécurité à l'esclave de sécurité. Les données d'entrée de sécurité correspondent aux données de sécurité envoyées par l'esclave de sécurité au maître de sécurité.

Tableau 17 – SPDU de la commande S_SAFE_DATA

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_SAFE_DATA = 0x06	S_SAFE_DATA = 0x06
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Données de sortie de sécurité 1	Données d'entrée de sécurité 1
9	Données de sortie de sécurité 2	Données d'entrée de sécurité 2
10	Données de sortie de sécurité 3	Données d'entrée de sécurité 3
11	Données de sortie de sécurité 4	Données d'entrée de sécurité 4
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

6.1.6 S_DISCONNECT

Le maître de sécurité ou l'esclave de sécurité envoie la commande S_DISCONNECT pour couper la connexion de sécurité et réinitialiser les paramètres de connexion. Un maître de sécurité réinitialise les paramètres de connexion dans l'esclave de sécurité en transmettant cette commande.

Le Tableau 18 correspond à l'exemple d'une SPDU de 16 octets. La Tableau 19 répertorie les valeurs utilisées dans la SPDU de la commande S_DISCONNECT.

Tableau 18 – SPDU de la commande S_DISCONNECT

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_DISCONNECT (L) = 0x03	S_DISCONNECT (L) = 0x03
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7) = 0x00	ID de connexion (bits 0-7) = 0x00
3	ID de connexion (bits 8-15) = 0x00	ID de connexion (bits 8-15) = 0x00
4	Numéro de séquence du maître (bits 0-7) = 0x01	Numéro de séquence de l'esclave (bits 0-7) = 0x01
5	Numéro de séquence du maître (bits 8-15) = 0x00	Numéro de séquence de l'esclave (bits 8-15) = 0x00
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	Facteur (voir le Tableau 19)	Facteur (voir le Tableau 19)
9	0x00	0x00
10	0x00	0x00
11	0x00	0x00
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

Tableau 19 – Facteur de la commande S_DISCONNECT

Valeur	Symbole	Description
0x00	(Réservé)	Pour un usage ultérieur.
0x01	Demande reçue	Une demande DISCONNECT envoyée par l'application a été reçue.
0x02	Réception d'une demande envoyée par le maître	Une commande S_DISCONNECT envoyée par le maître de sécurité a été reçue.
0x03-0x0F	(Réservé)	Pour un usage ultérieur.
0x10	Discordance des adresses	L'adresse de l'esclave reçue dans les commandes S_CONNECT_START et S_CONNECT_CONF ne correspond pas à sa propre adresse.
0x11	Discordance des clés de connexion	La clé de connexion du maître reçue dans la commande S_CONNECT_START envoyée par un esclave de sécurité ne correspond pas à la clé générée par le maître de sécurité. Ce facteur est stocké uniquement dans la SPDU envoyée par un esclave de sécurité.
0x12	Discordance des ID de connexion	L'ID de connexion reçu dans la commande S_CONNECT_CONF envoyée par un esclave de sécurité ne correspond pas à l'ID généré par le maître de sécurité. Ce facteur est stocké uniquement dans la SPDU envoyée par un esclave de sécurité.

Valeur	Symbole	Description
0x13	Discordance des informations d'appareil	Les informations d'appareil (adresse de nœud de l'esclave, ID de fournisseur de l'esclave, code d'appareil de l'esclave) reçues dans la commande S_PRM_SET ne correspondent pas à ses propres informations d'appareil.
0x14	Discordance des paramètres	Le paramètre de sécurité reçu dans la commande S_PRM_SET ne correspond pas à son propre paramètre de sécurité.
0x15	Discordance du CRC de paramètre	Le CRC de paramètre dans la commande S_PRM_APPLY ne correspond pas au CRC de paramètre qu'il a calculé.
0x16-0x1F	(Réservé)	Pour un usage ultérieur.
0x20	Commande non valide	La commande de la SPDU reçue n'est pas valide.
0x21	ID de connexion non valide	L'ID de connexion de la SPDU reçue n'est pas valide.
0x22	Numéro de séquence non valide	Le numéro de séquence de la SPDU reçue n'est pas valide.
0x23	État non valide	L'état de la SPDU reçue n'est pas valide.
0x24	Erreur de CRC	Le CRC de la SPDU reçue ne correspond pas au résultat de calcul.
0x25	Discordance des données	Les données de la SPDU reçue, à l'exclusion du CRC, ne correspondent pas entre le Bloc 1 et le Bloc 2.
0x26	Erreur de contre-vérification	Le résultat de la contre-vérification est NG.
0x27	Erreur de temporisateur de réponse	Une SPDU normale n'a pas été reçue pendant le temps de surveillance de réponse.
0x28-0xFF	(Réservé)	Pour un usage ultérieur.

Pour le calcul du CRC de cette commande, la valeur initiale 0x0001 doit être utilisée pour le numéro de séquence du maître et le numéro de séquence de l'esclave, et la valeur initiale 0x00000000 doit être utilisée pour le numéro de séquence étendu du maître et le numéro de séquence étendu de l'esclave.

6.1.7 S_FAIL_SAFE

Après le passage à l'état de sécurité intrinsèque, le maître de sécurité et l'esclave de sécurité doivent envoyer la commande S_FAIL_SAFE. Le maître de sécurité et l'esclave de sécurité ne doivent pas exécuter de traitement pour cette commande. L'ID de connexion, le numéro de séquence du maître, le numéro de séquence de l'esclave et le numéro d'état conservent la valeur qu'ils avaient immédiatement avant le passage à l'état de sécurité intrinsèque. Le Tableau 20 correspond à l'exemple d'une SPDU de 16 octets.

Tableau 20 – SPDU de la commande S_FAIL_SAFE

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_FAIL_SAFE = 0x07	S_FAIL_SAFE = 0x07
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7)	ID de connexion (bits 0-7)
3	ID de connexion (bits 8-15)	ID de connexion (bits 8-15)
4	Numéro de séquence du maître (bits 0-7)	Numéro de séquence de l'esclave (bits 0-7)
5	Numéro de séquence du maître (bits 8-15)	Numéro de séquence de l'esclave (bits 8-15)

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	0x00	0x00
9	0x00	0x00
10	0x00	0x00
11	0x00	0x00
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

6.1.8 S_NOP

Le maître de sécurité et l'esclave de sécurité ne doivent pas traiter cette commande. Le Tableau 21 correspond à l'exemple d'une SPDU de 16 octets.

Tableau 21 – SPDU de la commande S_NOP

octet	SPDU du maître de sécurité (maître de sécurité → esclave de sécurité)	SPDU de l'esclave de sécurité (esclave de sécurité → maître de sécurité)
0	S_NOP = 0x00	S_NOP = 0x00
1	Réservé = 0x00	Réservé = 0x00
2	ID de connexion (bits 0-7) = 0x00	ID de connexion (bits 0-7) = 0x00
3	ID de connexion (bits 8-15) = 0x00	ID de connexion (bits 8-15) = 0x00
4	Numéro de séquence du maître (bits 0-7) = 0x00	Numéro de séquence de l'esclave (bits 0-7) = 0x00
5	Numéro de séquence du maître (bits 8-15) = 0x00	Numéro de séquence de l'esclave (bits 8-15) = 0x00
6	Numéro d'état	Numéro d'état
7	Réservé = 0x00	Réservé = 0x00
8	0x00	0x00
9	0x00	0x00
10	0x00	0x00
11	0x00	0x00
12	CRC (bits 0-7)	CRC (bits 0-7)
13	CRC (bits 8-15)	CRC (bits 8-15)
14	CRC (bits 16-23)	CRC (bits 16-23)
15	CRC (bits 24-31)	CRC (bits 24-31)

7 Protocole SCL

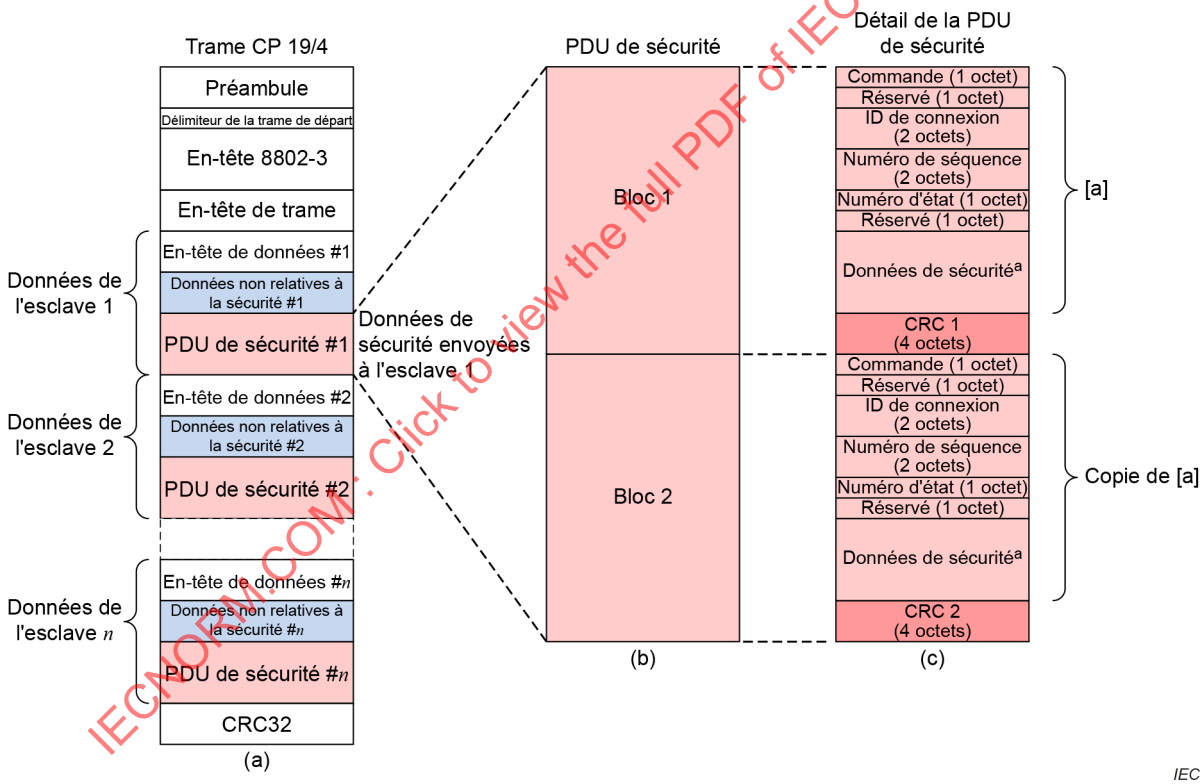
7.1 Format de la SPDU

7.1.1 Structure de la SPDU

La Figure 11 a) représente un exemple de PDU CP 19/4 qui comporte des SPDU FSCP 19 dans la charge utile. La longueur de la PDU CP 19/4 est fixe, mais la longueur des données non relatives à la sécurité peut être configurée comme un pourcentage de la longueur de la SPDU. La valeur zéro pour cent n'est pas admise.

La Figure 11 b), c) représente le format de la SPDU. Une SPDU est constituée de deux blocs (Bloc 1 et Bloc 2) et les données des deux blocs doivent être identiques, à l'exception de la valeur de CRC. Les polynômes générateurs de CRC doivent être uniques au sein des éléments suivants: Bloc 1, Bloc 2 et PDU CP 19/4 (voir le 7.1.6 pour une description des polynômes générateurs de CRC).

La taille de la SPDU, en octets, est configurable par incréments de 8 octets, dans la plage comprise entre 32 et 1488. Par conséquent, chaque taille de bloc, en octets, est configurable par incréments de 4 octets, dans la plage comprise entre 16 et 744.



[a] La taille des données de sécurité est limitée par la FAL. Le CP19/4 repose sur l'ISO/IEC/IEEE 8802-3 qui limite les données de sécurité à 732 octets.

Figure 11 – Format de la PDU de sécurité

La relation entre la taille des données utilisateur à accepter par le point d'accès de service de la FAL et la taille maximale des données utilisateur de sécurité en octets est calculée par la Formule (4).

$$\text{longueur\_maximale\_données\_utilisateur\_sécurité} = \frac{\text{longueur\_données\_utilisateur\_acceptée\_par\_la\_FAL}}{2} - 12 \quad (4)$$

7.1.2 ID de connexion

L'ID de connexion est utilisé pour l'authentification par le maître de sécurité et l'esclave de sécurité. Lors de l'établissement de la connexion de sécurité, le maître de sécurité et l'esclave de sécurité échangent leurs clés de connexion. Le maître de sécurité génère l'ID de connexion à partir des clés de connexion, et confirme que cet ID est unique. Après cette confirmation, le maître de sécurité envoie l'ID de connexion à l'esclave de sécurité. Pour plus d'informations, voir le 7.3.1.2.

7.1.3 Numéro de séquence

Lorsqu'une SPDU est envoyée, le maître de sécurité et l'esclave de sécurité ajoutent un numéro de séquence différent à la SPDU. Pour plus d'informations, voir le 5.3.2.

7.1.4 Commande

La Tableau 22 répertorie les commandes et leurs codes de commande associés. Pour plus d'informations, voir le 6.1.

Tableau 22 – Liste des commandes

Code	Commande	Description
0x0000	S_NOP	Aucune opération. Le maître de sécurité et les esclaves de sécurité n'effectuent aucune opération lorsqu'ils reçoivent cette commande.
0x0001	S_CONNECT_START	Demande de démarrage de l'établissement de la connexion. Le maître de sécurité envoie la clé de connexion du maître à l'esclave de sécurité. En réponse, l'esclave de sécurité envoie la clé de connexion de l'esclave au maître de sécurité. Le maître de sécurité génère l'ID de connexion à partir des clés de connexion du maître et de l'esclave.
0x0002	S_CONNECT_CONF	Demande de confirmation de l'établissement de la connexion. Si l'ID de connexion généré par le maître de sécurité est valide, le maître de sécurité envoie cette commande à l'esclave de sécurité et établit une connexion.
0x0003	S_DISCONNECT	Demande de coupure de la connexion. La connexion établie est coupée.
0x0004	S_PRM_SET	Demande de réglage des paramètres. Le maître de sécurité envoie les paramètres de communication et d'application à l'esclave de sécurité.
0x0005	S_PRM_APPLY	Demande d'application des paramètres. Les paramètres envoyés dans la commande S_PRM_SET sont appliqués par le maître de sécurité à l'esclave de sécurité.
0x0006	S_SAFE_DATA	Demande d'envoi des données de sécurité. Le maître de sécurité envoie les données de sortie de sécurité à l'esclave de sécurité. En réponse, l'esclave de sécurité envoie les données d'entrée de sécurité au maître de sécurité.
0x0007	S_FAIL_SAFE	Notification de l'état de sécurité intrinsèque. Cette commande est une notification précisant qu'une SCL est passée à l'état de sécurité intrinsèque. Le maître de sécurité et les esclaves de sécurité n'effectuent aucune opération en réponse lorsqu'ils reçoivent cette commande.

7.1.5 Numéro d'état

Ce nombre indique l'état actuel de la SCL. À l'état de sécurité intrinsèque, le numéro d'état conserve la valeur qu'il avait immédiatement avant le passage à l'état de sécurité intrinsèque. Pour plus d'informations sur le numéro d'état, voir le 7.2.

7.1.6 CRC

Il existe un CRC 32 bits pour chaque bloc. Pour plus d'informations, voir le 5.3.5.

7.1.7 Données redondantes

Lors de la transmission d'une SPDU, l'ensemble de données complet (à l'exception du CRC) est dupliqué à des fins de redondance. Chaque copie des données est vérifiée avec un CRC différent dans la SPDU.

Pour plus d'informations, voir le 5.3.6.1 et le 5.3.6.2.

7.2 Machine de protocole de service FAL de sécurité

7.2.1 Transition d'état du maître de sécurité

7.2.1.1 Couche de communication de sécurité

Pour la SCL du maître de sécurité, le diagramme de transition d'état est représenté à la Figure 12, les états sont décrits dans le Tableau 23 et la matrice de transition d'état est expliquée dans le Tableau 24.

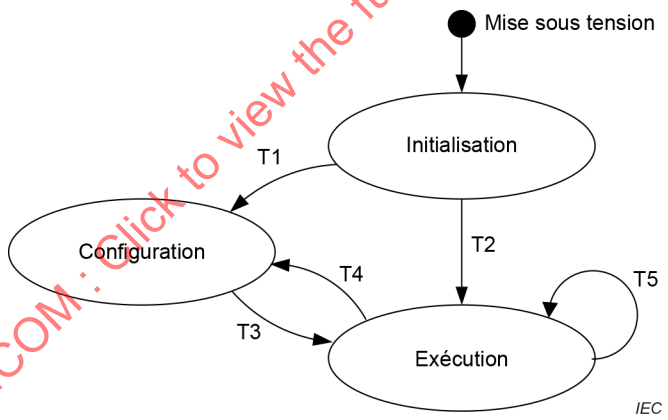


Figure 12 – SCL du maître de sécurité – diagramme de transition d'état

Tableau 23 – SCL du maître de sécurité – description des états

État	Description
Initialisation	Le traitement initial est en cours d'exécution.
Configuration	La configuration est en cours d'exécution. Lorsque la configuration est achevée, la SCL passe à l'état d'exécution.
Exécution	Les connexions entre les esclaves de sécurité peuvent être établies. Pour la connexion de sécurité, le diagramme de transition d'état est représenté à la Figure 13.

Tableau 24 – SCL du maître de sécurité – matrice de transition d'état

État	Transition	Condition	Action	État de destination
Initialisation	T1	L'initialisation est achevée. && Une demande de configuration a été reçue.	Initialiser les variables relatives à la connexion.	Configuration
Initialisation	T2	L'initialisation est achevée. && Aucune demande de configuration n'a été reçue.	Initialiser les variables relatives à la connexion.	Exécution
Configuration	T3	La configuration est achevée.	-	Exécution
Exécution	T4	La configuration a démarré. && Tous les modules sont à l'état de repos.	-	Configuration
Exécution	T5	La configuration a démarré. && Certains modules NE sont PAS à l'état de repos.	-	Exécution

7.2.1.2 Connexion de sécurité

Pour la connexion de sécurité du maître de sécurité, le diagramme de transition d'état est représenté à la Figure 13, les états sont décrits dans le Tableau 25 et la matrice de transition d'état est expliquée dans le Tableau 26.

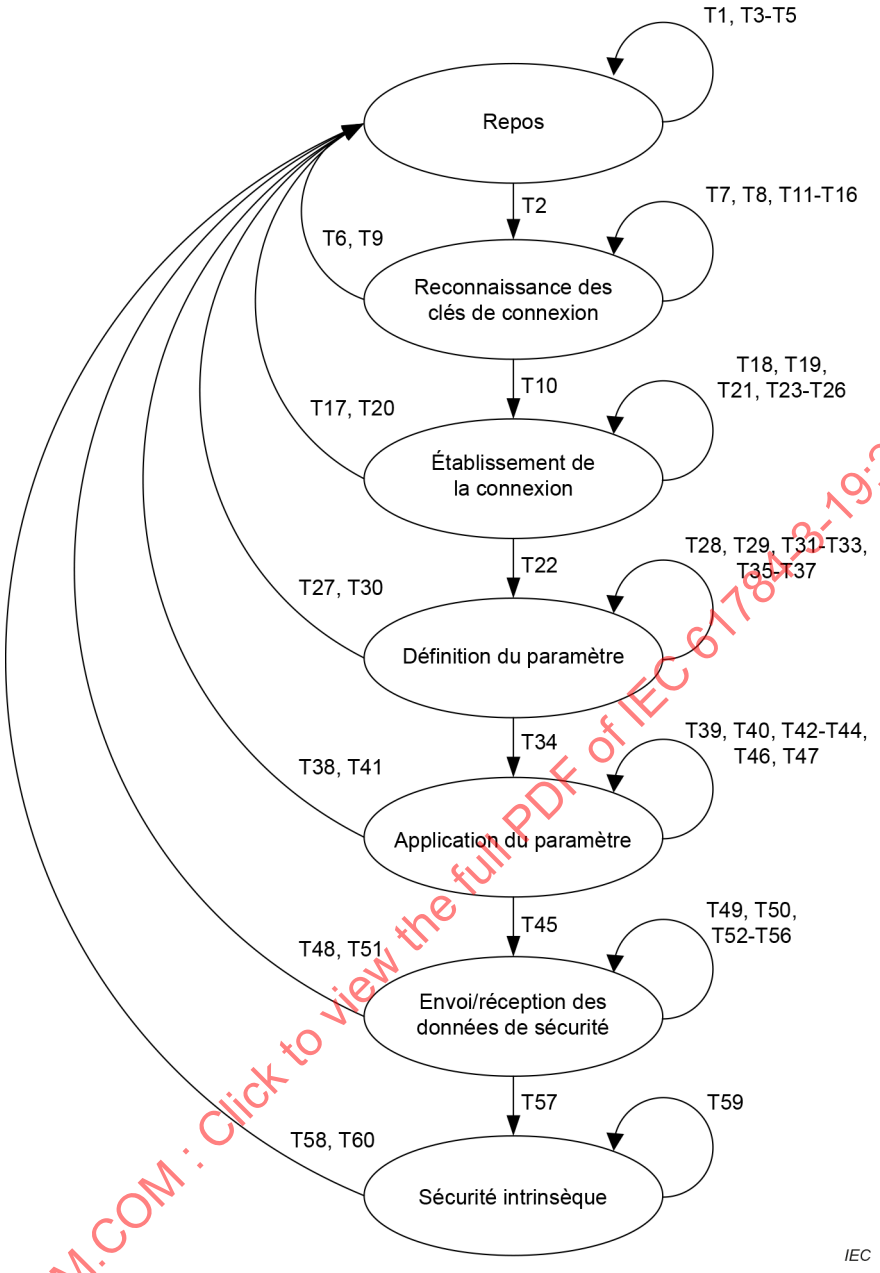


Figure 13 – Connexion de sécurité du maître de sécurité – diagramme de transition d'état

Tableau 25 – Connexion de sécurité du maître de sécurité – description des états

État	Description
Repos	Dans cet état, la communication de sécurité n'est pas exécutée et aucune connexion de sécurité n'est établie. Une instance de connexion attend la demande de démarrage/redémarrage de la communication de sécurité par l'utilisateur. Une porte est mise en place pour protéger le système contre une reprise automatique imprévue et assurer un "démarrage" en toute sécurité en cas de fonctionnement normal ou d'achèvement du paramétrage initial, ou un "redémarrage" après la détection et/ou l'élimination d'anomalies vérifiées par l'utilisateur.
Reconnaissance des clés de connexion	L'instance de connexion échange la clé de connexion avec l'esclave de sécurité.
Établissement de la connexion	L'instance de connexion établit la connexion avec l'esclave de sécurité.
Définition du paramètre	L'instance de connexion définit le paramètre d'esclave sur l'esclave de sécurité.
Application du paramètre	L'instance de connexion demande l'application du paramètre d'esclave sur l'esclave de sécurité.
Envoi/réception des données de sécurité	L'instance de connexion envoie et reçoit les données de sécurité par l'intermédiaire de l'esclave de sécurité.
Sécurité intrinsèque	L'instance de connexion envoie la commande S_FAIL_SAFE. La commande S_DISCONNECT doit être envoyée par l'esclave de sécurité, ou l'utilisateur doit intervenir pour déclencher le passage à l'état de repos.
NOTE Le maître de sécurité occupe cet état pour chaque connexion avec différents esclaves de sécurité.	

Tableau 26 – Connexion de sécurité du maître de sécurité – matrice de transition d'état

État	Transition	Condition	Action	État de destination
Repos	T1	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos
Repos	T2	Demande de démarrage de la communication de sécurité reçue.	Initialiser les variables relatives à la connexion. Envoyer la commande S_CONNECT_START. Démarrer le temporisateur de réponse.	Reconnaissance des clés de connexion
Repos	T3	Demande d'arrêt de la communication de sécurité reçue.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Repos
Repos	T4	Commande S_DISCONNECT reçue.	Initialiser les variables relatives à la connexion. Arrêter le temporisateur de réponse.	Repos
Repos	T5	Expiration du temporisateur de réponse.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Repos
Reconnaissance des clés de connexion	T6	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos

État	Transition	Condition	Action	État de destination
Reconnaissance des clés de connexion	T7	Demande d'arrêt de la communication de sécurité reçue.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T8	SPDU non valide reçue. ID de connexion non valide. Numéro de séquence non valide. CRC non valide. Commande non valide. Données non valides.	Signaler l'erreur à l'application.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T9	Commande S_DISCONNECT reçue.	Initialiser les variables relatives à la connexion. Arrêter le temporisateur de réponse.	Repos
Reconnaissance des clés de connexion	T10	Commande S_CONNECT_START reçue. && L'ID de connexion généré n'est PAS utilisé.	Envoyer la commande S_CONNECT_CONF. Démarrer le temporisateur de réponse.	Établissement de la connexion
Reconnaissance des clés de connexion	T11	Commande S_CONNECT_START reçue. && L'ID de connexion généré est déjà utilisé.	Envoyer la commande S_CONNECT_START. Démarrer le temporisateur de réponse.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T12	Commande S_CONNECT_CONF reçue.	Signaler l'erreur à l'application.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T13	Commande S_PRM_SET reçue.	Signaler l'erreur à l'application.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T14	Commande S_PRM_APPLY reçue.	Signaler l'erreur à l'application.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T15	Commande S_SAFE_DATA reçue.	Signaler l'erreur à l'application.	Reconnaissance des clés de connexion
Reconnaissance des clés de connexion	T16	Expiration du temporisateur de réponse.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Reconnaissance des clés de connexion
Établissement de la connexion	T17	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos
Établissement de la connexion	T18	Demande d'arrêt de la communication de sécurité reçue.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Établissement de la connexion

État	Transition	Condition	Action	État de destination
Établissement de la connexion	T19	SPDU non valide reçue. ID de connexion non valide. Numéro de séquence non valide. CRC non valide. Commande non valide. Données non valides.	Signaler l'erreur à l'application.	Établissement de la connexion
Établissement de la connexion	T20	Commande S_DISCONNECT reçue.	Initialiser les variables relatives à la connexion. Arrêter le temporisateur de réponse.	Repos
Établissement de la connexion	T21	Commande S_CONNECT_START reçue.	Signaler l'erreur à l'application.	Établissement de la connexion
Établissement de la connexion	T22	Commande S_CONNECT_CONF reçue.	Envoyer la commande S_PRM_SET. Démarrer le temporisateur de réponse.	Définition du paramètre
Établissement de la connexion	T23	Commande S_PRM_SET reçue.	Signaler l'erreur à l'application.	Établissement de la connexion
Établissement de la connexion	T24	Commande S_PRM_APPLY reçue.	Signaler l'erreur à l'application.	Établissement de la connexion
Établissement de la connexion	T25	Commande S_SAFE_DATA reçue.	Signaler l'erreur à l'application.	Établissement de la connexion
Établissement de la connexion	T26	Expiration du temporisateur de réponse.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Établissement de la connexion
Définition du paramètre	T27	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos
Définition du paramètre	T28	Demande d'arrêt de la communication de sécurité reçue.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Définition du paramètre
Définition du paramètre	T29	SPDU non valide reçue. ID de connexion non valide. Numéro de séquence non valide. CRC non valide. Commande non valide. Données non valides.	Signaler l'erreur à l'application.	Définition du paramètre
Définition du paramètre	T30	Commande S_DISCONNECT reçue.	Initialiser les variables relatives à la connexion. Arrêter le temporisateur de réponse.	Repos
Définition du paramètre	T31	Commande S_CONNECT_START reçue.	Signaler l'erreur à l'application.	Définition du paramètre
Définition du paramètre	T32	Commande S_CONNECT_CONF reçue.	Signaler l'erreur à l'application.	Définition du paramètre

État	Transition	Condition	Action	État de destination
Définition du paramètre	T33	Commande S_PRM_SET reçue. && Toutes les données de paramètre n'ont PAS été transmises.	Envoyer la commande S_PRM_SET. Démarrer le temporisateur de réponse.	Définition du paramètre
Définition du paramètre	T34	Commande S_PRM_SET reçue. && Toutes les données de paramètre ont été transmises.	Envoyer la commande S_PRM_APPLY. Démarrer le temporisateur de réponse.	Application du paramètre
Définition du paramètre	T35	Commande S_PRM_APPLY reçue.	Signaler l'erreur à l'application.	Définition du paramètre
Définition du paramètre	T36	Commande S_SAFE_DATA reçue.	Signaler l'erreur à l'application.	Définition du paramètre
Définition du paramètre	T37	Expiration du temporisateur de réponse.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Définition du paramètre
Application du paramètre	T38	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos
Application du paramètre	T39	Demande d'arrêt de la communication de sécurité reçue.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Application du paramètre
Application du paramètre	T40	SPDU non valide reçue. ID de connexion non valide. Numéro de séquence non valide. CRC non valide. Commande non valide. Données non valides.	Signaler l'erreur à l'application.	Application du paramètre
Application du paramètre	T41	Commande S_DISCONNECT reçue.	Initialiser les variables relatives à la connexion. Arrêter le temporisateur de réponse.	Repos
Application du paramètre	T42	Commande S_CONNECT_START reçue.	Signaler l'erreur à l'application.	Application du paramètre
Application du paramètre	T43	Commande S_CONNECT_CONF reçue.	Signaler l'erreur à l'application.	Application du paramètre
Application du paramètre	T44	Commande S_PRM_SET reçue.	Signaler l'erreur à l'application.	Application du paramètre
Application du paramètre	T45	Commande S_PRM_APPLY reçue.	Envoyer la commande S_SAFE_DATA. Démarrer le temporisateur de chien de garde.	Envoi/réception des données de sécurité
Application du paramètre	T46	Commande S_SAFE_DATA reçue.	Signaler l'erreur à l'application.	Application du paramètre

État	Transition	Condition	Action	État de destination
Application du paramètre	T47	Expiration du temporisateur de réponse.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Application du paramètre
Envoi/réception des données de sécurité	T48	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos
Envoi/réception des données de sécurité	T49	Demande d'arrêt de la communication de sécurité reçue.	Arrêter le temporisateur de chien de garde. Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T50	SPDU non valide reçue. ID de connexion non valide. Numéro de séquence non valide. CRC non valide. Commande non valide. Données non valides.	Signaler l'erreur à l'application.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T51	Commande S_DISCONNECT reçue.	Arrêter le temporisateur de chien de garde. Signaler l'erreur à l'application. Initialiser les variables relatives à la connexion.	Repos
Envoi/réception des données de sécurité	T52	Commande S_CONNECT_START reçue.	Signaler l'erreur à l'application.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T53	Commande S_CONNECT_CONF reçue.	Signaler l'erreur à l'application.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T54	Commande S_PRM_SET reçue.	Signaler l'erreur à l'application.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T55	Commande S_PRM_APPLY reçue.	Signaler l'erreur à l'application.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T56	Commande S_SAFE_DATA reçue.	Arrêter le temporisateur de chien de garde. Fournir les données d'entrée de sécurité à l'application. Envoyer la commande S_SAFE_DATA. Démarrer le temporisateur de chien de garde.	Envoi/réception des données de sécurité
Envoi/réception des données de sécurité	T57	Expiration du temporisateur de chien de garde.	Arrêter le temporisateur de chien de garde. Signaler l'erreur à l'application. Envoyer la commande S_FAIL_SAFE.	Sécurité intrinsèque
Sécurité intrinsèque	T58	Demande de réinitialisation reçue.	Initialiser les variables relatives à la connexion.	Repos

État	Transition	Condition	Action	État de destination
Sécurité intrinsèque	T59	Demande d'arrêt de la communication de sécurité reçue.	Envoyer la commande S_DISCONNECT. Démarrer le temporisateur de réponse.	Sécurité intrinsèque
Sécurité intrinsèque	T60	Commande S_DISCONNECT reçue.	Initialiser les variables relatives à la connexion.	Repos

7.2.2 Transition d'état de l'esclave de sécurité

7.2.2.1 Couche de communication de sécurité

Pour la SCL de l'esclave de sécurité, le diagramme de transition d'état est représenté à la Figure 14, les états sont décrits dans le Tableau 27 et la matrice de transition d'état est expliquée dans le Tableau 28.

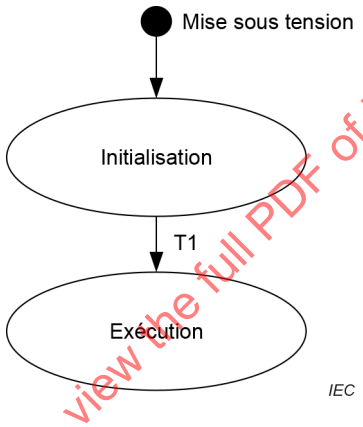


Figure 14 – SCL de l'esclave de sécurité – diagramme de transition d'état

Tableau 27 – SCL de l'esclave de sécurité – description des états

État	Description
Initialisation	Le traitement initial est en cours d'exécution.
Exécution	Les connexions entre les maîtres de sécurité peuvent être établies. Pour la connexion de sécurité, le diagramme de transition d'état est représenté à la Figure 15.

Tableau 28 – SCL de l'esclave de sécurité – matrice de transition d'état

État	Transition	Condition	Action	État de destination
Initialisation	T1	L'initialisation est achevée.	Initialiser les variables relatives à la connexion.	Exécution

7.2.2.2 Connexion de sécurité

Pour la connexion de sécurité de l'esclave de sécurité, le diagramme de transition d'état est représenté à la Figure 15, les états sont décrits dans le Tableau 29 et la matrice de transition d'état est expliquée dans le Tableau 30.

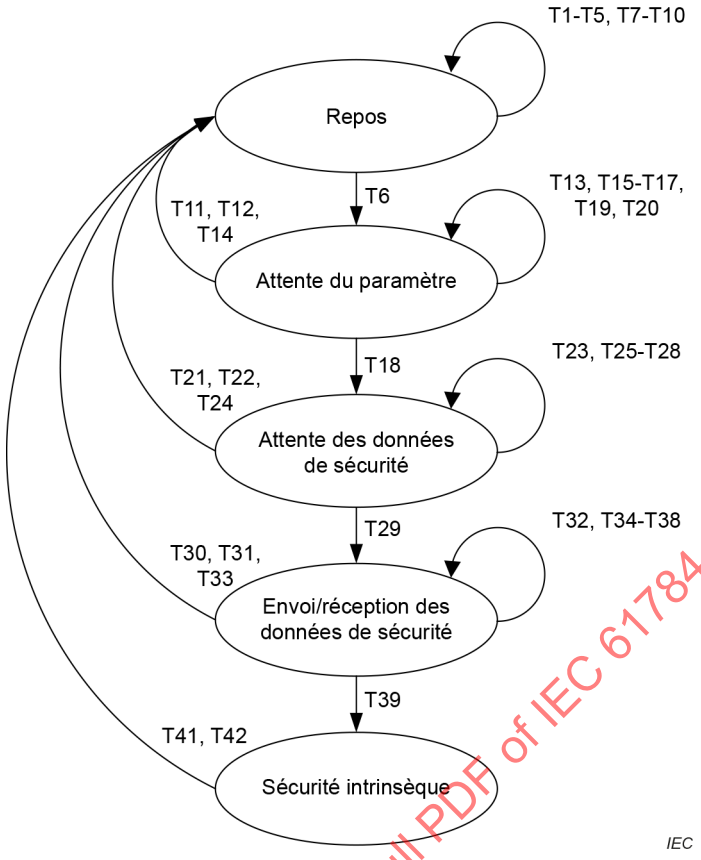


Figure 15 – Connexion de sécurité de l'esclave de sécurité – diagramme de transition d'état

Tableau 29 – Connexion de sécurité de l'esclave de sécurité – description des états

État	Description
Repos	L'instance de connexion attend l'établissement de la connexion par le maître de sécurité.
Attente du paramètre	L'instance de connexion attend la définition du paramètre par le maître de sécurité.
Attente des données de sécurité	L'instance de connexion attend la réception des premières données de sécurité par le maître de sécurité.
Envoi/réception des données de sécurité	L'instance de connexion envoie et reçoit les données de sécurité par l'intermédiaire du maître de sécurité.
Sécurité intrinsèque	L'instance de connexion envoie la commande S_FAIL_SAFE. La commande S_DISCONNECT doit être envoyée par l'esclave de sécurité, ou l'utilisateur doit intervenir pour déclencher le passage à l'état de repos.
NOTE L'esclave de sécurité occupe cet état pour chaque connexion avec différents maîtres de sécurité.	