IO-Link Safety System Extensions

with SMI

Specification

V1.1.3
March 2022

Order No: 10.092

File name: **IO-Link_Safety_System-Extensions_10092_V1.1.3_Mar22.docx**

This specification has been prepared by the IO-Link Safety technology working group, incorporating the final Standardized Master Interface (SMI), and covering CR-IDs up to 117. It is the basis for the international standard IEC 61139-2.

Any comments, proposals, requests on this document are appreciated through the IO-Link CR database www.io-link-projects.com. Please provide name and email address.
Login: *IOL-Safety11*
Password: *Report*

Important notes:

NOTE 1 The IO-Link Community Rules shall be observed prior to the development and marketing of IO-Link products. The document can be downloaded from the www.io-link.com portal.

NOTE 2 Any IO-Link device shall provide an associated IODD file. Easy access to the file and potential updates shall be possible. It is the responsibility of the IO-Link device manufacturer to test the IODD file with the help of the IODD-Checker tool available per download from www.io-link.com.

NOTE 3 Any IO-Link device shall provide an associated manufacturer declaration on the conformity of the device. A corresponding form with references to relevant documents is available per download from www.io-link.com.

Conventions:

In this specification the following key words (in **bold** text) will be used:
shall: indicates a mandatory requirement. Designers **shall** implement such mandatory requirements to ensure interoperability and to claim conformity with this specification.
should: indicates flexibility of choice with a strongly preferred implementation.
can: indicates flexibility of choice with no implied preference (possibility and capability).
may: indicates a permission

Publisher:
IO-Link Community
c/o PROFIBUS Nutzerorganisation e.V.
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 721 / 96 58 590
Fax: +49 721 / 96 58 589
E-mail: info@io-link.com
Web site: www.io-link.com

CONTENTS

INTRODUCTION

The base technology of IO-Link™[1] is subject matter of the international standard IEC 61131-9 being part of a series of standards on programmable controllers and the associated peripherals such as remote I/O (RIO).

It specifies a single-drop digital communication interface technology – named SDCI, which extends the traditional switching input and output interfaces as defined in IEC 61131-2 towards a point-to-point communication link using coded switching. This technology enables the cyclic exchange of digital input and output process data between a Master and its associated Devices (sensors, actuators, I/O terminals, etc.). The Master can be part of a fieldbus communication system or any stand-alone processing unit. The technology also enables the acyclic transfer of parameters to Devices and the propagation of diagnosis information from the Devices to the upper-level automation system (controller, host) via the Master.

Physical topology is point-to-point from each Device to the Master using 3 wires over distances up to 20 m. The SDCI physical interface is backward compatible with the usual 24 V I/O signalling specified in IEC 61131-2 and supports three transmission rates of 4,8 kbit/s, 38,4 kbit/s and 230,4 kbit/s are supported.

The main advantages of the SDCI technology are:

- dual use of either switching signals (DI/DO) or coded switching communication respectively,
- traditional switching sensors and actuators now providing alternatively single drop digital communication within the same Device,
- one thin, robust, very flexible cable without shielding for power supply and signalling,
- lowest-cost digital communication down to the lowest end sensors and actuators.

The functional safety variant of SDCI is called SDCI-FS. Figure 1 shows an example positioning of SDCI-FS in functional safety automation.

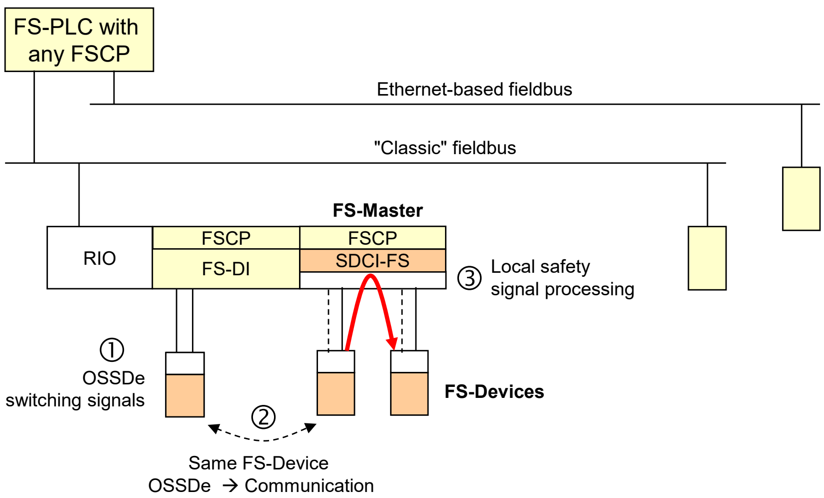


Figure 1 – Positioning of SDCI-FS in functional safety automation

In this example, a remote I/O is connected to a functional safety programmable controller using one of the FSCPs of the IEC 61784-3 series to communicate with an FS-DI module and a gateway to an SDCI-FS FS-Master. FS-Devices with OSSDe can be connected to FS-DIs or FS-Masters. All FS-Devices can communicate with any FS-Master using the SDCI-FS protocol

---

[1] IO-Link™ is a trade name of the "IO-Link Community". This information is given for the convenience of users of this specification and does not constitute an endorsement by the IO-Link Community of the trade name holder or any of its products. Compliance to this document does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Community".

regardless of the upper-level FSCP-system. The same is true for safety actuators (FS-Devices) such as drives with integrated safety. This means the largest component commonality ① for sensors and actuators similar to the DI and DO interfaces standardized within IEC 61131-2.

Safety sensors with OSSDe interfaces – equipped with SDCI-FS communication – can be parameterized via auxiliary tools such as "USB-Masters", then connected to an FS-DI and operated in OSSDe mode. They also can be operated in OSSDe mode on an FS-Master that supports OSSDe. In case these safety sensors are equipped with SDCI-FS communication in addition, they can be operated in both modes ②, either OSSDe or SDCI-FS. This corresponds to the SDCI SIO paradigm.

The concept of SDCI-FS allows for local safety signal processing if the gateway/FS-Master provides a local safety controller ③.

This document provides the necessary extensions to IEC 61131-9 for functional safety communication including standardization of OSSDe and parameterization within the domain of safety for machinery. Figure 2 shows its relationships to international fieldbus and safety standards as well as to relevant specifications (see Clause 2 and bibliography). Any functional safety starts with risk assessment and risk reduction (ISO 12100). One possibility of risk reduction is the usage of electrical or electronic control systems. For the design of those, standards such as IEC 61508, IEC 62061, and ISO 13849 can be used. Environmental conditions such as EMC are covered by for example IEC 61000-6-7. Further aspects are installations and security issues. A number of product standards complement the generic or sector standards.

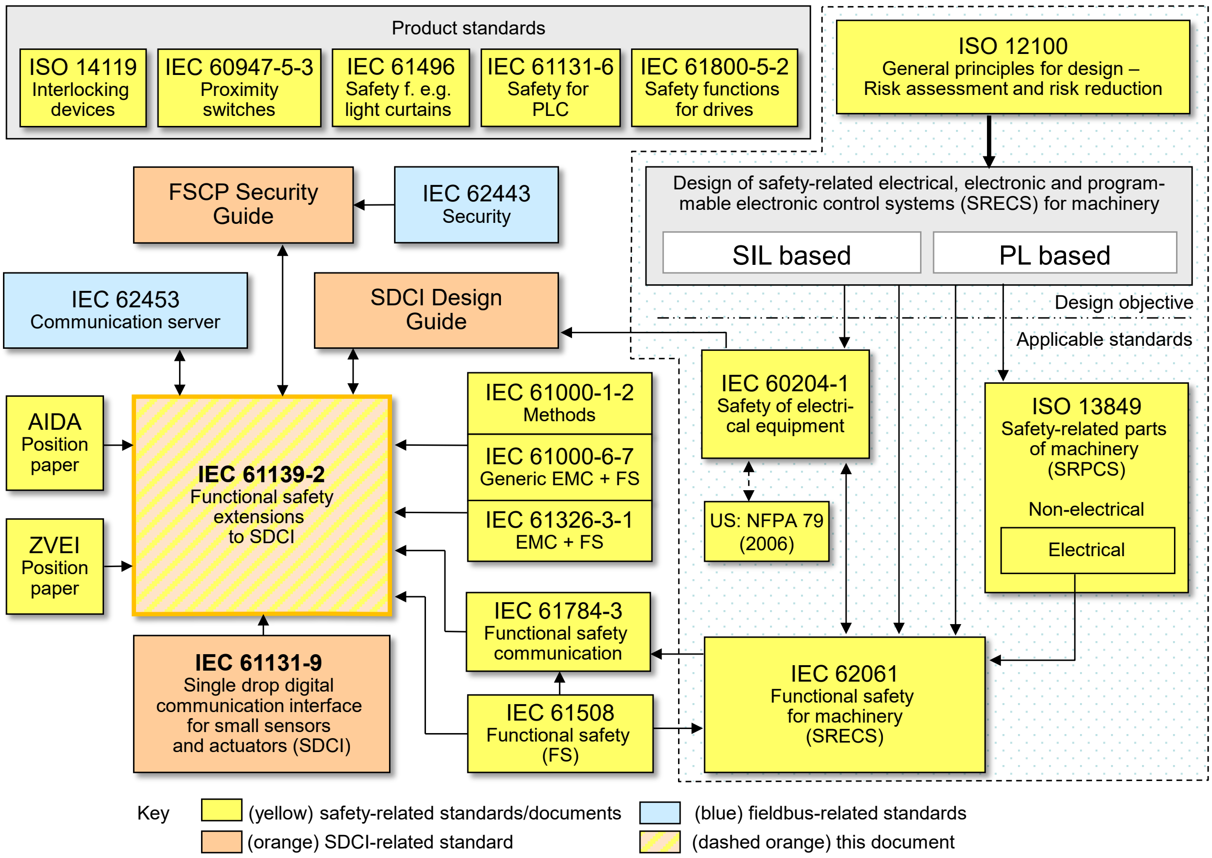


Figure 2 – Relationship of this document to standards

SDCI-FS can be used for functional safety applications according to IEC 62061 and IEC 61508 up to SIL3 and/or according to ISO 13849 up to PLe.

INDUSTRIAL NETWORKS – SINGLE-DROP DIGITAL COMMUNICATION INTERFACE –

Part 2: Functional safety extensions

1 Scope

This part of IEC 61139 specifies the extensions to SDCI in IEC 61131-9 for functional safety. This comprises:

- a standardized OSSDe interface for redundant switching signals based on IEC 61131-2,
- minor modifications/extensions to state machines of SDCI to support the safety operations,
- a lean functional safety communication protocol on top of the standard SDCI communication which is a black channel according to IEC 61784-3:2021,
- protocol management functions for configuration, parameterization, and commissioning,
- IODD extensions for functional safety,
- a Device tool interface to support Dedicated Tools according to functional safety standards.

This document does not cover:

- communication interfaces or systems including multi-point or multi-drop linkages,
- communication interfaces or systems including multi-channel or encrypted linkages,
- wireless communication interfaces or systems,
- integration of SDCI-FS into upper-level systems such as fieldbuses/FSCPs.

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 60204-1, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*

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

IEC 61131-2, *Industrial-process measurement and control – Programmable controllers – Part 2: Equipment requirements and tests*

IEC 61131-9:2022[2], *Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)*

IEC 61496-1, *Safety of machinery – Electro-sensitive protective equipment – Part 1: General requirements and tests*

IEC 61508-3, *Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements*

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

---

2 Under preparation. Stage at the time of publication: IEC/AFDIS 61131-9:2021

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

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

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

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 61131-1 and IEC 61131-2 apply. ISO and IEC maintain terminological databases for use in standardization at the following addresses:

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

3.2 Common terms and definitions

3.2.1
address
part of the M-sequence control to reference data within data categories of a communication channel

3.2.2
application layer
AL
<SDCI>[3] part of the protocol responsible for the transmission of Process Data objects and On-request Data objects

3.2.3
block parameter
consistent parameter access via multiple Indices or Subindices

3.2.4
checksum
<SDCI> complementary part of the overall data integrity measures in the data link layer in addition to the UART parity bit

3.2.5
coded switching
SDCI communication, based on the standard binary signal levels of IEC 61131-2

3.2.6
COM1
SDCI communication mode with transmission rate of 4,8 kbit/s

3.2.7
COM2
SDCI communication mode with transmission rate of 38,4 kbit/s

3.2.8
COM3
SDCI communication mode with transmission rate of 230,4 kbit/s

3.2.9
COMx
one out of three possible SDCI communication modes COM1, COM2, or COM3

---

[3] Angle brackets indicate validity of the definition for the SDCI technology

3.2.10
communication channel
logical connection between Master and Device

NOTE 1 to entry: Four communication channels are defined: process channel, page and ISDU channel (for parameters), and diagnosis channel.

3.2.11
communication error
unexpected disturbance of the SDCI transmission protocol

3.2.12
cycle time
time to transmit an M-sequence between a Master and its Device including the following idle time

3.2.13
Device
single passive peer to a Master such as a sensor or actuator

NOTE 1 to entry: Uppercase "Device" is used for SDCI equipment, while lowercase "device" is used in a generic manner.

3.2.14
Direct Parameter
directly (page) addressed parameter transferred acyclically via the page communication channel without acknowledgement

3.2.15
dynamic parameter
part of a Device's parameter set defined by on-board user interfaces such as teach-in buttons or control panels in addition to the static parameters

3.2.16
Event
instance of a change of conditions in a Device

NOTE 1 to entry: Uppercase "Event" is used for SDCI Events, while lowercase "event" is used in a generic manner.

NOTE 2 to entry: An Event is indicated via the Event flag within the Device's status cyclic information, then acyclic transfer of Event data (typically diagnosis information) is conveyed through the diagnosis communication channel.

3.2.17
fallback
transition of a Port from coded switching to switching signal mode

3.2.18
ISDU
indexed service data unit used for acyclic acknowledged transmission of parameters that can be segmented in a number of M-sequences

3.2.19
M-sequence
sequence of two messages comprising a Master message and its subsequent Device message

3.2.20
M-sequence control
first octet in a Master message indicating the read/write operation, the type of the communication channel, and the address, for example offset or flow control

3.2.21
M-sequence type
one particular M-sequence format out of a set of specified M-sequence formats

3.2.22
Master
active peer connected through Ports to one up to n Devices and which provides an interface to the gateway to the upper-level communication systems or PLCs

NOTE 1 to entry: Uppercase "Master" is used for SDCI equipment, while lowercase "master" is used in a generic manner.

3.2.23
message
<SDCI> sequence of UART frames transferred either from a Master to its Device or vice versa following the rules of the SDCI protocol

3.2.24
On-request Data
acyclically transmitted data upon request of the Master application consisting of parameters or Event data

3.2.25
physical layer
first layer of the ISO-OSI reference model, which provides the mechanical, electrical, functional, and procedural means to activate, maintain, and de-activate physical connections for bit transmission between data-link entities

NOTE 1 to entry: Physical layer also provides means for wake-up and fallback procedures.

[SOURCE: ISO/IEC 7498-1, 7.7.2, modified – text extracted from subclause, note added]

3.2.26
Port
<SDCI> communication medium interface of the Master to one Device

3.2.27
Port operating mode
state of a Master's Port that can be either INACTIVE, DO, DI, FIXEDMODE, or SCANMODE

3.2.28
Process Data
input or output values from or to a discrete or continuous automation process cyclically transferred with high priority and in a configured schedule automatically after start-up of a Master

3.2.29
SIO
Port operation mode in accordance with digital input and output defined in IEC 61131-2 that is established after power-up or fallback or unsuccessful communication attempts

3.2.30
switching signal
binary signal from or to a Device when in SIO mode (as opposed to the "coded switching" SDCI communication)

3.2.31
System Management
SM
<SDCI> means to control and coordinate the internal communication layers and the exceptions within the Master and its Ports, and within each Device

3.2.32
UART frame
<SDCI> bit sequence starting with a start bit, followed by eight bits carrying a data octet, followed by an even parity bit and ending with one stop bit

3.2.33
wake-up
procedure for causing a Device to change its mode from SIO to SDCI

3.2.34
wake-up request
WURQ
physical layer service used by the Master to initiate wake-up of a Device, and put it in a receive ready state

3.3 Terms and definitions related to SDCI-FS

For the purposes of this document, the following additional terms and definitions apply.

3.3.1
error
discrepancy between a computed, observed, or measured value or condition and the true, specified or theoretically correct value or condition

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

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

SOURCE: [IEC 61508-4:2010, 3.6.11, modified – The notes have been added.]

3.3.2
failure
termination of the ability of a functional unit to perform a required function or operation of a functional unit in any way other than as required

NOTE 1 to entry: The definition in IEC 61508-4 is the same, with additional notes.

NOTE 2 to entry: Failure may be due to an error (for example, problem with hardware/software design or message disruption)

SOURCE: [IEC 61508-4:2010, 3.6.4, modified – The notes have been removed and replaced by a new note and the figure has been deleted.]

3.3.3
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: IEV 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, modified – The reference to Figure 4 of IEC 61508-4:2010 in the note has been removed.]

3.3.4
FS-Device
single passive peer such as a functional safety sensor or actuator to a Master with functional safety capabilities

3.3.5
FS-Master
active peer with functional safety capabilities connected through Ports to one up to *n* Devices or FS-Devices and which provides an interface to the gateway to the upper-level communication systems (NSR or SR) or controllers with functional safety capabilities

3.3.6
FSP parameter
parameter set for the administration and operation of the SDCI-FS protocol

3.3.7
FST parameter
parameter set for the safety-related technology of an FS-Device, for example light curtain

3.3.8
Safety PDU
Safety Protocol Data Unit
SPDU
PDU transferred through the safety communication channel

[SOURCE: IEC 61784-3:2021, 3.1.47, modified – Notes have been removed and admitted term has been added.]

3.4 Symbols and abbreviated terms

AIDA	Automatisierungsinitiative Deutscher Automobilhersteller (Automation iniative of the German automotive manufacturers)	
AL	application layer	
BEP	bit error probability	
C/Q	connection for communication (C) or switching (Q) signal (SIO)	
CRC	cyclic redundancy check	
DDO	Device data object	
DI	digital input	
DIP	dual in-line package	
DL	data link layer	
DO	digital output	
DS	data storage	
DTI	Device Tool Interface	
DTM	Device Type Manager	[IEC 62453 all parts]
FDI	Field Device Integration	[IEC 62769 all parts]
FDT	Field Device Tool	[IEC 62453 all parts]
FS	functional safety	
FSCP	functional safety communication profile (e.g. IEC 61784-3-x series)	
FS-AI/AO	functional safety analog input/output	
FS-DI/DO	functional safety digital input/output	
FSP	functional safety protocol (parameter)	
FST	functional safety technology (parameter)	
I/O	input / output	
IODD	IO Device Description	
IOPD	SDCI Parameterization and Diagnostic tool	
I/Q	connection with several options: DI, OSSD2e, or DO	
L-	power supply (-)	
L+	power supply (+)	
LSO, MSO	least significant octet, most significant octet	
M12	circular connector	[IEC 61076-2-113]
N24	24 V extra power supply (-); Port class B	
NSR	non-safety-related	
OD	On-request Data	
OK	"OK", values or state correct	
OSSD	output signal switching device (self-testing electronic device with built-in OSSD)	[IEC 61496-1]

OSSDe	output signal switching device (self-testing electronic device with built-in OSSD according to this document)	
OSSD1/2e	pin assigment of both OSSDe signals	
OSSDm	output signal switching device (relay and solid state outputs)	[IEC 60947-5-5]
P24	24 V extra power supply (+); Port class B	
PD	Process Data	
PDin	functional safety input process data (from an FS-Master's view)	
PDout	functional safety output process data (from an FS-Master's view)	
PDCT	Port and Device configuration tool	
PDU	protocol data unit	
PFH	average frequency of a dangerous failure per hour	[IEC 61508-4]
PID	program interface description	
PL	physical layer	
PLC	programmable logic controller	
PS	power supply (measured in V)	
RIO	remote I/O	
rms	root mean square ("effective")	
SCL	safety communication layer	
SDCI	single-drop digital communication interface	[IEC 61131-9]
SDCI-FS	single-drop digital communication interface for functional safety	
SIO	standard input output (digital switching mode)	[IEC 61131-2]
SLM	safety layer manager	
SM	system management	
SPDU	safety protocol data unit	
SR	safety-related	
SSI	synchronous serial interface (usually for encoders)	
TAF	temporary acknowledgment file	
TBF	temporary backchannel file	
TPF	temporary parameter file	
UART	universal asynchronous receiver transmitter	
UML 2	unified modeling language, edition 2	[ISO/IEC 19505-2]
USB	universal serial bus	[www.usb.org]
WURQ	wake-up request pulse	
XML	extensible markup language	

3.5 Conventions

3.5.1 Behavioral descriptions

For the behavioral descriptions, the notations of UML 2 are used, mainly for state and sequence diagrams (see [1] [4], [2], [3]).

Events to trigger a transition usually can be a signal, service call, or timeout. Logic conditions (true/false) shall be the result of a [guard]. To alleviate the readability and the maintenance of the state machines, the diagrams do not provide the actions associated with a transition. These actions are listed within a separate state-transition table according to [4].

---

4 Numbers in square brackets refer to the bibliography.

The state diagrams shown in this document are entirely abstract descriptions. They do not represent a complete specification for implementation.

3.5.2 Memory and transmission octet order

Figure 3 demonstrates the order that shall be used when transferring WORD based data types from memory to transmission and vice versa.

NOTE Existing microcontrollers can differ in the way WORD based data types are stored in memory: "big endian" and "little endian". If designs are not considering this fact, octets can be erroneously permuted for transmission.

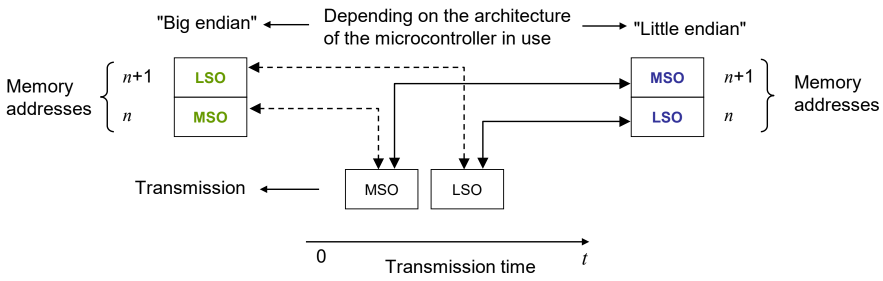


Figure 3 – Memory and transmission octet order

4 Overview of SDCI-FS

4.1 Purpose of the technology and feature levels

4.1.1 Base SDCI-FS technology

This document specifies a new lean functional safety communication protocol on top of the existing SDCI transmission system specified in IEC 61131-9. Figure 4 illustrates how the corresponding SDCI-FS communication layers are located within the architectural models of Master and Device such that they become FS-Master and FS-Device. Most of the original SDCI design remains unchanged for this document.

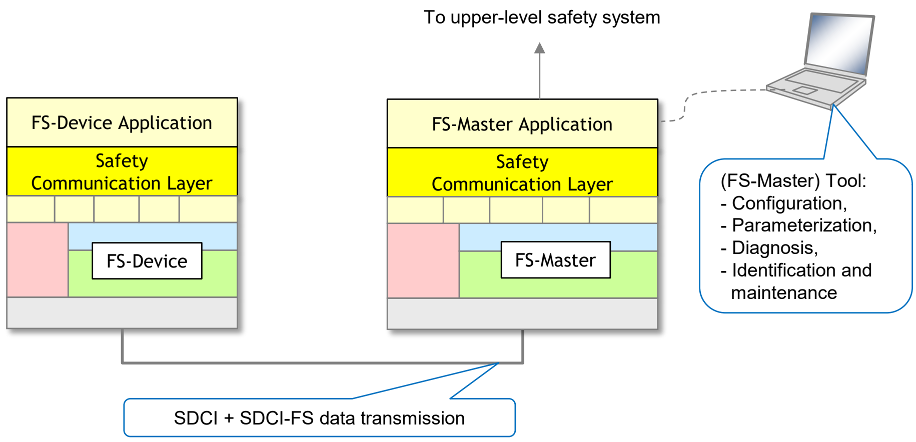


Figure 4 – SDCI-FS communication layer model

The SDCI-FS communication layer accommodates the functional safe transmission protocol. This protocol generates a safety PDU consisting of the FS-I/O data, protocol control or status data, and a CRC signature. The safety PDU together with optionally non-safety-related data is transmitted as SDCI Process Data between an FS-Master and one single FS-Device (point-to-point). It is suitable for functional safety applications up to SIL3 or PLe and the PFH for one connection is less than 10^{-9}/h.

SDCI-FS increases the number of Port modes and thus requires changes to the Physical Layer and Configuration/System Management.

Changes are required for the Master-(Software)-tool to provide the necessary safety-related configuration and parameterization of the protocol (FSP-Parameter) as well as of the particular FS-Device technology (FST-Parameter).

SDCI-FS also supports OSSDe as a migration strategy, like the SIO mode. It does not support

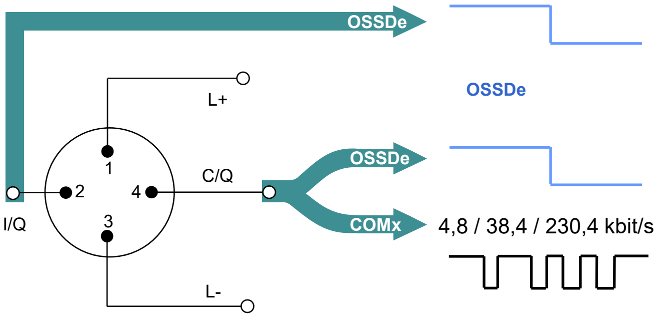
- wireless connections between FS-Master and FS-Device (see Clause H.2),
- cascaded FS-Master/FS-Device systems.

4.1.2 From "analog" and "switching" to communication

In "Safety-for-Machinery", usually the switch states (on/off) of relays or sensors are transmitted similar to standard SDCI (SIO) as a 24 V or 0 V signal to FS-DI-Modules within remote I/Os. In contrast to standard SDCI-FS, due to safety requirements, these signals are redundant, either equivalent (OSSDe = 11→00) or antivalent (OSSDm = 01→10) switching.

NOTE OSSDe stands for IEC 61496-1 and OSSDm for IEC 60947-5-5 concepts (see [5]).

The electrical characteristics for the OSSDe interface are following IEC 61131-2, type 1 (see Figure 5).



Pin	Signal	Definition	Standard
1	L+	24 V	IEC 61131-2
2	I/Q	Not connected, DI / **OSSD2e**, or DO	IEC 61131-2
3	L-	0 V	IEC 61131-2
4	Q	DI / **OSSD1e**, or DO	IEC 61131-2
	C	"Coded switching" COM1…3 + new SDCI-FS	IEC 61131-9 IEC 61139-2

Key: OSSDe = Equivalent switching redundant signals

Figure 5 – Port interface extensions for SDCI-FS

Measurement of physical quantities such as temperature, pressure, position, or strain (FS-AI-Modules) has several interface solutions such as 4 to 20 mA, 0 to 10 V, or SSI, but no common signal transmission technology (see Figure 6, left).

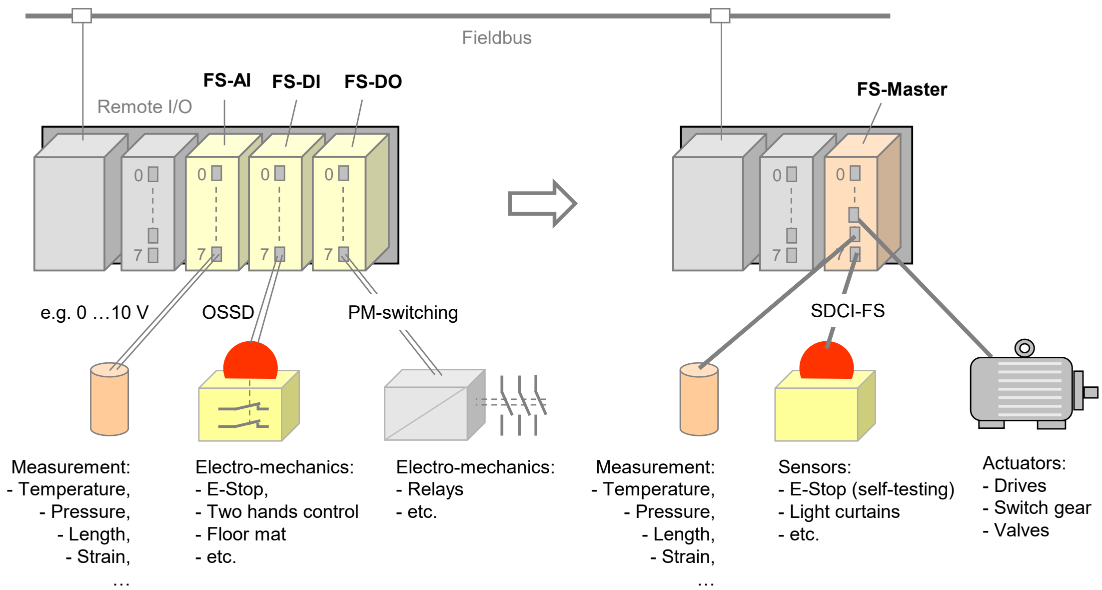


Figure 6 – Migration to SDCI-FS

Actuators such as motors can be de-energized via FS-DO-Modules and connected relays as shown in Figure 6 (left).

Without additional interfaces, it was not possible in all cases to configure or parameterize the safety devices or to receive diagnosis information.

SDCI-FS can now provide a functional safe and reliable solution for process data exchange (signal states and measurement values) via single drop digital communication (SDCI), as well as parameterization and diagnosis (see Figure 6, right).

4.1.3 Minimized paradigm shift from FS-DI to FS-Master

Similar to nowadays safety devices for FS-DI modules (see Figure 7) and in contrast to FSCP-based safety devices, it is not necessary to

a) setup an *authenticity code switch or adequate software solution*,

b) assign a *watchdog time*,

c) use any software tool in case of *FS-Device replacement*.

Authenticity is guaranteed through checking of the correct FS-Device to the assigned FS-Master Port during commissioning like FS-DI modules. However, SDCI-FS provides means to discover any incorrect plugging.

SDCI-FS uses a watchdog timer for the transmission of safety data in time (Timeliness). The system is able to calculate the required watchdog time automatically due to the point-to-point nature of the transmission.

FS-Device replacement without tools can be achieved using the original SDCI Data Storage mechanism.

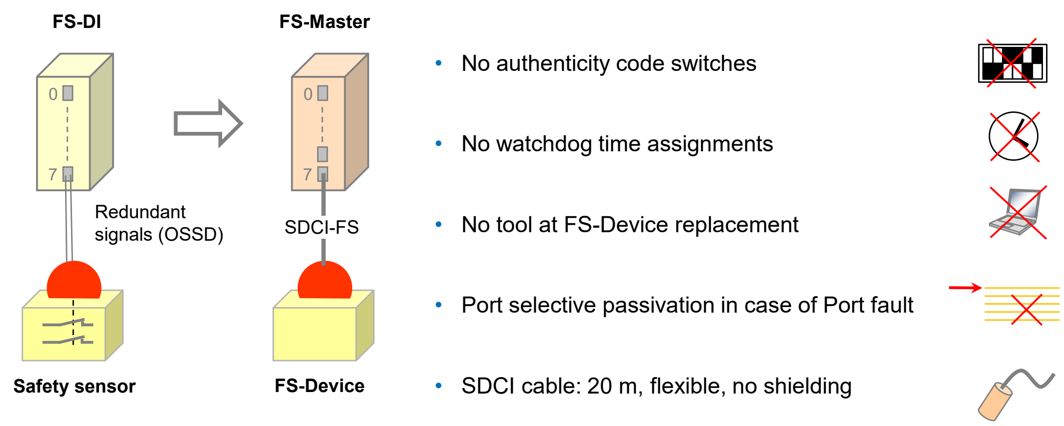


Figure 7 – Minimized paradigm shift from FS-DI to FS-Master

The FS-Master supports *Port selective passivation* in case of a Port fault and *signal granular passivation* in case of a channel fault within for example a remote I/O terminal ("Hub") connected to an FS-Master Port.

Cables are the same as with SDCI, i.e. unshielded with a maximum of 20 m. However, due to the higher permitted power supply current of 1000 mA per Port, the overall loop resistance RL_{eff} can only be 1,2 Ohm (see Table 9 and IEC 61131-9).

NOTE Compliance to AIDA rules requires cable color to be any except yellow. However, the connector color shall be yellow (RAL 1004).

4.1.4 Following the SDCI paradigm (SIO vs. OSSDe)

Standard SDCI-FS supports a Port type A (4 pin) without extra power supply and a Port type B (5 pin) with extra 24 V power supply (see IEC 61131-9). SDCI-FS takes care of several specification levels "a" to "d" (see Figure 8). The number of pins refers to the possible FS-Master pins.

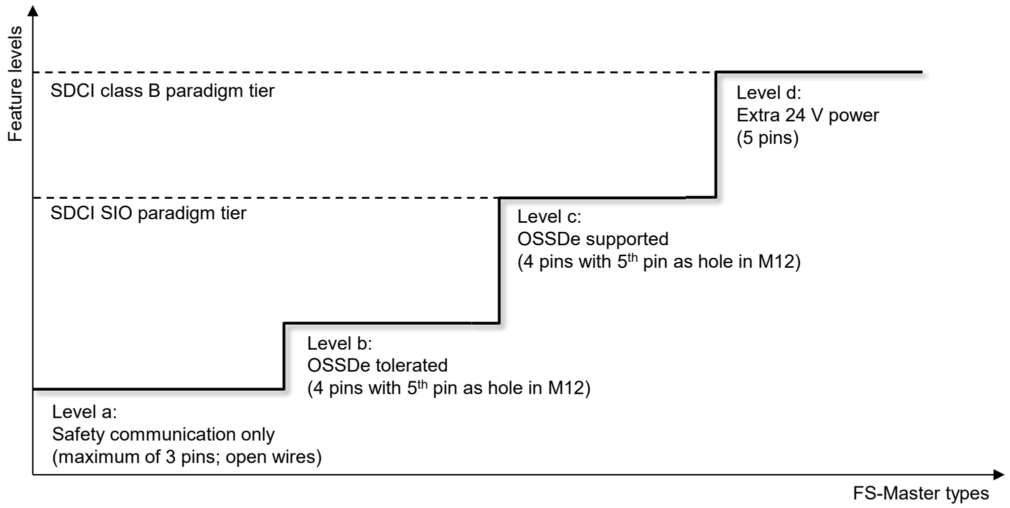


Figure 8 – FS-Master types and feature levels

The original pin layouts of SDCI for Port class A are shown in Figure 9 together with the extensions for level "a" through "c". Table 1 shows the details of these levels.

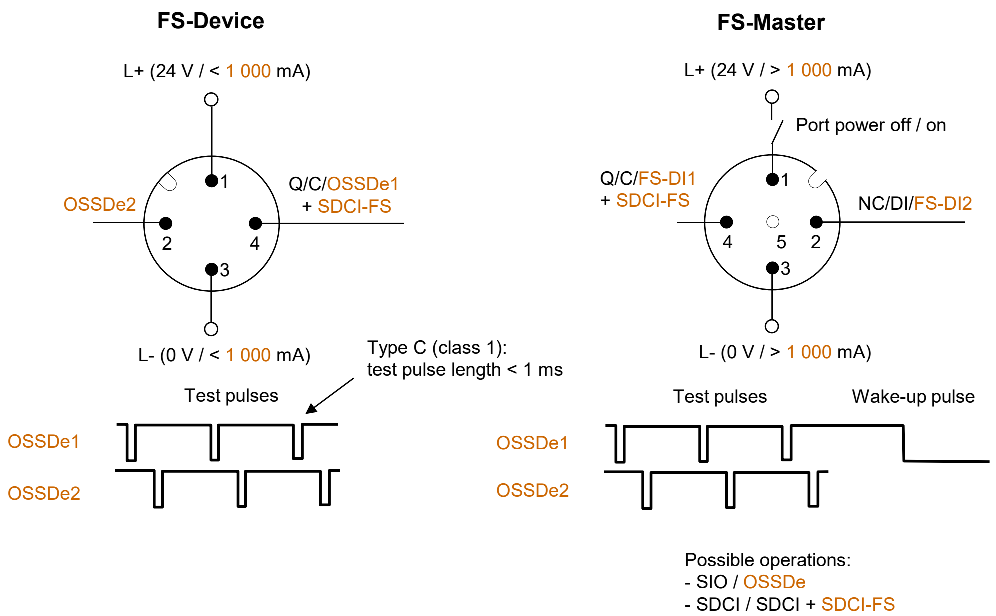


Figure 9 – Original pin layout of SDCI (Port class A)

Level "a" provides communication only (Pin 1, 3, and 4). That means support for sensor-type FS-Devices and actuator-type FS-Devices.

Due to the redundant nature of most of the safety device interfaces, SDCI-FS considers pin 2 for the redundant signal path (e.g. OSSD2e) besides pin 4 for the primary signal path (e.g. OSSD1e)[5]. Thus, level "b" allows FS-Devices to provide OSSDe outputs besides the SDCI-FS communication capability. They can be parameterized with the help of a "USB-Master" and be connected to any FS-DI module in switching mode. When connected to an FS-Master, safety and standard non-safety communication is possible.

---

5 FS-Devices are based on electronics and not on relays. Thus, the electronic version OSSDe is considered.

Level "c" corresponds to the SIO level of standard SDCI Master. In this case, the FS-Master supports an OSSDe mode besides communication (Pin 1, 3, 4 and 2).

Table 1 shows the pin layout and possible operational modes for the feature levels "a" to "c" of the Port class A FS-Device and FS-Master.

Table 1 – Operational modes of feature level "a" to "c" (Port class A)

Feature level	FS-Device		FS-Master	
	Pin 2	Pin 4	Pin 2	Pin 4
"a"	- NC, DI, DO	- DI, DO - SDCI - SDCI + SDCI-FS	- NC, DI, DO	- DI, DO - SDCI - SDCI + SDCI-FS
"b"	- NC, DI, DO - OSSD2e	- DI, DO - OSSD1e - SDCI - SDCI + SDCI-FS	- NC, DI, DO	- DI, DO - SDCI - SDCI + SDCI-FS
"c"	- NC, DI, DO - OSSD2e	- DI, DO - OSSD1e - SDCI - SDCI + SDCI-FS	- NC, DI, DO - FS-DI2	- DI, DO - FS-DI1 - SDCI - SDCI + SDCI-FS

Figure 10 shows the optimized OSSDe commissioning with FS-Masters:

- No filter adjustments due to fixed maximum test pulse length of 1 ms according to type C and class 1 in [6], and
- No discrepancy time adjustments due to fixed maximum discrepancy.

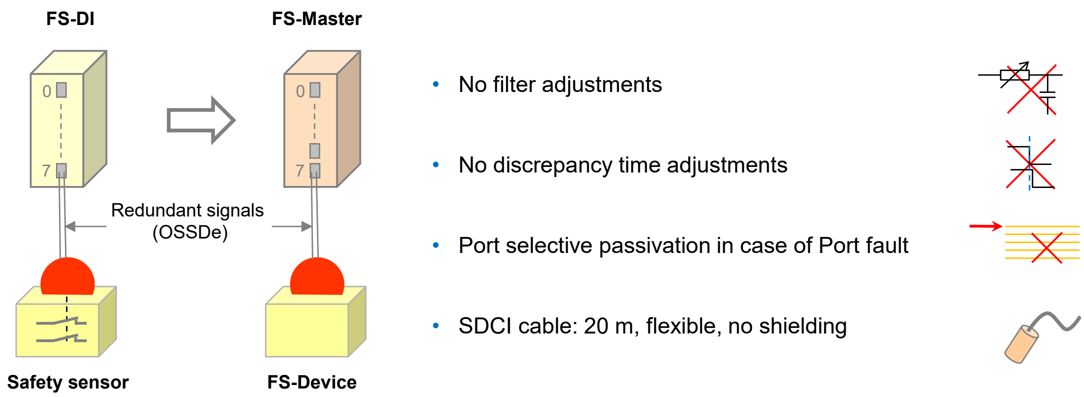


Figure 10 – Optimized OSSDe commissioning with FS-Master

4.1.5 Port class B

A Port class B provides for an extra 24 V power supply for actuators supplementing the main 24 V power supply of SDCI. See IEC 61131-9 for constraints on this extra power supply especially the requirement for electrical isolation of Power 2 from Power 1.

Figure 11 shows the pin layout, signal, and power supply assignment as well as the internal switch for L+.

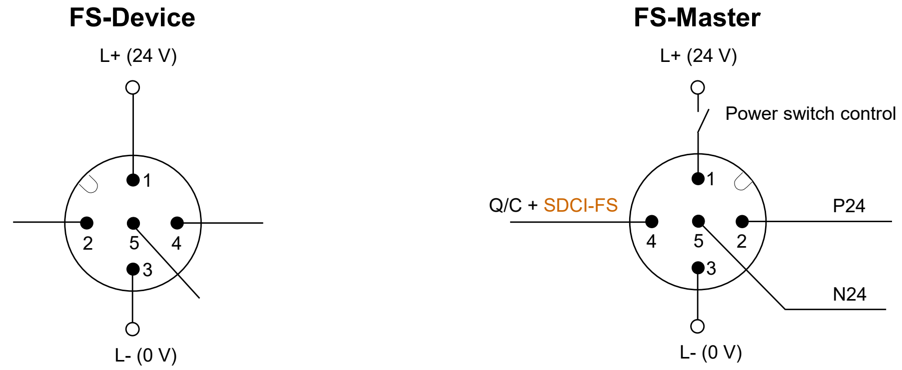


Figure 11 – Level "d" of an FS-Master (Class B)

4.1.6 "USB-Master" with safety parameterization

It is possible to use upgraded "USB-Masters" for off-site configuration, parameterization and test as shown in Figure 12.

Due to functional safety requirements, it will be necessary to extend the Master-tool software for the functional safe configuration and parameterization of the FS-Device technology (FST-Parameters). See Annex F.4.1 for use case.

Table 2 shows the device types that can be supported by such a "USB-Master".

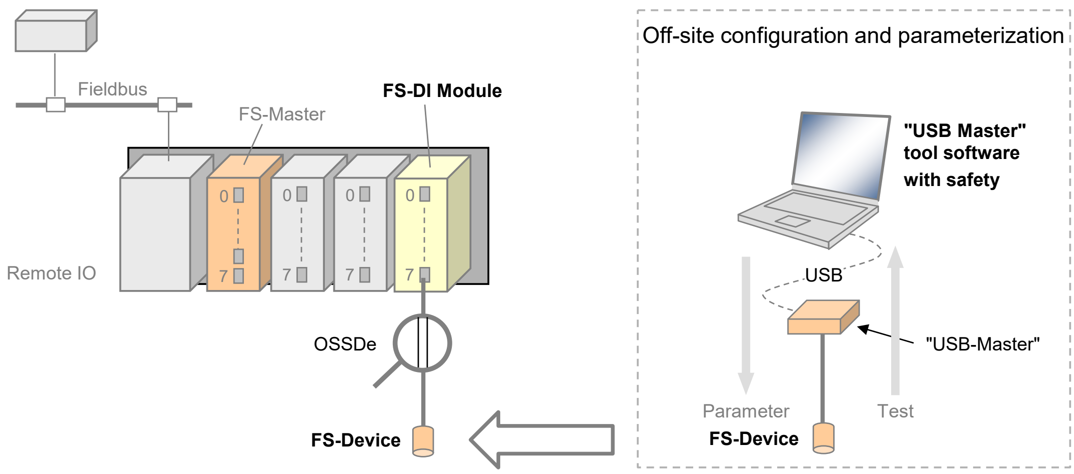


Figure 12 – Off-site configuration and parameterization

4.1.7 Interoperability matrix of safety devices

Table 2 provides an overview of typical safety sensors and actuators and their interoperability with FS-Masters of different feature levels, a "USB-Master" upgraded to safety parameterzation, and conventional FS-DI modules connected to FSCPs.

Table 2 – Interoperability matrix of safety devices

Device type	FS-Master			"USB-Master" with safety parameteri-zation	FS-DI module (FSCP)
	Communi-cation "a"	OSSDe tolerated "b"	OSSDe supported "c"		
Sensor with OSSDe [a]	-	-	OSSDe	-	OSSDe
Sensor with OSSDe and SDCI	-	-	OSSDe	SDCI [b]	OSSDe
Sensor with OSSDe and SDCI-FS	SDCI-FS	SDCI-FS	OSSDe or SDCI-FS	SDCI	OSSDe

Device type	FS-Master			"USB-Master" with safety parameterization	FS-DI module (FSCP)
	Communication "a"	OSSDe tolerated "b"	OSSDe supported "c"		
Sensor with SDCI-FS communication only, e.g. light curtain	SDCI-FS	SDCI-FS	SDCI-FS	SDCI	-
Sensor with OSSDm, e.g. E-Stop	-	-	-	-	OSSDm
Actuator with SDCI-FS, e.g. 400 V power drive, low voltage switch gear	SDCI-FS	SDCI-FS	SDCI-FS	SDCI	-

Key SDCI-FS = SR and NSR data exchange
a Pin layout according to [7]
b Pin layout may differ
USB = Universal Serial Bus, currently the most common interface amongst possible others for offsite parameterization tools due to fast communication combined with power supply

4.2 Positioning within the automation hierarchy

Figure 13 shows the positioning of SDCI-FS within the automation hierarchy.

Classic safety is relay based and thus seemed to be straightforward, easily manageable, and reliable. However, the same criteria that led to the success of fieldbuses, led to the success of functional safety communication profiles (FSCP) on top of the fieldbuses also: reduced wiring, variable parameterization, detailed diagnosis, and more flexibility. SDCI is the perfect complement to the fieldbus communication and bridges the gap to the lowest cost sensors and actuators. It not only provides communication, but power supply on the same flexible and unshielded cable. One type of sensor can be used in the traditional switching mode or in the coded switching mode (communication).

SDCI-FS follows exactly this paradigm with its OSSDe.

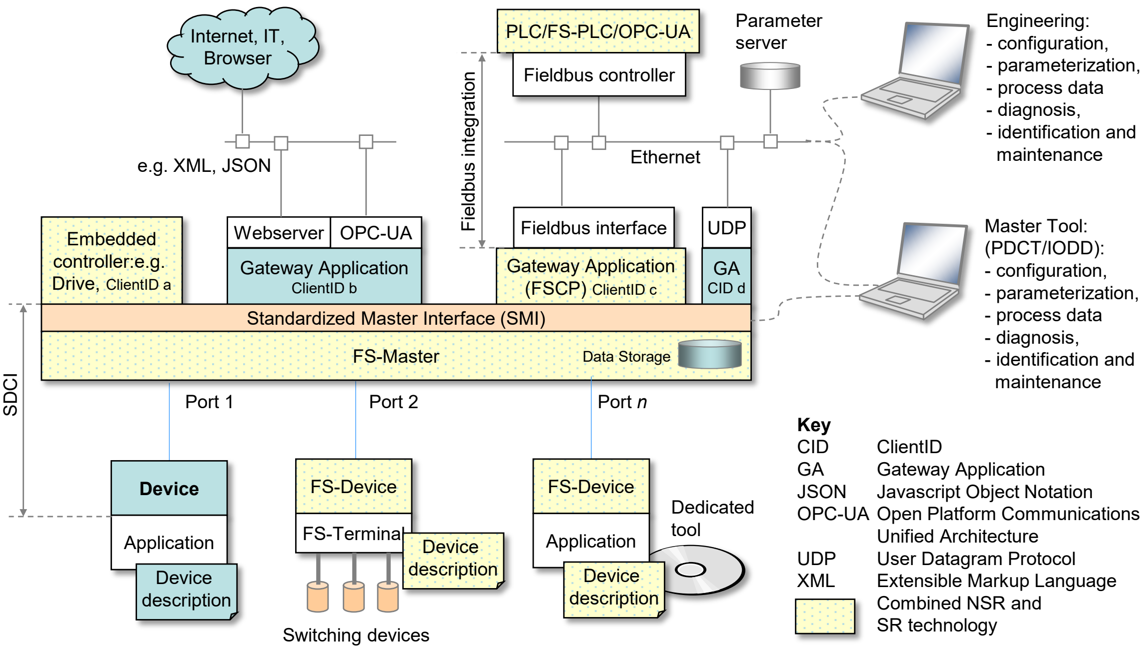


Figure 13 – SDCI-FS within the automation hierarchy

It aims for two main application areas. One is building up safety functions across the SDCI-FS communications and the functional safety communications across fieldbuses. The other builds up safety functions "locally" between an "embedded" safety controller and safety sensors/actuators using SDCI-FS communication.

SDCI-FS allows for building up power saving FS-Devices ("green-line"), for self-testing safety sensors in order to avoid yearly testing, for the reduction of interface types (e.g. 0 to 10 V, 4 to 20 mA, etc.), and for robust and reliable transmission of safety information.

Finally, it is a precondition for new automation concepts such as Industry 4.0 or the Internet-of-Things (IoT).

4.3 Wiring, connectors, and power supply

Port class A types (3 to 4 wires): Cables and connectors as specified in IEC 61131-9 for Class A can be used for SDCI-FS also. However, due to the higher permitted power supply current of up to 1000 mA per Port, the overall loop resistance RL_{eff} can only be 1,2 Ohm. No shielding is required.

Port class B types (5 wires): Cable, wire gauges, shielding, maximum switched currents, interference, signal levels, etc. are not specified within this document.

4.4 Relationship to SDCI

The SDCI communication and its SIO mode are used as the base vehicle ("black channel") for SDCI-FS. Besides SDCI-FS, any FS-Master Port can also be configured for standard SDCI operation.

The independent signal inputs of the SIO mode on Pin 2 and Pin 4 are scanned by an FS-Master simultaneously to achieve an OSSDe interface. The result is propagated to the upper-level safety system as one safety signal. A new Safety Layer Manager supports this feature.

Another new Port configuration mode enables the SDCI-FS communication. Standard state machines are slightly extended to support

- detection of a Ready pulse from the FS-Device on Pin 4;
- power supply (Pin 1) switching OFF/ON in case an FS-Device missed the Wake-up sequence and started its OSSDe operation;
- transmission of functional safety protocol parameters (FSP) during PREOPERATE from FS-Master to the FS-Device;
- activation of the SDCI-FS communication layer (SCL);
- activation of the FS Process Data Exchange within the Safety Layer Manager:

4.5 Communication features and interfaces

FS Process Data from and to an FS-Device are always packed into a safety code envelop consisting of the Port number, a safety PDU counter, protocol Control/Status information, and a 16/32-bit CRC signature. The minimum safety PDU size is 4 octets in case of no FS Process Data. SDCI-FS uses M-Sequence TYPE_2_V.

Only a subset of the SDCI data types is permitted: Boolean (packed as record), IntegerT(16), and IntegerT(32).

Parameterization within the domain of safety for machinery requires a "Dedicated Tool" per FS-Device or FS-Device family. The Device Tool Interface (DTI) based on proven technology has been chosen for the links between FS-Master tool, FS-Device, and its "Dedicated Tool". The FS-Master tool shall provide communication means for a "Dedicated Tool" to allow for the transmission of safety technology parameters (FST parameters) to and from an FS-Device. The "Dedicated Tool" and the FS-Device are both responsible for sufficient means to secure the transmitted data, for example via CRC signature or read-back.

4.6 Parameterization

SDCI-FS comprises a three-tier concept. The first tier is IODD based and contains all basic non-safety parameters for a Device or FS-Device.

The second tier requires an extension of the IODD for the fixed set of protocol parameters (FSP). These parameters are safety-related and secured via CRC signature against unintended changes of the IODD file. The interpreter of the FS-Master tool provides a safety-related extension for the handling of the FSP parameters. Usually, the FS-Master tool is able to determine and suggest the FSP parameter assignments (instance values) automatically and thus relieves the user from assigning these values initially. He can check the plausibility of the values and modify them if required.

The third tier deals with technology specific safety parameters (FST) of an FS-Device. SDCI-FS classifies two types of FS-Devices. Type "basic" requires only a few orthogonal FST parameters, whereas type "complex" can have a number of FST parameters requiring business rules and verification or validation wizards. Usually, the latter comes already with existing PC software ("Dedicated Tool") used for several functional safety communication profiles for fieldbuses.

The FST parameters for type "basic" are coded as any non-safety parameter within the IODD. They can be modified and downloaded to the FS-Device as usual. However, a diverse second path allows for checking these assignments for correctness. At the end of a parameterization session, the user launches a safety-related "Dedicated Tool" (FS-IOPD) for the calculation of a CRC signature across all FST instance values provided by the FS-Master tool.

For both types of FS-Devices, the "Dedicated Tool" presents a CRC signature, which the user can copy into one of the FSP parameters. Upon reception of the FSP parameters at start-up, the FS-Device calculates a CRC signature across the locally stored instance values and compares it with the received CRC signature.

This method is used also for the check after using the SDCI Data Storage mechanism.

4.7 Role of FS-Master and FS-Gateway

The role of the FS-Master is extended to safe monitoring of Process Data, transferred to and from FS-Devices with respect to timeliness, authenticity, and data integrity according to IEC 61784-3:2021. Concerning authenticity, it uses the authenticity code assigned to the FS-Master by the upper-level FSCP system and the Port number. This prevents from local Port related misconnections and misconnections whenever several FS-Masters are located side by side.

An FS-Master can be equipped by a safety controller, for example according to IEC 61131-6, or vice versa, and thus build-up a stand-alone safety system with its own complete safety functions.

With the help of an FS-Gateway in conjunction with the FS-Master, safety functions can be build-up across the upper-level FSCP system using the safety sensors and actuators connected to the FS-Master.

4.8 Mapping to upper-level systems

Specification of the mapping to an upper-level FSCP system is the responsibility of the fieldbus organization. SDCI-FS made provisions to meet the majority of FSCPs for example via reduced number of data types, descriptions of safety IO data, Port selective passivation, and operator acknowledgment signals to prevent from automatic restart of machines.

4.9 Structure of the document

The structure of this document complies mostly with the structure of IEC 61131-9. Clause 5 specifies the extensions to the Physical Layer (PL), mainly the OSSDe issues, the wake-up behavior, and the additional Port modes. Extensions to SIO are specified in Clause 6, those to data link layer (DL) in Clause 7, those to system management (SM) in Clause 8, those to the FS-Device in Clause 9, and those to the FS-Master in Clause 10.

The core part of this document is the safety communication layer (SCL) in Clause 11. It comprises the SCL services, protocol, state machines, and management. In addition, it deals with integrity measures, with protocol (FSP) and technology (FST) parameters, with the integration of "Dedicated Tools" via Device Tool Interface (DTI) technology, with Port selective passivation, and with SCL diagnosis. Clause 12 complements the core part by functional safety processing either through mapping to the upper-level system or local.

Extensions to parameters and commands are specified in Annex A, those to EventCodes in Annex B, and those to data types in Annex C. CRC polynomial issues are presented in Annex D, the IODD aspects in Annex E, the Device Tool Interface technology in Annex F, main scenarios in Annex G, and the system requirements in Annex H. Annex I provides information on test and assessment. Annexes A, C, D, and E are safety related.

5 Extensions to the Physical Layer (PL)

5.1 Overview

Figure 14 shows the adapted physical layer of an FS-Master (class A).

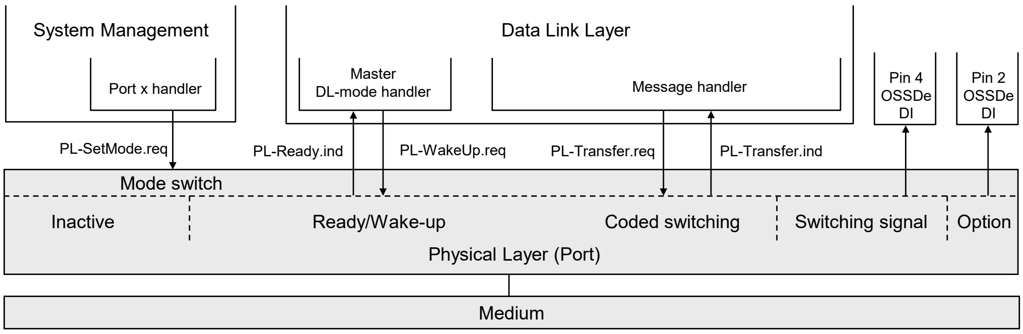


Figure 14 – The SDCI physical layer of an FS-Master (class A)

Pins 2 and 4 shall be scanned simultaneously to achieve OSSDe functionality. The FS-Master shall scan the C/Q line for the Ready signal of the FS-Device (see A.2.11).

Figure 15 shows the adapted physical layer of an FS-Device (class A).

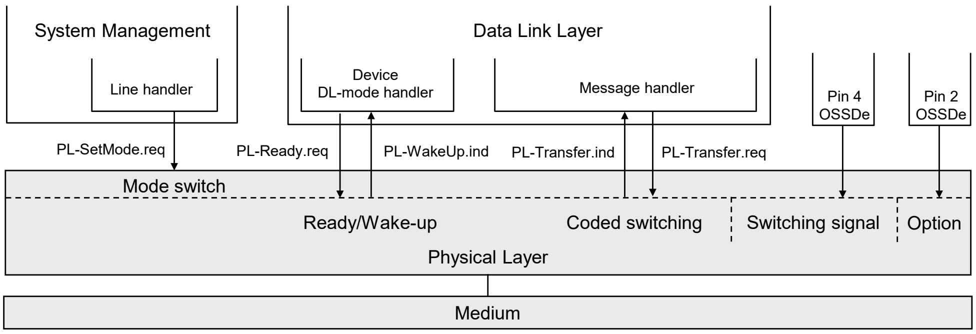


Figure 15 – The physical layer of an FS-Device (class A)

Pins 2 and 4 carry the OSSDe signals. FS-Device shall set the Ready pulse after internal testing.

5.2 Extensions to PL services

5.2.1 PL_SetMode

The PL-SetMode service is extended by the additional TargetMode "OSSDe" (C/Q line and I/Q line in digital input mode).

5.2.2 PL_Ready

The PL-Ready service initiates or indicates a Ready signal on the C/Q line. Whenever the FS-Device finished its internal safety-related hardware and software tests, it sets this signal. The FS-Master polls this signal and upon reception initiates the wake-up sequence. This unconfirmed service has no parameters. The service primitives are listed in Table 3.

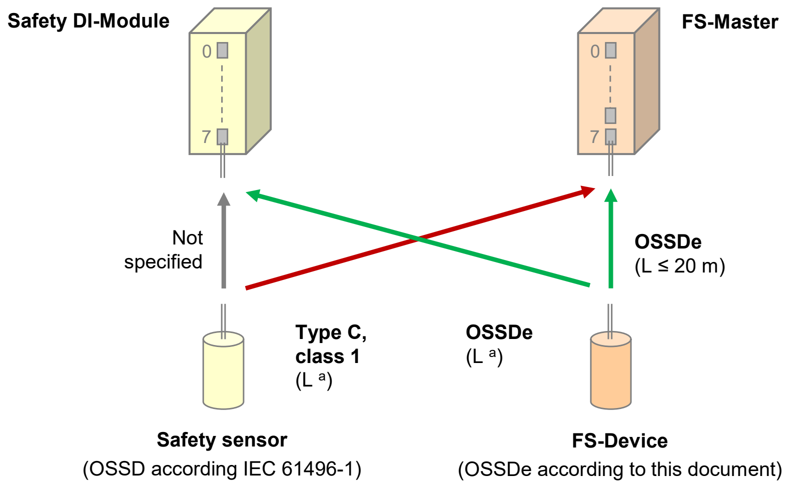
Table 3 – PL_Ready

Parameter name	.req	.ind
<none>		

5.3 Transmitter/receiver

5.3.1 Assumptions for the expansion to OSSDe

Figure 16 shows the cross compatibility between OSSD based safety sensors and OSSDe based FS-Devices.



Key a length is manufacturer specific

Figure 16 – Cross compatibility OSSD and OSSDe

The following assumptions are the basis for the design of the OSSDe expansion:

- The SIO paradigm of SDCI shall apply for SDCI-FS in order to allow manufacturers the combined function of OSSDe and SDCI-FS communication within one FS-Device.
- A Port on the FS-Master (with "FS-DI" according to Figure 9) shall have fixed configurations as either SDCI-FS or OSSDe interface with no or minor adjustments in respect to addressing, watchdog times, discrepancy times, or filter times.
- In order to allow OSSD based sensors on the market to be connected to the FS-Master, the FS-DI interface shall support the necessary adjustments for Type "C", class "1" devices according to [6].
- The OSSDe interface shall only be designed as input for the FS-Master Port (safety sensors, Class A connectors). Most actuators are supplied by three-phase alternating current such as power drives, low voltage switch gears, motor starters, etc.
- Actuators such as valves with diversity and relays shall be supported by FS-Master with Ports "level d" (see Clause 6).

5.3.2 OSSDe specifics

5.3.2.1 General

Similar to the SIO approach, FS-Master according to level "c" support connectivity to existing functional safety devices with OSSDe. OSSDe in this document is defined as two outputs with signals that are both switching in equivalent manner as opposed to antivalent manner, where one signal is normally off and the other normally on (OSSDm).

The FS-Master Port is designed to achieve a maximum of possible compatibility to existing OSSD devices using interface type C, class 1 defined in [6].

Figure 17 shows a corresponding reference model from [6], adapted to SDCI-FS. The information-"source" on the left corresponds for example to a sensor device, whereas the information-"sink" on the right side represents an input of the FS-Master Port class A. Power is supplied by the sink.

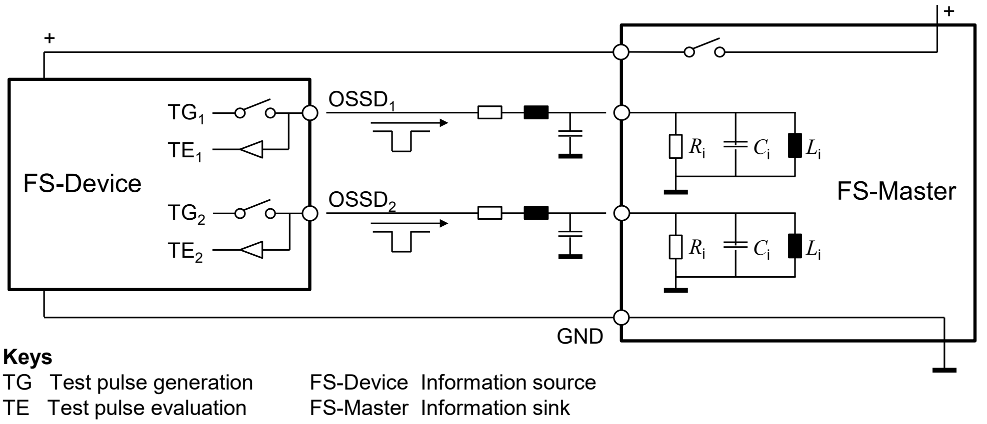


Figure 17 – Principle OSSDe function

The worst-case values for the line resistance and capacitance are defined in Table 9. In case of SDCI-FS, line inductance is negligible at a length of 20 m. The design of the FS-Master Port shall ensure values for R_i, C_i, and L_i guaranteeing proper signal behavior according to Table 8.

Table 4 shows the OSSD states and conditions defined in IEC 61496-1:2020.

Table 4 – OSSD states and conditions

State	Cause	Voltage range	Current
OFF	Demand	- 3 V to + 2 Vrms (+ 5 V peak)	< 2 mA (leakage) NOTE
ON	No demand	+ 11 V to + 30 V	> 6 mA
NOTE IEC 61131-9 permits 5 mA for the voltage range of 5 V to 15 V			

OFF state:

For this interface, the OFF state is defined as the "powerless" state, where voltage and current of at least one OSSDe shall be within (voltage) and below (current) defined limits (see Table 4). If the safety function is demanded, or the source (the device) detects a fault, the OSSDe signals shall go to the OFF state. Antivalent voltage levels, so-called discrepancy, on both OSSDe outputs of the device shall be treated as OFF state. The duration of this state shall be within a specified discrepancy tolerance time. If the tolerance time is exceeded, the Port is considered to be faulty.

ON state:

For this interface, the ON state is defined as the "powered" state, where voltage and current on both OSSDe outputs shall be within the voltage range and above defined current limits, when sinked by IEC 61131-2 inputs (see Table 4). Test pulses within specified ranges in voltage levels, durations and intervals are permitted. Antivalent voltage levels, so-called discrepancy, on both OSSDe outputs of the device shall be treated as OFF state.

5.3.2.2 Detection of cross connection faults

Tests are required for the detection of the cross-connection faults specified in IEC 61496-1 and shown in Table 5.

Table 5 – Cross connection faults

Fault	Diagnostics
Short circuit between OSSD 1 and OSSD 2	Test pulses (runtime diagnosis)
Short circuit between OSSD 1 or OSSD 2 and V+	Test pulses (runtime diagnosis)
Short circuit between OSSD 1 or OSSD 2 and V-	Test pulses (runtime diagnosis)
Open circuit of the power supply return cable (V-)	Type test, maximum leakage current
Open circuit of the functional earth (bonding) conductor	Type test, no functional earth
Open circuit of the screen of screened cable	Not required due to no shielding
Incorrect wiring	Discrete wiring only, organizational issue (test during commissioning)

The means for detecting short circuits are test pulses at runtime. The means for testing the behavior in case of open circuits is the type test during the assessment. Figure 18 shows the test pulses approach for the detection of cross connection faults. A cross fault occurs if one channel detects both signal patterns.

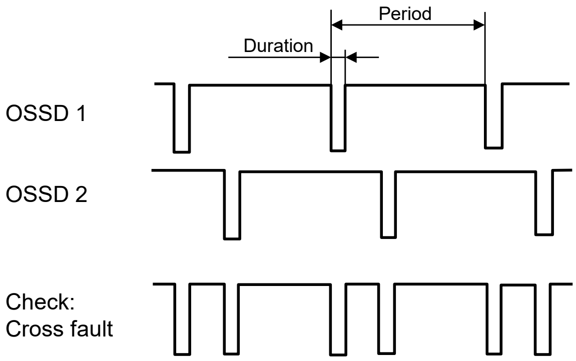


Figure 18 – Test pulses to detect cross connection faults

Three methods of testing (intervals) are commonly used:

- Test pulses at each program cycle of the safety device (dependency on configuration)
- Test pulses at fixed times
- Test pulses after any commutation OFF → ON

5.3.2.3 FS-Device OSSDe output testing

The test pulses of this interface type for testing the transmission line are created and evaluated on the safety device side. This way the source can diagnose the correct functioning of the output stage. In case of any detected error both OSSDe outputs shall be switched to the safe state (Lock-out condition = OFF).

The test pulses are created in a periodic manner on both OSSD lines. In order to detect short circuits between the lines or between the lines and power-supply, the test pulses of both lines can be time-shifted to each other (see Figure 19).

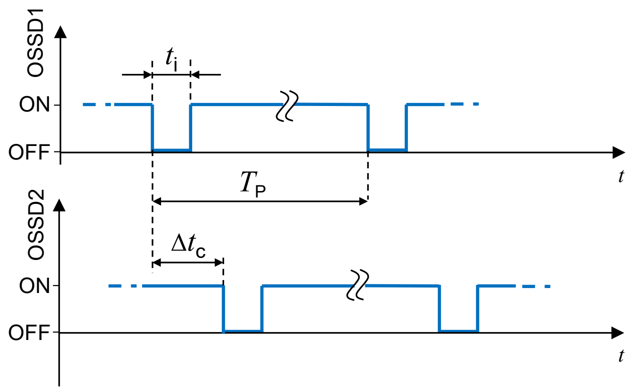


Figure 19 – OSSD timings

The following parameters specify the characteristics of the test pulses on the OSSD interface:

- Period of test pulses (T_P)
- Duration of test pulses (t_i)
- Time-shift between test pulses of both channels (Δt_c)

The characteristics of test pulses are classified in [6]. FS-Devices shall meet type C and class 1 requirements with a test pulse length $t_i \leq 1000$ µs (see Table 7).

5.3.3 Start-up of an FS-Device (Ready pulse)

Figure 20 shows the typical start-up sequence of an OSSD sensor without SDCI-FS capability. During self-test for functional safety, both OSSD signals shall be OFF. When finished, the sensor switches to ON and starts test pulses. A demand causes the sensor to switch OFF. A fault causes the sensor to switch to lock-out condition (OFF) and to remain in this state until repair.

NOTE For simplicity, the figure shows only one OSSD channel.

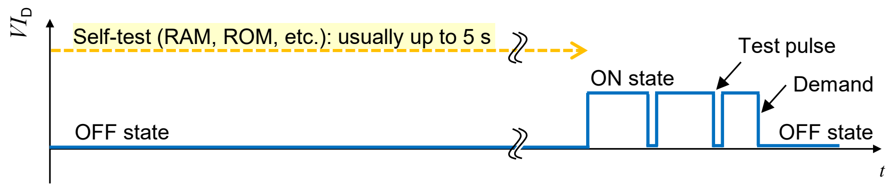


Figure 20 – Typical start-up of an OSSD sensor

Figure 21 shows the start-up of an FS-Device with OSSDe capability connected to a classic FS-DI module.

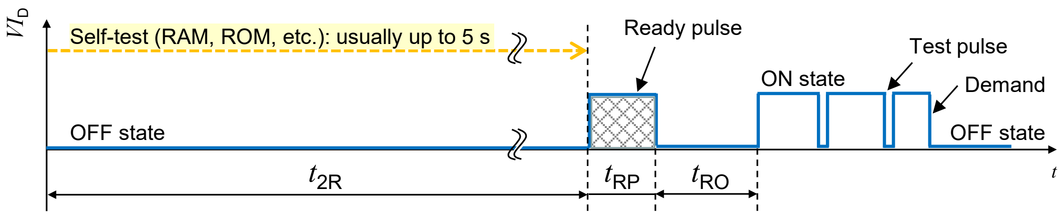


Figure 21 – Start-up of an FS-Device

In contrast to a classic sensor, the FS-Device provides only on pin 4 (see Figure 9) a so-called Ready-pulse of a certain length to indicate the FS-Master its readiness after self-testing. After a certain recovery time, the FS-Device switches to ON and starts test pulses like a classic safety sensor.

Timings and Wake-up behavior of the FS-Device are specified in 5.7.

5.3.4 Electric characteristics of a receiver in FS-Device and FS-Master

The voltage range and switching threshold definitions are the same for FS-Master and FS-Device since FS-Master Ports shall be able to operate with non-safety SDCI Devices. The definitions in Table 6 in accordance with IEC 61131-9 apply.

Table 6 – Electric characteristics of a receiver

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
$VTHH_{D,M}$	Input threshold 'ON'	10,5	n/a	13	V	See NOTE 1
$VTHL_{D,M}$	Input threshold 'OFF'	8	n/a	11,5	V	See NOTE 1
$VHYS_{D,M}$	Hysteresis between input thresholds 'ON' and 'OFF'	0	n/a	n/a	V	Shall not be negative See NOTE 2
$VIL_{D,M}$	Permissible voltage range 'OFF'	$V0_{D,M}$ -1,0	n/a	n/a	V	With reference to relevant negative supply voltage
$VIH_{D,M}$	Permissible voltage range 'ON'	n/a	n/a	$V+_{D,M}$ + 1,0	V	With reference to relevant positive supply voltage.
NOTE 1 Thresholds are compatible with the definitions of type 1 digital inputs in IEC 61131-2. NOTE 2 Hysteresis voltage $VHYS = VTHH - VTHL$						

Figure 22 demonstrates the switching thresholds for the detection of OFF and ON signals.

NOTE 'OFF' and 'ON' correspond to 'L' (Low) and 'H' (High) in IEC 61131-9.

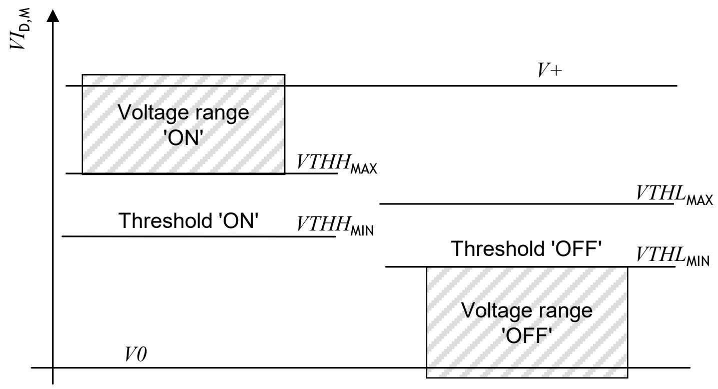


Figure 22 – Switching thresholds for FS-Device and FS-Master receivers

The FS-Master ignores pulses below 11 V (max. 15 mA or max. 30 mA) that are shorter than 1 ms.

5.4 Electric and dynamic characteristics of an FS-Device

In general, the specified values and ranges of IEC 61131-9 apply (see Figure 23).

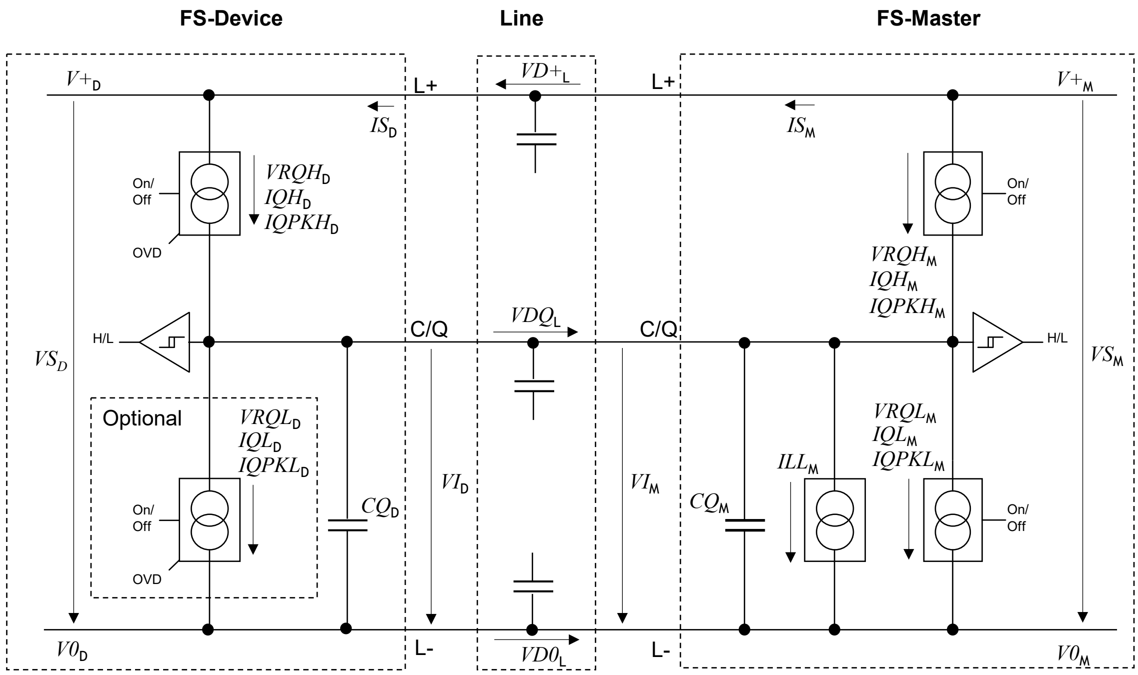


Figure 23 – Reference schematics (one OSSDe channel)

The subsequent illustrations and parameter tables refer to the voltage level definitions in Figure 24.

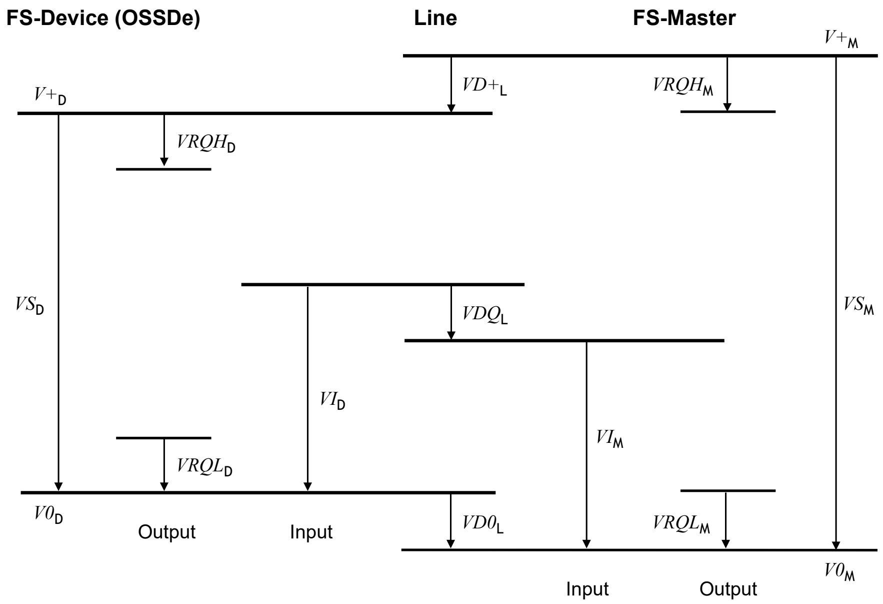


Figure 24 – Voltage level definitions

The electric and dynamic parameters for the OSSDe interface of an FS-Device are specified in Table 7 in accordance with IEC 61131-9.

Table 7 – Electric and dynamic characteristics of the FS-Device (OSSDe)

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
VS_D	Supply voltage	18	24	30	V	See Figure 24
ΔVS_D	Ripple	n/a	n/a	1,3	V_{pp}	Peak-to-peak absolute value limits shall not be

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
						exceeded. f_{ripple} = DC to 100 kHz
IS_{D}	Supply current	n/a	n/a	1000	mA	See 5.9
QIS_{D}	Power-up consumption	n/a	n/a	70	mAs	See equation (1) and associated text
$VRQH_{\text{D}}$	Residual voltage 'ON'	n/a	n/a	3	V	Voltage drop compared with $V{+}_{\text{D}}$ (see [8])
$VRQL_{\text{D}}$	Residual voltage 'OFF'	n/a	n/a	3	V	Voltage drop compared with $V0_{\text{D}}$ NOTE 1
IQH_{D}	DC driver current P-switching output ('ON' state)	50	n/a	minimum ($IQPKL_{\text{M}}$)	mA	Minimum value due to fallback to digital input in accordance with IEC 61131-2, type 2
IQL_{D}	DC driver current N-switching output ('ON' state)	0	n/a	minimum ($IQPKH_{\text{M}}$)	mA	Only for push-pull output stages
IQQ_{D}	Quiescent current to $V0_{\text{D}}$ ('OFF' state)	0	n/a	15	mA	Pull-down or residual current with deactivated output driver stages
CQ_{D}	Input capacitance	0	n/a	1,0	nF	Effective capacitance between C/Q and L+ or L- of Device in receive state. See IEC 61131-9 for constraints on transmission rates.
t_{2R}	Time to Ready pulse	n/a	n/a	5 (default), > 5 to 655 as exception	s	See Figure 21 and A.2.11
t_{RP}	Duration of Ready pulse	500	n/a	1000	µs	See Figure 21
t_{RW}	End of Ready pulse to ready for Wake-up	n/a	n/a	50	µs	See Figure 27
t_{RO}	End of Ready pulse to OSSD mode	1,1	n/a	Data sheet	s	See Figure 21
$t_{\text{i}}/T_{\text{P}}$	Test pulse ratio	Data sheet	n/a	1/100	–	See [6] and Figure 19
t_{i}	Test pulse duration	n/a	n/a	1000	µs	See Figure 19
Δt_{c}	Time-shift	0	n/a	Data sheet	ms	See Figure 19
t_{disD}	Discrepancy time	n/a	n/a	1	ms	Demands may occur during tests

NOTE 1 Pull-down of residual voltage with deactivated high-side output driver stage and activated low-side driver stages (if available e.g. push-pull drivers) with externally limited DC driver current of 50 mA maximum

NOTE 2 Characteristics in this table assume OSSD type "C", class "1" or class "2" according to [6] and interface type 1 according to IEC 61131-2

It is the responsibility of the FS-Device designer to select appropriate ASICs according to IEC 61131-9 and/or to provide mitigating circuitry to meet the requirements of IEC 61496-1.

The FS-Device shall be able to reach a stable operational state (ready for Wake-up: T_{RDL}) while consuming the maximum charge (see equation (1)).

$$QIS_D = ISIR_M \times 50\ ms + (T_{RDL} - 50\ ms) \times IS_M \quad (1)$$

5.5 Electric and dynamic characteristics of an FS-Master Port (OSSDe)

In general, the specified values and ranges of IEC 61131-9 apply (see Figure 23 and Figure 24). The definitions in Table 8 are valid for the electrical characteristics of an FS-Master Port.

Table 8 – Electric and dynamic characteristics of the Port interface

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
VS_M	Supply voltage for FS-Devices	20	24	30	V	See Figure 24
IS_M	Supply current for FS-Devices	200	n/a	1000	mA	Rules in 5.9. shall be considered
$ISIR_M$	Current pulse capability for FS-Devices	400	n/a	n/a	mA	See Figure 25
ILL_M	Load or discharge current for 0 V < VI_M < 5 V 5 V < VI_M < 15 V 15 V< VI_M < 30 V	 0 5 5	 n/a n/a n/a	 15 15 15	 mA mA mA	See NOTE 1
$VRQH_M$	Residual voltage 'H'	n/a	n/a	3	V	Voltage drop relating to $V+_M$ at maximum driver current IQH_M
$VRQL_M$	Residual voltage 'L'	n/a	n/a	3	V	Voltage drop relating to $V0_M$ at maximum driver current IQL_M
IQH_M	DC driver current 'H'	100	n/a	n/a	mA	–
$IQPKH_M$	Output peak current 'H'	500	n/a	n/a	mA	Absolute value See NOTE 2
IQL_M	DC driver current 'L'	100	n/a	n/a	mA	–
$IQPKL_M$	Output peak current 'L'	500	n/a	n/a	mA	Absolute value See NOTE 2
CQ_M	Input capacitance	n/a	n/a	1,0	nF	f=0 MHz to 4 MHz
t_{disM}	Discrepancy time	n/a	n/a	3	ms	See NOTE 3 and Table 7
t_i	Test pulse duration	n/a	n/a	1	ms	See Figure 19
T_P	Period of test pulses	10	n/a	n/a	ms	See Figure 19

NOTE 1 Currents are compatible with the definition of type 1 digital inputs in IEC 61131-2. However, for the range 5 V < VI_M < 15 V, the minimum current is 5 mA instead of 2 mA in order to achieve short enough slew rates for pure p-switching Devices.

NOTE 2 Wake-up request current (See 5.3.3.3 in IEC 61131-9:2022).

NOTE 3 Characteristics assume OSSD type "C", class "1" in [6]

The Master shall provide a charge of at least 20 mAs within the first 50 ms after a power-on time of 5 ms without any overload-shutdown (see Figure 25). After 50 ms the current limitations for IS_M in Table 8 apply.

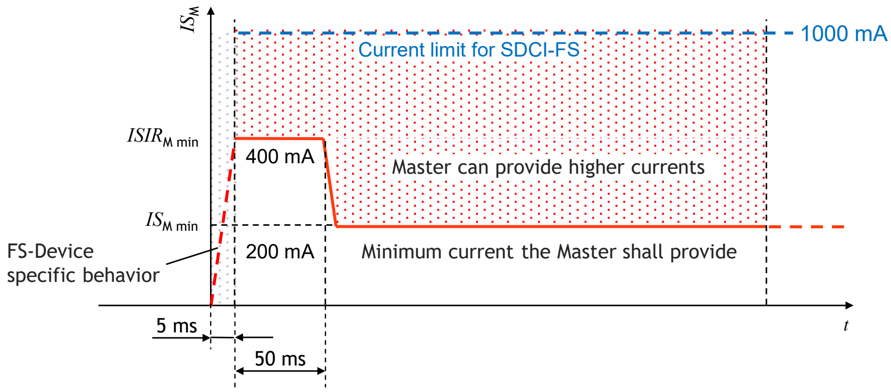


Figure 25 – Charge capability at power-up

5.6 FS-Master Port FS-DI interface

Since OSSD safety sensors can provide different test pulse patterns, the FS-Master Port shall have a suitable input filter, or evaluation algorithm. For the sake of EMC considerations, a combination of both can be used. This means, that the time, in which the signal is below U_{Hmin} shall be less than the maximum allowed test pulse duration.

Any state different to both signals "high", except test pulses, shall be interpreted as safe state.

The EMC levels shall be considered for the layout of an input filter. The communication transmission rate 230 kbit/s conflicts with the input filter. Possible conflict resolution is shown in Figure 26.

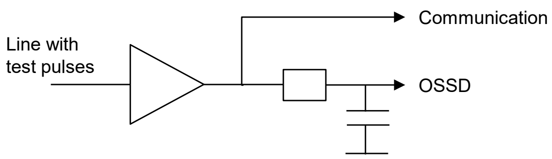


Figure 26 – OSSDe input filter conflict resolution

In general, the specified values and ranges of IEC 61131-9 apply. Basis is interface type 1 of IEC 61131-2. Deviating and supplementary electric and dynamic parameters for the FS-DI interfaces are specified in Table 8.

5.7 Wake-up coordination

Figure 27 shows the start-up of an FS-Device (see IEC 61131-9 for standard timing definitions). After accomplished self-tests, it indicates its readiness for Wake-up through an ON/Ready pulse on the C/Q line (see A.2.11). If no Wake-up occurs within a defined time frame, it starts with test pulses (see Figure 20).

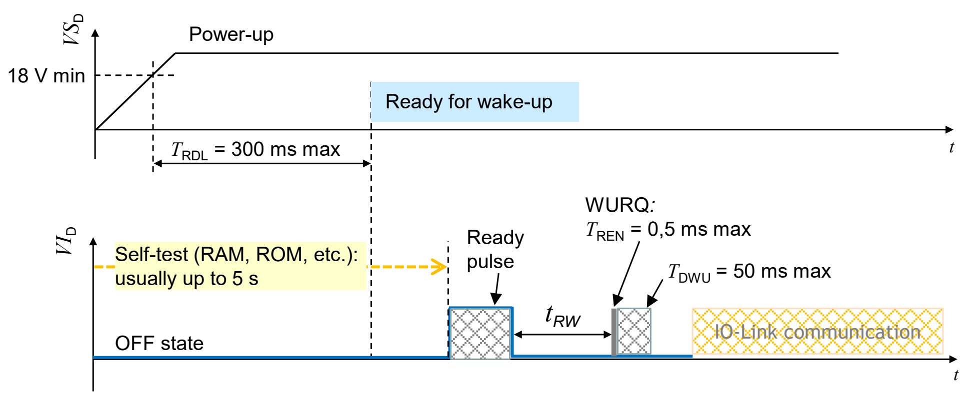


Figure 27 – Start-up of an FS-Device

NOTE Actually, some safety light curtain vendors offer activation of functionality if some connection conditions are activated during start-up phase (e.g. override)

5.8 Fast start-up

Figure 28 illustrates requirements for certain functional safety applications such as for a tool changer on a robot. A so-called fast start-up in non-safety cases shall be achieved within 0,5 s and in case of functional safety within 1 s.

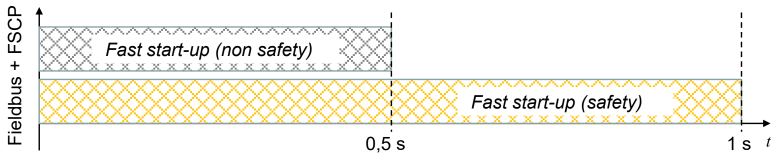


Figure 28 – Required fast start-up timings

NOTE Current safety devices usually require up to 5 seconds for self-testing prior to functional safe operation.

The Ready-pulse concept allows for easier achievable realizations of these requirements (see A.2.11).

5.9 Power supply

An FS-Master Port shall be able to switch its power supply on and off. This enables the FS-Master to restart an FS-Device once it failed to establish communication and started OSSDe operation instead.

The FS-Master Port is the only power supply for SDCI related parts of the FS-Device. Any external power source of the FS-Device shall be totally nonreactive to these parts.

FS-Master shall provide all Ports with a minimum supply of 200 mA and at least one Port with a minimum supply of 1000 mA. The FS-Master shall specify the total maximum current consumption of all its Ports and the derating rules.

Higher currents can conflict with the power switching components and cause interference with the signal lines. The "ripple" requirement in Table 7 shall be considered. The overall cable loop resistance shall be not more than 1,2 Ω (see Table 8 and Table 9).

5.10 Medium

5.10.1 Constraints

For the sake of simplicity in technology and commissioning, SDCI-FS expects a wired point-to-point connection or equivalent consistent transmission and powering between FS-Master and an FS-Device. No storing elements in between are permitted.

5.10.2 Connectors

Connectors as specified in IEC 61131-9 for Class A are permitted.

5.10.3 Cable characteristics

Table 9 shows the cable characteristics for SDCI-FS and non-safety Devices if higher power supply currents than 200 mA are applied.

Table 9 – Cable characteristics

Property	Designation	Minimum	Typical	Maximum	Unit
L	Cable length	0	n/a	20	m
RL_{eff}	Overall loop resistance	n/a	n/a	1,2	Ω
CL_{eff}	Effective line capacitance	n/a	n/a	3,0	nF (<1 MHz)
NOTE These characteristics can deviate from the original characteristics defined in IEC 61131-9.					

6 Extensions to SIO

SIO is only defined for Pin 4 of the Master/Device Port in IEC 61131-9. OSSDe requires inclusion of Pin 2 as specified in Clause 5. Configuration can be performed within the Master/Device applications layer (see Figure 32 and Figure 34).

7 Extensions to the data link layer (DL)

7.1 Overview

Figure 32 and Figure 34 show the DL building blocks of FS-Device and FS-Master. No new or changed services are required. However, both DL-mode handlers are extended by the Ready-pulse feature as shown in 7.2 and 7.3.

7.2 State machine of the FS-Master DL-mode handler

Figure 29 shows the modifications of the FS-Master DL-mode handler versus the Master DL-mode handler in IEC 61131-9.

A new state "WaitOnReadyPulse_10" considers the requirement for the FS-Master to wait on the Ready-pulse of an FS-Device (see 5.7) prior to establish communication via DL_SetMode_STARTUP.

The maximum waiting time is t_{2R} as defined in Table 7. Whenever the time expired, the FS-Master shall run a power-OFF/ON cycle for the connected FS-Device in order to initiate a retry for another Ready-pulse.

The criterion to use the extra path is the guard [safety], which is derived from the new Port configuration "FS_PortModes" (see 10.4.2).

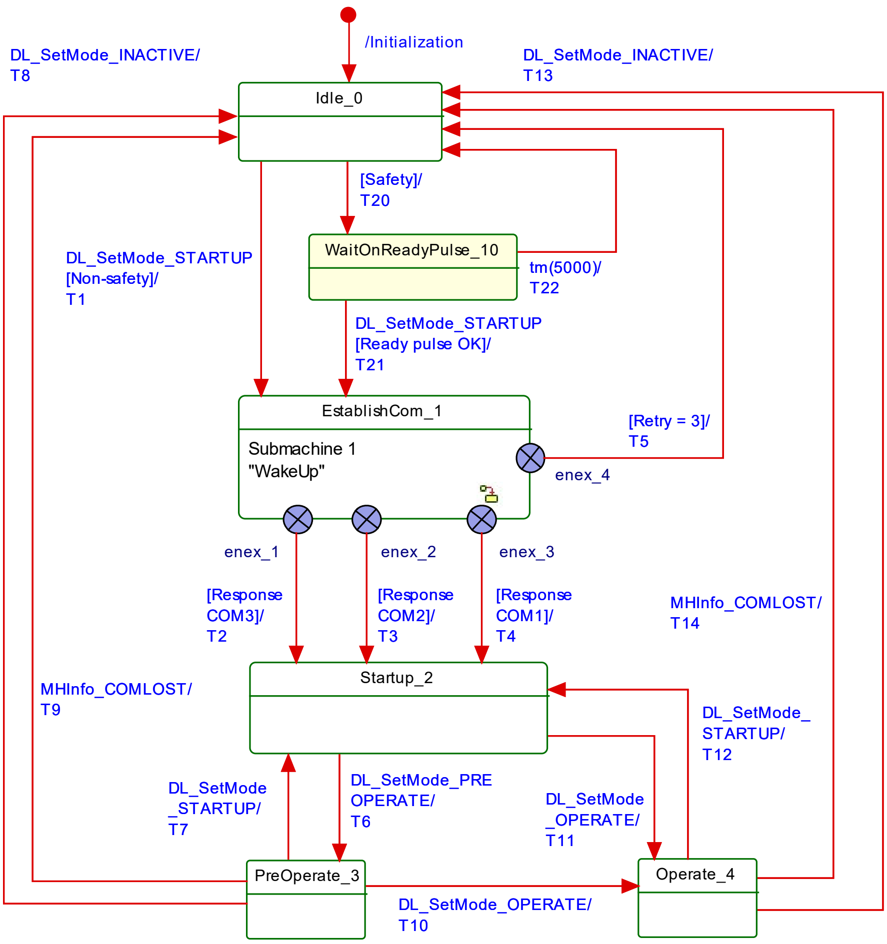


Figure 29 – State machine of the FS-Master DL-mode handler

Table 10 shows the additional state and transitions as well as internal items considering the Ready-pulse feature.

Table 10 – State transition tables of the FS-Master DL-mode handler

STATE NAME	STATE DESCRIPTION
Idle_0 to SM: Retry_9	See Table 42 in IEC 61131-9:2022
WaitOnReadyPulse_10	Waiting on the Ready-pulse from FS-Device. A timer is started with the given value of the parameter FSP_TimeToReady within FSPortConfigList (see A.2.11 and 10.3.4).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1 to T19	*	*	See Table 42 in IEC 61131-9:2022
T20	0	10	This path is taken only if the new configuration parameter "Safety" has been assigned to "SafetyCom" or "MixedSafetyCom" respectively
T21	10	1	Set Retry = 0.
T22	10	0	FS-Master was not able to detect a Ready-pulse within 5 s. It will initiate a power OFF/ON cycle for the FS-Device to retry the Ready-pulse.

INTERNAL ITEMS	TYPE	DEFINITION
MH_xxx to xx_Conf…	Call	See Table 42 in IEC 61131-9:2022
Safety	Guard	New configuration parameter "Safety": either value "SafetyCom" or "MixedSafetyCom"
Ready pulse OK	Guard	Ready pulse detected

7.3 State machine of the FS-Device DL-mode handler

Figure 30 shows the modifications of the FS-Device DL-mode handler versus the Device DL-mode handler in IEC 61131-9.

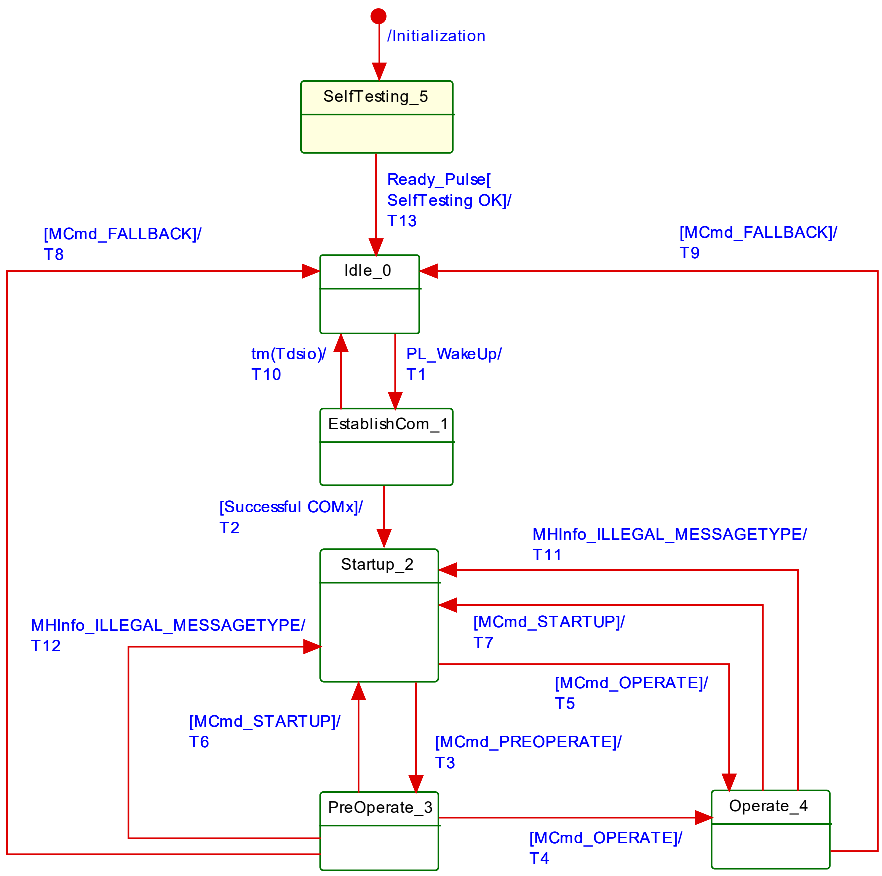


Figure 30 – State machine of the FS-Device DL-mode handler

A new state "SelfTesting_5" considers the requirement for the FS-Device to indicate its readiness for a wake-up procedure after its internal safety self-testing via a test pulse in pin 4. Self-testing may actually take more than the maximum permitted start-up time T_{RDL} of a non-safety Device (see 5.7).

Table 11 – State transition tables of the FS-Device DL-mode handler

STATE NAME	STATE DESCRIPTION
Idle_0 to Operate_4	See Table 43 in IEC 61131-9:2022
SelfTesting_5	Safety check through self-testing of µC, RAM, etc. This may take more than the permitted start-up time T_{RDL} of a non-safety Device (see A.2.11).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1 to T12	*	*	See Table 43 in IEC 61131-9:2022
T13	5	0	Create a signal (Ready_Pulse) on pin 4 for duration of t_{RP}, when self-testing is completed (see t_{2R} in Table 7).

INTERNAL ITEMS	TYPE	DEFINITION
T_{RDL}	Time	See Table 10 in IEC 61131-9:2022
t_{RP}	Time	See Table 7
Self-testing OK	Guard	Self-testing completed

8 Extensions to the Master Configuration Manager (CM)

One part of the integrity measures is a verification record (VerifyRecord) an FS-Master sends to the FS-Device during start-up as explained in 11.7.4 and shown in Figure 57. This requires an extension to the Configuration Manager as shown in Figure 31.

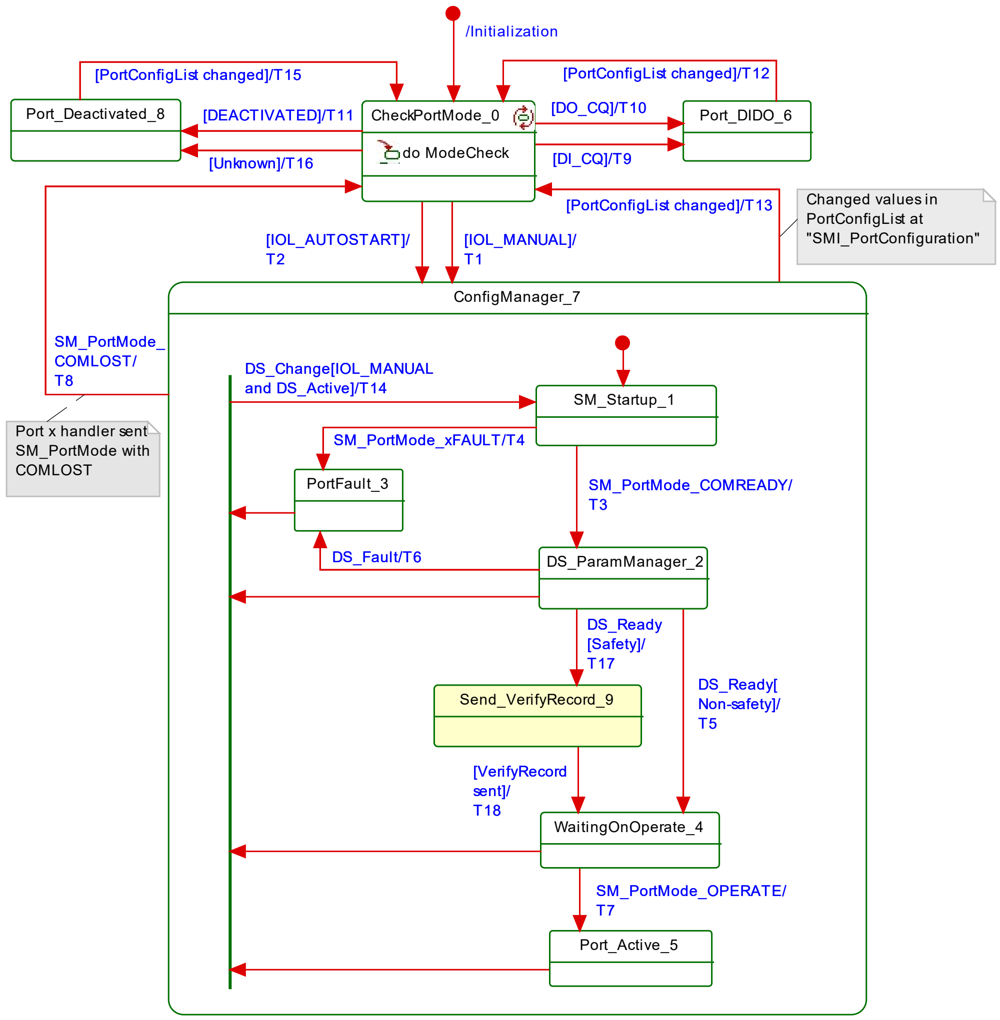


Figure 31 – Extension to the Configuration Manager (VerifyRecord)

A new state "Send_VerifyRecord_9" considers the requirement for the FS-Master to send the VerifyRecord and Table 12 the additional state, transitions, and internal items.

Table 12 – State transition tables of the Configuration Manager

STATE NAME	STATE DESCRIPTION
CheckPortMode_0 to Port_Deactivated_8	See Table 126 in IEC 61131-9:2022
Send_VerifyRecord_9	FS_Master sends its stored FSP_VerifyRecord to "hidden" Index 0x4202 (see A.2.10)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1 to T16	*	*	See Table 43 in IEC 61131-9:2022
T17	2	9	–
T18	9	4	SM_Operate

INTERNAL ITEMS	TYPE	DEFINITION
Safety	Guard	FS-Device in FS mode

INTERNAL ITEMS	TYPE	DEFINITION
Non-safety	Guard	FS-Device in non-FS mode

9 Extensions of the FS-Device

9.1 Principle architecture and models

9.1.1 FS-Device architecture

Figure 32 shows the principle architecture of the FS-Device. It does not include safety measures for implementation such as redundancy for the safety-related parts.

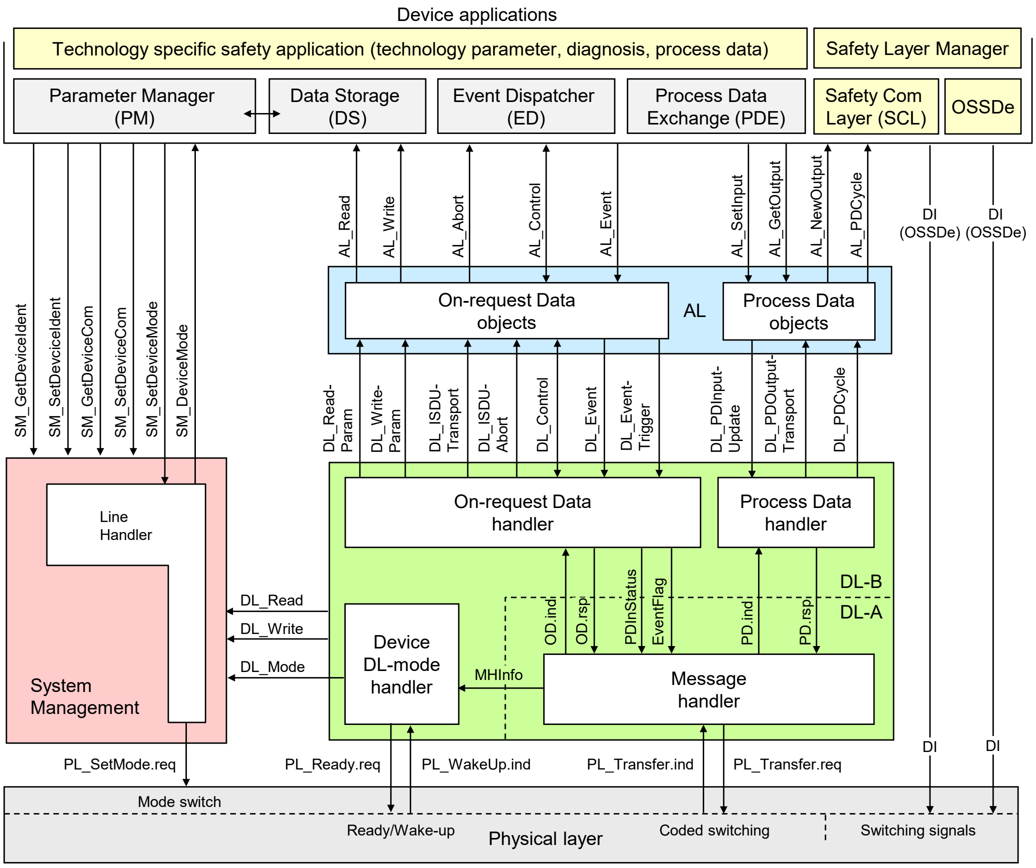


Figure 32 – Principle architecture of the FS-Device

An FS-Device comprises first the technology specific functional safety application. "Emergency switching off" safety devices for example can be designed such that "classic" OSSDe operation or safety communication can be configured. A Safety Layer Manager is responsible for the handling of a safety bit via the OSSDe building block or a safety PDU using the Safety Communication Layer (see Clause 11).

It checks correctness of parameters at each start-up of the FS-Device whenever the FSP_VerifyRecord has been written during PREOPERATE. The safety communication layer (SCL) engine will be started if all parameters are verified to be correct. Otherwise, an error message will be indicated and the SCL remains inactive or stops.

9.1.2 FS-Device model

According to the requirement of mixed NSR and SR parameter and process data, the FS-Device model has been modified and adapted.

That means the FS-Device Index model is split into an NSR and an SR part. Figure 33 shows the areas of concern. The allocation of the SR part ("FSP" parameter) is defined within the IODD of the FS-Device.

During commissioning, the assignment of FSP parameter values take place. These instance values are secured by CRC signatures and transferred as record to the FS-Master and to the FS-Device (see 11.7.4). At each start-up of an FS-Device, the stored entire verification record (VerifyRecord) in the FS-Master is transferred in a diverse manner and the FS-Device can check the locally stored instance parameter values for integrity via comparison and CRC signatures. This check includes technology specific "FST" parameters, which are not transferred at each start-up. The FS-Device displays its FSP parameters at predefined Indices (see Figure 33).

Technology specific parameters (FST) could be handled either in an open manner to a certain extend as standard non-safety parameters (see 11.7.8) or in a protected manner in hidden internal memory (see 11.7.9).

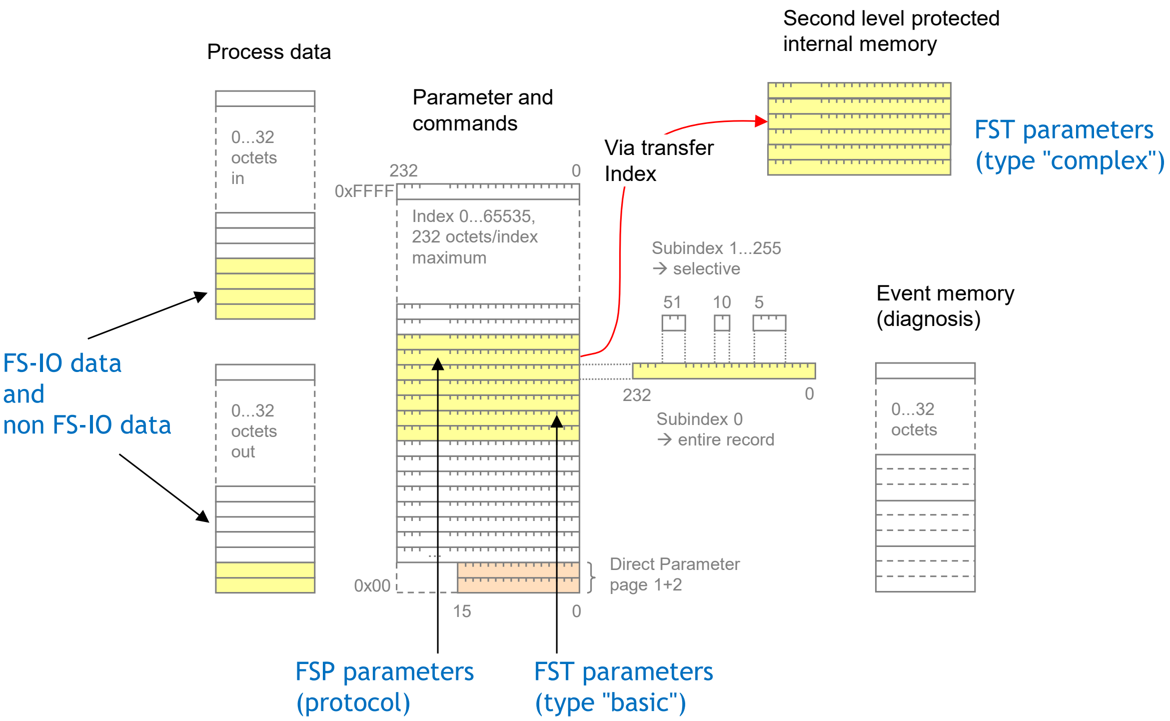


Figure 33 – The FS-Device model

The maximum space for FS-I/O data and non-FS-I/O data to share is 32 octets. The space shall be filled with FS-I/O data first followed by the non-FS-I/O data. The border is variable. Assuming a maximum safety protocol trailer of 6 octets, the maximum possible space for FS-I/O data is 25 octets.

9.2 Parameter Manager (PM)

There are no extensions or modifications of the Parameter Manager required.

9.3 Process Data Exchange (PDE)

Depending on "Safety" configuration, Process Data Exchange takes over or passes FS-Process Data (see 11.4.3 Safety PDU) from/to the Safety Layer Manager.

9.4 Data Storage (DS)

9.4.1 General considerations and extensions including safety

The technology specific (FST) parameters are secured by a particular CRC signature (FSP_TechParCRC) included in the FSP parameter set. Additional authenticity parameters are used in case of FS-Device replacement.

The Data Storage mechanism for FS-Devices is based on the general mechanism for non-safety-related Devices as specified in IEC 61131-9. This version of Data Storage requires that Device Access Lock (Index 0x000C) bit "0" and "1" shall always be unlocked (= "0").

A small extension is required to the Data Storage (DS) state machine of the FS-Device with respect to the "Back-to-box" mechanism (authenticity values = "0"). Transition T8 in IEC 61131-9:2022, Table 100 considers this check as shown in Table 13.

Table 13 – Extension to Data Storage (DS) state machine

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1 to T7	*	*	See Table 100 in IEC 61131-9:2022
T8	2	3	In case of Safety: If back-to-box conditions are incorrect, return 0x8022 – *service temporarily not available – Device control*. If correct: Lock local parameter access, set State_Property = "Upload" or "Download"
T9 to T11	*	*	See Table 100 in IEC 61131-9:2022

9.4.2 Backup levels

Table 14 lists the Data Storage backup levels specified in IEC 61131-9. This Clause describes some specialties to be considered for functional safety.

Table 14 – Data Storage Backup Levels

Backup Level	Behavior
Commissioning ("Disable")	Any change of active parameters within the Device will *not* be copied/saved. Device replacement *without* automatic/semi-automatic Data Storage.
Production ("Backup/Restore")	Changes of active parameters within the Device will be copied/saved. Device replacement *with* automatic/semi-automatic Data Storage supported.
Production ("Restore")	Any change of active parameters within the Device will *not* be copied/saved. If the parameter set is marked to be saved, the "frozen" parameters will be restored by the Master. However, Device replacement *with* automatic/semi-automatic Data Storage *of frozen parameters* is supported.

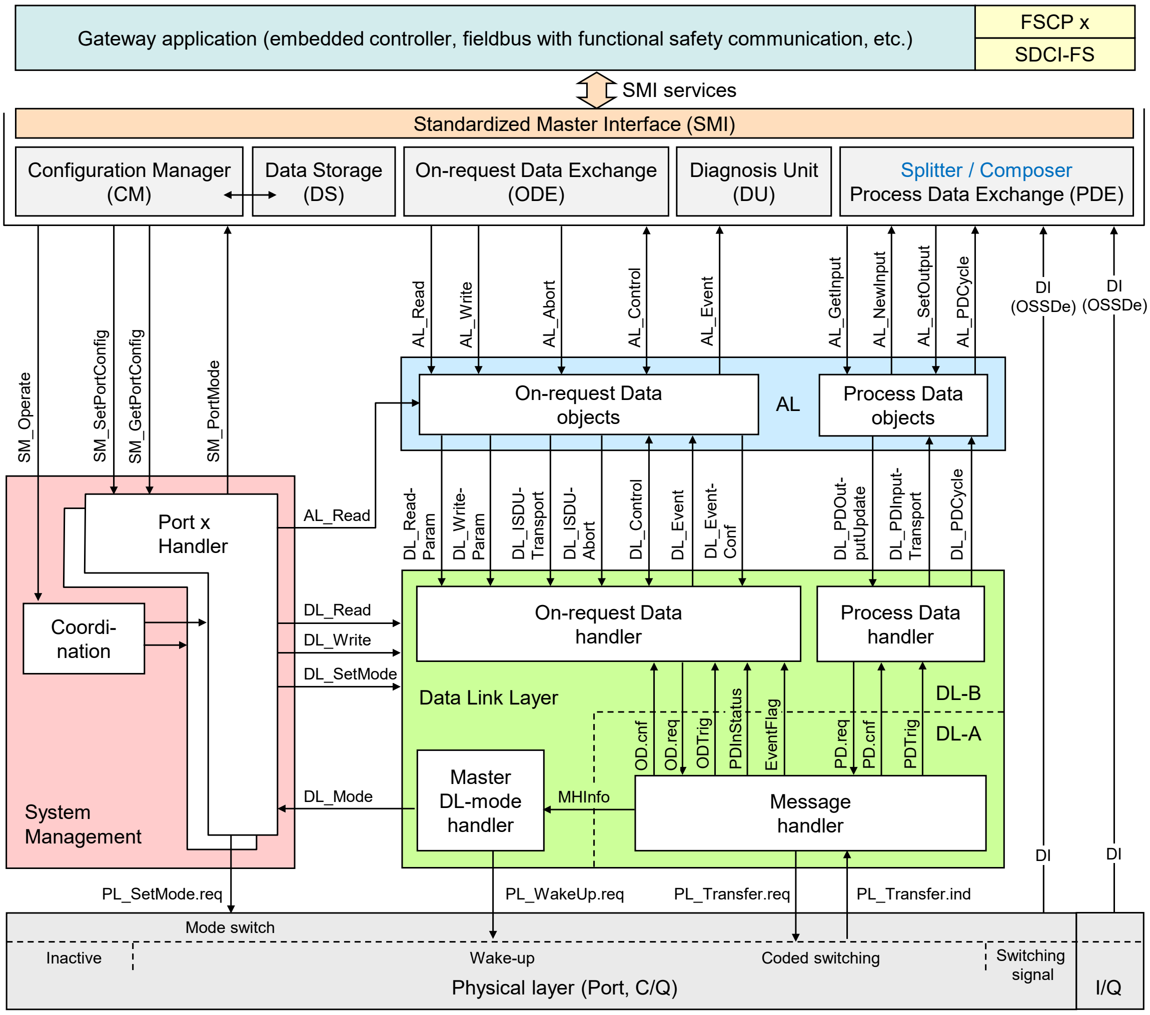
In case of functional safety, commissioning cannot be completed without verification and validation of FSP and FST parameters as well as of entire safety functions according to the relevant safety manuals.

The difference to "Criteria for backing up parameters" in IEC 61131-9:2022 is that in case of local parameter modifications (for example by means of teach-in or touch panel), the FSP_TechParCRC shall be assigned with the help of the FS-Master tool and "Dedicated Tool".

10 Extensions of the FS-Master

10.1 Principle architecture

Figure 34 shows the principle architecture of the FS-Master offering the extended Standard Master Interface (SMI) according to IEC 61131-9:2022. It allows for a stringent separation of the standard Master as "Black Channel" and the functional safety parts of SDCI-FS and an FSCP x that can be "encapsulated" within the Gateway Application layer.



Key yellow marked parts are safety-related; Master part below SMI is "Black channel"

Figure 34 – Principle architecture of the FS-Master

An FS-Master contains the original standard Master ("black channel") except for the Ready-pulse and its handling (see 5.3.3 and 7.2), the second DI at Pin 2 (M12) for OSSDe operation. The Master application Configuration Manager (CM) has been modified to cope with more Port configurations and to send a verification record at each start-up. The Process Data Exchange (PDE/Splitter/Composer) application is now responsible for splitting mixed incoming SR and NSR Process Data respectively for composing outgoing SR and NSR Process Data.

10.2 SMI service extensions

10.2.1 Overview

Basics of SMI services have been introduced in IEC 61131-9. In this document two additional SMI services are specified as shown in Table 15 and in Figure 35: SMI_SPDUIn and SMI_SPDUOut. Both are handling the safety parts (SPDU = complete safety data and safety code) of mixed SR and NSR Process Data. Table 15 provides an overview of the important SMI services used for FS-Masters. The entire set of services can be retrieved from IEC 61131-9.

Table 15 – SMI services used for FS-Master

Service name	Master	ArgBlockID	Remark
SMI_MasterIdentification	R	0x0001	–
SMI_FSMasterAccess	R	0x0100	See 10.2.2
SMI_PortConfiguration	R	0x8100	See IEC 61131-9
SMI_ReadbackPortConfiguration	R	0x8100	See IEC 61131-9

Service name	Master	ArgBlockID	Remark
SMI_PortStatus	R	0x9100	See IEC 61131-9
SMI_DSToParServ	R	0x7000	Data Storage to parameter server
SMI_ParServToDS	R	0x7000	Data Storage from parameter server
SMI_DeviceWrite	R	0x3000	ISDU transport
SMI_DeviceRead	R	0x3001	ISDU transport
SMI_PortPowerOffOn	R	0x7003	–
SMI_DeviceEvent	I	–	–
SMI_PortEvent	I	–	–
SMI_PDIn	R	0x1001	–
SMI_PDOut	R	0x1002	–
SMI_PDInOut	R	0x1003	–
SMI_SPDUIn	R	0x1101	See 10.2.3
SMI_SPDUOut	R	0x1102	See 10.2.4
SMI_FSPDInOut	R	0x1103	See 10.2.5
SMI_PDInIQ	R	0x1FFE	–
SMI_PDOutIQ	R	0x1FFF	–

Key
I Initiator of service
R Receiver (Responder) of service

yellow marked services are either extended or additional ones for SDCI-FS

Figure 35 provides an overview of the important SMI services used for FS-Master, the safety layers within the Gateway and details of the FS-Master applications.

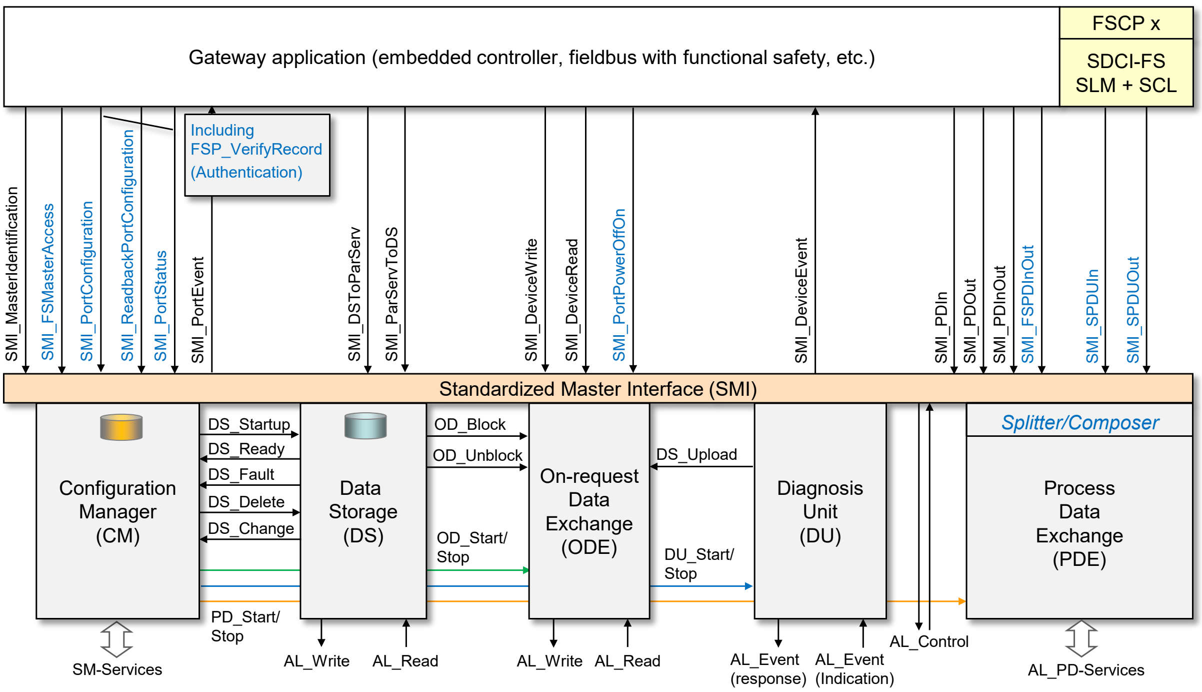


Figure 35 – SMI service extensions

The SMI_MasterIdentification presents as MasterType an FS-Master (= "3" according to IEC 61131-9). The corresponding SMI_FSMasterAccess service provides the FSCP Authenticity codes of the FS-Master being an FSCP device on a safety fieldbus. The SMI services for configuration and Port status are only expanded by using different Arguments (ArgBlocks) as shown in 10.3. By means of the SMI service "SMI_PortConfiguration", for example, the authenticity, protocol, and IO data structure information is transferred to the Configuration Manager and stored there. See 10.4 on how this information is used to accommodate the Safety Communication Layers and to authenticate safety operation. The Port command service "SMI_PortPowerOffOn" is responsible for switching OFF and ON power of a particular Port.

Two new SMI services provide access to the safety parts of a mixed SR and NSR I/O process data structure as shown in 10.2.3, 10.2.4, and 10.5.

10.2.2 SMI_FSMasterAccess

User role and corresponding password can be provided to the FS-Master safety projects and MasterType specific information can be retrieved by this SMI service (see Figure 35).

Table 16 shows the structure of the service following the rules defined in IEC 61131-9.

Table 16 – SMI_FSMasterAccess

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber	M	
ExpArgBlockID (0x0101)	M	
ArgBlockLength	M	
ArgBlock ("FSMasterAccess": 0x0100)	M	
Result (+)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0x0100)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0x0100)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument
The specific parameters of the service request are transmitted in the argument.

ClientID

PortNumber

ExpArgBlockID
This parameter contains the ArgBlockID of "FSCPAuthenticity" (0x0101, see Table 20)

ArgBlockLength
This parameter contains the length of the subsequent ArgBlock

ArgBlock
This parameter contains the ArgBlock "FSMasterAccess" (0x0100, see Table 19)

Result (+):
This selection parameter indicates that the service request has been executed successfully.

ClientID

PortNumber

RefArgBlockID
This parameter contains as reference the ID of the ArgBlock sent by the request (0x0100)

ArgBlockLength
This parameter contains the length of the subsequent ArgBlock

ArgBlock
This parameter contains the ArgBlock associated to the ExpArgBlockID (see Table 20)

Result (-):
This selection parameter indicates that the service request failed

ClientID

PortNumber

RefArgBlockID
This parameter contains as reference the ID of the ArgBlock sent by the request (0x0100)

ArgBlockLength
This parameter contains the length of the "JobError" ArgBlock

ArgBlock
This parameter contains the ArgBlock "JobError" (0xFFFF, see IEC 61131-9)

Permitted values in prioritized order:
PORT_NUM_INVALID (incorrect Port number)
ARGBLOCK_NOT_SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)
ARGBLOCK_INCONSISTENT (incorrect ArgBlock content type)
SERVICE_TEMP_UNAVAILABLE (Master busy)

10.2.3 SMI_SPDUIn

This service allows for cyclically reading Safety Protocol Data Units (SPDU) from an FSInBuffer (see 10.5) and shall only be available for internal use by the SCL. Coding of this SMI service follows the definitions in IEC 61131-9.

The expected ArgBlockID is "0x1101". The ArgBlock is specified in 10.3.6.

10.2.4 SMI_SPDUOut

This service allows for cyclically writing Safety Protocol Data Units (SPDU) to an FSOutBuffer (see 10.5) and shall only be available for internal use by the SCL. Coding of this SMI service follows the definitions in IEC 61131-9.

The ArgBlockID is "0x1102". The ArgBlock is specified in 10.3.7.

10.2.5 SMI_FSPDInOut

This service allows for periodically reading input from an FSInBuffer and InBuffer and output from an FSOutBuffer and OutBuffer (see 10.5). Table 17 shows the structure of the service.

Table 17 – SMI_FSPDInOut

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber	M	
ExpArgBlockID (0x1103)	M	
ArgBlockLength	M	
ArgBlock (VoidBlock: 0xFFF0)	M	
Result (+)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFF0)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

The specific parameters of the service request are transmitted in the argument.

ClientID

PortNumber

ExpArgBlockID

This parameter contains the ArgBlockID of "FSPDInOut" (0x1103, see Table 25)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock

ArgBlock

This parameter contains the ArgBlock "VoidBlock" (0xFFF0)

Result (+):

This selection parameter indicates that the service request has been executed successfully.

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (see Table 25)

Result (-):

This selection parameter indicates that the service request failed

ClientID

PortNumber

RefArgBlockID

This parameter contains as reference the ID of the ArgBlock sent by the request (0xFFF0)

ArgBlockLength

This parameter contains the length of the "JobError" ArgBlock

ArgBlock

This parameter contains the ArgBlock "JobError" (0xFFFF)

Permitted values in prioritized order:
PORT_NUM_INVALID (incorrect Port number)
ARGBLOCK_NOT_SUPPORTED (ArgBlock unknown)
ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)
DEVICE_NOT_IN_OPERATE (Process Data not accessible)

10.3 ArgBlock extensions

10.3.1 Overview

Table 18 shows six new ArgBlock types for FS-Masters: "FSMasterAccess", "FSCPAuthenticity", "FSPortConfigList", "FSPortStatusList", "SPDUIn", "SPDUOut", and "FSPDInOut".

Table 18 – ArgBlock types and ArgBlockIDs

ArgBlock type	ArgBlockID	Remark
FSMasterAccess	0x0100	See 10.3.2
FSCPAuthenticity	0x0101	See 10.3.3

ArgBlock type	ArgBlockID	Remark
PDIn	0x1001	–
PDOut	0x1002	–
PDInOut	0x1003	–
SPDUIn	0x1101	See 10.3.6
SPDUOut	0x1102	See 10.3.7
FSPDInOut	0x1103	See 10.3.8
DS_Data	0x7000	Data Storage object
PortPowerOffOn	0x7003	–
PortConfigList	0x8000	–
FSPortConfigList	0x8100	See 10.3.4
PortStatusList	0x9000	–
FSPortStatusList	0x9100	See 10.3.5
Key (yellow box) yellow marked ArgBlocks are additional ones for SDCI-FS		

10.3.2 FSMasterAccess

The ArgBlock "FSMasterAccess" in Table 19 shows the password for FS-Master access and the corresponding password to reset the entire FS-Master project including the existing password.

Table 19 – FSMasterAccess

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x0100
2	FSMasterPassword	This element contains the password for accessing the entire FS-Master project. Default is 0x0000.	Unsigned32	–
6	FSResetMasterPW	This element contains the password for resetting the entire FS-Master project including the FSMasterPassword. Default is 0x0000.	Unsigned32	–
10	FSUserRole	Reserved. Default: 0x00	Unsigned8	–

10.3.3 FSCPAuthenticity

The ArgBlock "FSCPAuthenticity" in Table 20 shows FSCP authenticity codes assigned to the FS-Master through the upper-level FSCP engineering tool or via DIP switches.

Table 20 – FSCPAuthenticity

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x0101
2	FSP_Authenticity1	FSCP A-Code part1	Unsigned32	–
6	FSP_Authenticity2	FSCP A-Code part2	Unsigned32	–

10.3.4 FSPortConfigList

Table 21 shows the ArgBlockType "FSPortConfigList" suitable for FS-Masters. It considers additional PortModes and expands by Safety PDU lengths, the FSP_VerifyRecord (see 10.3.3 and A.2.10) as well as the FS I/O data structure description (see 11.7.7 and Table A.4).

Table 21 – FSPortConfigList

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x8100
2	PortMode	This element contains the Port mode expected by the SMI client, e.g. gateway application. All modes are mandatory, except OSSDe. They shall be mapped to the Target Modes of "SM_SetPortConfig" (see IEC 61131-9:2022). 0: DEACTIVATED (SM: INACTIVE → Port is deactivated; input and output Process Data are "0"; Master shall not perform activities at this Port) 1: IOL_MANUAL (SM: CFGCOM → Target Mode based on user defined configuration including validation of RID, VID, DID) 2: IOL_AUTOSTART [a] (SM: AUTOCOM →Target Mode w/o configuration and w/o validation of VID/DID; RID gets highest revision the Master is supporting; Validation: NO_CHECK) 3: DI_C/Q (Pin 4 at M12) [b] (SM: DI → Port in input mode SIO) 4: DO_C/Q (Pin 4 at M12) [b] (SM: DO → Port in output mode SIO) 5 to 48: Reserved for future versions 49: SAFETYCOM (implying IOL_MANUAL behavior) 50: MIXEDSAFETYCOM (implying IOL_MANUAL behavior) 51: OSSDE (Pin 2 + Pin 4 at M12) (Not for FS-Masters of type Port Class B; Values in offset 14 to 35 are don't care; SPDUInLength in offset 38 = 1 octet; value in offset 39 is don't care) 52 to 96: Reserved for extensions such as Safety or Wireless) 97 to 255: Manufacturer specific	Unsigned8	0 to 0xFF
3	Validation&Backup	This element contains the InspectionLevel to be performed by the Device and the Backup/Restore behavior. 0: No Device check 1: Type compatible Device V1.0 2: Type compatible Device V1.1 3: Type compatible Device V1.1, Backup + Restore 4: Type compatible Device V1.1, Restore 5 to 255: Reserved	Unsigned8	0 to 0xFF
4	I/Q behavior (Manufacturer or profile specific)	This element defines the behavior of the I/Q signal (Pin2 at M12 connector). All assignments are "don't care" if PortMode is chosen to be OSSDE. 0: Disabled 1: Digital Input 2 to 4: Reserved 5: Power 2 (Port Class B) 6 to 255: Reserved	Unsigned8	0 to 0xFF
5	PortCycleTime	This element contains the Port cycle time expected by the SMI client. AFAP is default. They shall be mapped to the ConfiguredCycleTime of "SM_SetPortConfig"	Unsigned8	0 to 0xFF

Offset	Element name	Definition	Data type	Values
		0: AFAP (As fast as possible – SM: FreeRunning → Port cycle timing is not restricted. Default value in Port mode IOL_MANUAL) 1 to 255: TIME (SM: For coding see Table B.3 in IEC 61131-9. Device shall achieve the indicated Port cycle time. An error shall be created if this value is below MinCycleTime of the Device or in case of other misfits)		
6	VendorID	This element contains the 2 octets long VendorID expected by the SMI client	Unsigned16	1 to FFFF
8	DeviceID	This element contains the 3 octets long DeviceID expected by the SMI client	Unsigned32	1 to FFFFFF
12	FSP_TimeToReady	This element provides the time from power-up to the Ready pulse of the FS-Device such that the FS-Master knows how long to wait on it. Default maximum time is 5 s (see A.2.11)	Unsigned16	1 to 7FFF (see Table A.1)
14	FSP_MinShutDown Time	This element provides the minimum time for shut down of the FS-Device after Port power off prior to a restart	Unsigned16	64 to 03E8 (see Table A.1)
16	FSP_Authenticity1	FSCP A-Code part1 (see IEC 61784-3 series)	Unsigned32	–
20	FSP_Authenticity2	FSCP A-Code part2 (see IEC 61784-3 series)	Unsigned32	–
24	FSP_Port	Port number	Unsigned8	1 to 0xFF
25	FSP_AuthentCRC	CRC signature across complete authentication	Unsigned16	–
27	FSP_ProtVersion	Version of the SDCI-FS protocol	Unsigned8	1 to 0xFF
28	FSP_ProtMode	SDCI-FS protocol mode	Unsigned8	–
29	FSP_WatchdogTime	Watchdog time of FS-Master and FS-Device	Unsigned16	1 to FFFF
31	FSP_IO_StructCRC	CRC signature across FS IO data description	Unsigned16	–
33	FSP_TechParCRC	CRC signature across technology parameter	Unsigned32	–
37	FSP_ProtParCRC	CRC signature across protocol parameter	Unsigned16	–
39	IO_DescVersion	Version of this generic structure description	Unsigned8	1
40	SPDUInLength	OSSDe data (1 octet) or length of incoming SPDU (*m*); see 10.5 and Table A.4	Unsigned8	1 or 5 to (32 – *n)* octets
41	TotalOfInBits	Set of input BooleanT (bits)	Unsigned8	0 to 0xFF
42	TotalOfInOctets	Set of input BooleanT (octets)	Unsigned8	–
43	TotalOfInInt16	Input IntegerT(16)	Unsigned8	–
44	TotalOfInInt32	Input IntegerT(32)	Unsigned8	–
45	SPDUOutLength	Length of outgoing SPDU (*o*); see 10.5 and Table A.4	Unsigned8	5 to (32 – *p)* octets
46	TotalOfOutBits	Set of output BooleanT (bits)	Unsigned8	0 to 0xFF
47	TotalOfOutOctets	Set of output BooleanT (octets)	Unsigned8	–
48	TotalOfOutInt16	Output IntegerT(16)	Unsigned8	–
49	TotalOfOutInt32	Output IntegerT(32)	Unsigned8	–

[a] In PortMode "IOL_Autostart" parameters VendorID, DeviceID, and Validation&Backup are treated don't care.

[b] In PortModes "DI_C/Q" and "DO_C/Q" all parameters are don't care except for "InputDataLength" and "Output DataLength".

1495

10.3.5 FSPortStatusList

Table 22 shows the ArgBlockType "FSPortStatusList" suitable for FS-Masters. It allows only for the status report of the "Black Channel" part of the FS-Master. Content of "FSPortStatusInfo" shall be derived from "PortMode" in IEC 61131-9).

Table 22 – FSPortStatusList

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x9100
2	PortStatusInfo	This element contains status information of the Port. 0: NO_DEVICE No communication (COMLOST). However, Port configuration IOL_MANUAL or IOL_AUTOSTART was set. 1: DEACTIVATED Port configuration DEACTIVATED is set. 2: PORT_DIAG This value to be set If a DiagEntry indicates an upcoming diagnosis of the Port during startup, validation, and Data Storage (group error). Device is in PREOPERATE and DiagEntry contains the diagnosis cause. 3: Reserved 4: OPERATE This value to be set if the Device is in OPERATE, even in case of Device error. 5: DI_C/Q Port configuration "DI" is set. 6: DO_C/Q Port configuration "DO" is set. 7: OSSDE (DI at C/Q and I/Q) 8: SCL_ENABLED (Port ready for safety data exchange) 9 to 253: Reserved 254: PORT_POWER_OFF Shutdown of Port is active caused by SMI_PortPowerOffON 255: NOT_AVAILABLE (Port status currently not available)	Unsigned8 (enum)	0 to 0xFF
3	PortQualityInfo	This element contains status information on Process Data Bit0: 0 = VALID 1 = INVALID Bit1: 0 = PDOUTVALID 1 = PDOUTINVALID Bit2 to Bit7: Reserved	Unsigned8	–
4	RevisionID	This element contains information of the SDCI protocol revision of the Device 0: NOT_DETECTED (No communication at that Port) <>0: Copied from Direct parameters page, address 4	Unsigned8	0 to 0xFF
5	TransmissionRate	This element contains information on the effective Port transmission rate. 0: NOT_DETECTED (No communication at that Port) 1: COM1 (transmission rate 4,8 kbit/s) 2: COM2 (transmission rate 38,4 kbit/s) 3: COM3	Unsigned8	0 to 0xFF

Offset	Element name	Definition	Data type	Values
		(transmission rate 230,4 kbit/s) 4 to 255: Reserved for future use		
6	MasterCycleTime	This element contains information on the Master cycle time.	Unsigned8	–
7	InputDataLength	This element contains the input data length as number of octets of the Device provided by the PDIn service	Unsigned8	0 to 0x20
8	OutputDataLength	This element contains the output data length as number of octets for the Device accepted by the PDOut service	Unsigned8	0 to 0x20
9	VendorID	This element contains the 2 octets long VendorID expected by the SMI client	Unsigned16	1 to 0xFFFF
11	DeviceID	This element contains the 3 octets long DeviceID expected by the SMI client	Unsigned32	1 to FFFFFF
15	NumberOfDiags	This element contains the number x of diagnosis entries (DiagEntry0 to DiagEntryx	Unsigned8	0 to 0xFF
16	DiagEntry0	This element contains the "EventQualifier" and "EventCode" of a diagnosis (Event).	Struct Unsigned8/16	–
19	DiagEntry1	Further entries up to x if applicable…	…	–

10.3.6 SPDUIn

Table 23 shows the structure of the ArgBlock "SPDUIn" as illustrated in 10.5.

Table 23 – SPDUIn

Offset	Element name	Definition	Data type
0	ArgBlockID	0x1101	Unsigned16
2	SPDUIn0	Safety Protocol Data Unit in (octet 0)	Unsigned8
3	SPDUIn1	Safety Protocol Data Unit in (octet 1)	Unsigned8
		…	
SPDUInLength + 2	SPDUIn*m*	Safety Protocol Data Unit in (octet *m*)	Unsigned8

10.3.7 SPDUOut

Table 24 shows the structure of the ArgBlock "SPDUOut" as illustrated in 10.5.

Table 24 – SPDUOut

Offset	Element name	Definition	Data type
0	ArgBlockID	0x1102	Unsigned16
2	SPDUOut0	Safety Protocol Data Unit out (octet 0)	Unsigned8
3	SPDUOut1	Safety Protocol Data Unit out (octet 1)	Unsigned8
		…	
SPDUOutLength + 2	SPDUOut*o*	Safety Protocol Data Unit out (octet *o*)	Unsigned8

10.3.8 FSPDInOut

Table 25 shows the structure of the ArgBlock "FSPDInOut" as illustrated in 10.5.

Table 25 – FSPDInOut

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x1103
2	PQI	Port Qualifier Information	Unsigned8	–
3	OE	Output Enable	Unsigned8	–
4	InputDataLength ($m+n+2$)	This element contains the length of the Device's input Process Data contained in the following elements (SR + NSR).	Unsigned8	0 to 0x20
5	SPDUIn0	Safety Protocol Data Unit in (octet 0)	Unsigned8	0 to 0xFF
6	SPDUIn1	Safety Protocol Data Unit in (octet 1)	Unsigned8	0 to 0xFF
…				
$(m+1) + 4$	SPDUInm	Safety Protocol Data Unit in (octet m)	Unsigned8	0 to 0xFF
$(m+1) + 5$	PDI0	Input Process Data (octet 0)	Unsigned8	0 to 0xFF
$(m+1) + 6$	PDI1	Input Process Data (octet 1)	Unsigned8	0 to 0xFF
…				
$(m+1) + (n+1) + 4$	PDIn	Input Process Data (octet n)	Unsigned8	0 to 0xFF
$(m+1) + (n+1) + 5$	OutputDataLength ($o+p+2$)	This element contains the length of the Device's output Process Data contained in the following elements (SR + NSR).	Unsigned8	0 to 0x20
$(m+1) + (n+1) + 6$	SPDUOut0	Safety Protocol Data Unit out (octet 0)	Unsigned8	0 to 0xFF
$(m+1) + (n+1) + 7$	SPDUOut1	Safety Protocol Data Unit out (octet 1)	Unsigned8	0 to 0xFF
…				
$(m+1) + (n+1) + (o+1) + 5$	SPDUOut$o$	Safety Protocol Data Unit out (octet o)	Unsigned8	0 to 0xFF
$(m+1) + (n+1) + (o+1) + 6$	PDO0	Output Process Data (octet 0)	Unsigned8	0 to 0xFF
$(m+1) + (n+1) + (o+1) + 7$	PDO1	Output Process Data (octet 1)	Unsigned8	0 to 0xFF
…				
$(m+1) + (n+1) + (o+1) + (p+1) + 5$	PDOp	Output Process Data (octet p)	Unsigned8	0 to 0xFF

10.4 Safety Layer Manager (SLM)

10.4.1 Purpose

The Safety Layer Manager takes care of the safety PDU, whenever safety communication has been configured or of one safety bit, whenever OSSDe has been configured for a particular Port. It uses SMI services for this purpose as specified in 10.2.3 and 10.2.4.

It holds the FSP parameters consisting of the authenticity record and the protocol record (see 11.7.4) as well as of the FS I/O structure description (see Table A.1 and E.5.5) for the FS_IO_Data_Mapper.

After verification of all parameters, the safety communication layer (SCL) engine will be started and PortStatusInfo will be set to "8" (SCL_ENABLED) (see 10.3.5).

SLM or SCL respectively use the Diagnosis Unit (DU) in Figure 34 to propagate the PortEventCodes of Table B.2.

10.4.2 FS_PortModes

The FS-Master shall support the following PortModes of standard NSR Masters:

- DEACTIVATED

- IOL_MANUAL (basis of SCL operation)
- IOL_AUTOSTART (usually only in case of totally unknown connected Devices)
- DI_C/Q
- DO_C/Q (Devices and FS-Devices are short-circuit proof)

In addition, the FS-Master shall support two FS_PortModes and one optionally:

SAFETYCOM
This setting enables pure safety communication without NSR Process Data of a Port.

MIXEDSAFETYCOM
This setting enables safety communication of SR and NSR Process Data of a Port.

OSSDE
This setting enables OSSDe operation of a Port. This mode is optional.

All these PortModes can be set up via the SMI_PortConfiguration (see 10.2.1) and the ArgBlock "FSPortConfigList" (see 10.3.4).

10.4.3 FSP parameter

10.4.3.1 FSP parameter use cases

Figure 36 illustrates some use cases related to the FSP parameters (see A.1).

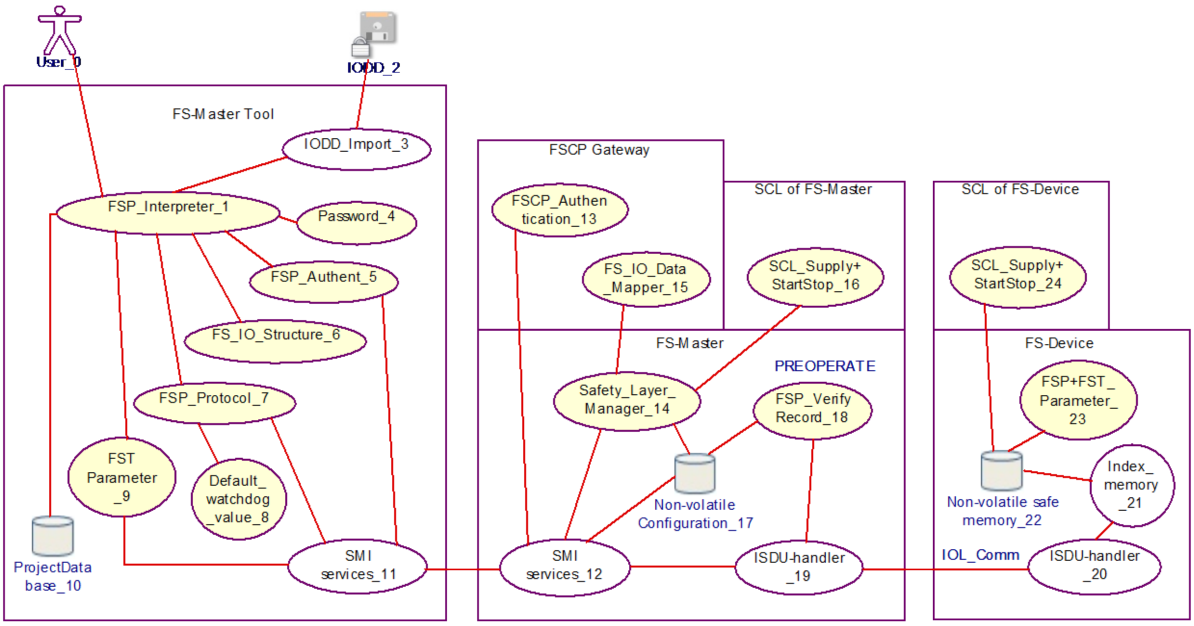


Figure 36 – FSP parameter use cases

Table 26 shows a listing of the items in Figure 36 and references clauses within this document or IEC 61131-9.

Table 26 – Use case reference table

No.	Item	Type	Reference	Remarks
0	User	Roles: - Observer - Maintenance - Specialist	–	Responsibility of the software tool manufacturer
1	FSP_Interpreter	GUI-functions	E.g. Figure 58	–
2	IODD (secured)	Device description	Annex E	–
3	IODD_Import	Activity	Annex E	–

No.	Item	Type	Reference	Remarks
4	Password	Activity	10.4.3.2	Access protection
5	FSP_Authent	Activity	11.7.5	–
6	FS_IO_Structure	FS I/O description	Clause A.1	–
7	FSP_Protocol	Activity	11.7.6	–
8	Default_watchdog_value	Activity	A.2.6	–
9	FST Parameter	Activity	–	–
10	ProjectDatabase	FS-Master tool	–	Proprietary
11	Standardized Master Interface (SMI)	Communication	10.4.3.1	–
12	Standardized Master Interface (SMI)	Communication	10.4.3.1	–
13	FSCP_Authentication	Activity	11.7.5	–
14	Safety_Layer_Manager	Activity	10.4	–
15	FS_IO_Data_Mapper	Gateway application	12.1	FSCP Integration
16	SCL_Supply+StartStop	FS-Master SCL	11.5.2	–
17	Non-volatile memory	FS-Master	–	Implementation
18	FSP_VerifyRecord	Verification	11.7.4	–
19	ISDU-Handler	FS-Master DL	IEC 61131-9	Standard SDCI
20	ISDU-Handler	FS-Device DL	IEC 61131-9	Standard SDCI
21	Index_memory	Activity	IEC 61131-9	Standard SDCI
22	Non-volatile memory	FS-Device	–	Implementation
23	FSP+FST parameter	Activities	–	–
24	SCL_Supply+StartStop	FS-Device SCL	11.5.3	–

In the following, a typical parameterization session of a project in the ProjectDatabase is described, where a new FS-Device is planned, configured, and parameterized for a particular Port. After installation of IODD and associated Dedicated Tool, the user of an FS-Master tool opens the parameter tab page (see illustration in Figure 58). After entry of the password for safety projects (see 10.4.3.2), FSP parameters are enabled to be displayed and Dedicated Tools are enabled to be launched.

When online, the FS-Master tool uses the Standardized Master Interface (SMI) to the FS-Master specified in IEC 61131-9. Any transmission error (see Table 27) can falsify the message bits and thus, each FSP parameter record is secured by CRC signature.

The choice of the SMI service call technology is the responsibility of the respective integration into a fieldbus (see IEC 61131-9).

The *authenticity parameter* values carry "0" as default within the IODD of an FS-Device. FS-Master tool acquires FSCP Authenticity values with the help of the SMI_FSMasterAccess service (see 10.2.2) and suggests these as actual values. For details see 10.4.3.3.

The IODD contains the *I/O data structure description* of the safety Process Data as a record secured by CRC signature (see A.2.7 and E.5.6). This information is used for the mapping to FSCP I/O data and checked by FS-Device (see 11.7.7).

Most of the *protocol parameter* values are preset by default values provided by the FS-Device manufacturer within the IODD, except for the value of FSP_TechParCRC, which has a particular responsibility. A value of "0" followed by a Port power OFF/ON means commissioning/test (see Annex G). The consequences are:

- only verification of FSP_TechParCRC = 0 is required to start SCL;
- changes of FST parameters become effective right upon acceptance by the FS-Device;
- no Data Storage backup.

From now on the SDCI-FS system can run in "monitored operational mode". That means personnel are required to watch the machine.

The user can now enter and test the technology specific parameters (see illustration in Figure 58). After verification and validation, the user launches the Dedicated Tool, confirms the value assignments, and transfers the CRC signature to the FSP_TechParCRC field. The corresponding SMI_PortConfiguration cares for the FSP_VerifyRecord within the FS-Master. With an FSP_TechParCRC value of ≠ "0", the system can be armed:

- Data Storage;
- Verification of authenticity, protocol, and technology parameters, as well as IO data description at start-up.

After parameter assignment, the FSP and FST parameter instance values can be stored in the ProjectDatabase.

Another Port power OFF/ON will cause the FS-Device to perform safety selftests prior to communication and the FS-Master to transmit the FSP_VerifyRecord to the FS-Device. Its Safety Layer Manager verifies all parameters, passes relevant protocol parameters, and launches the SCL. In case of mismatch a corresponding Event is activated and the SCL will not operate.

The SLM propagates the I/O structure description to the FS_IO_DataMapper. The FSP_VerifyRecord is propagated to the local FS-Master safety communication layer (SCL). It verifies all parameters, passes relevant protocol parameters, launches the SCL, and sets PortStatusInfo to "8" (SCL_ENABLED). In case of mismatch a corresponding Event is activated and the SCL will not operate.

10.4.3.2 Protection

An FS-Master tool creates and maintains safety projects and the FS-Master stores safety parameters within configuration data. Both require protection from easy manipulation. FS-Master tool can use password parameters within the SMI_FSMasterAccess service (10.2.2) for that purpose. Usage depends on the security concept of the upper-level system and on the inheritance concept of password protection. It is optional.

Dedicated Tools may have password mechanisms for their FS-Devices independent from the FS-Master (see 10.2.2).

10.4.3.3 FSP authenticity parameter record

FSP authenticity parameters are specified in Clause A.1. The FSP authenticity record includes the FSCP authenticity code, a Port number, and a CRC signature. An FS-Master tool shall always update the CRC signature when changes occur and only write entire consistent records.

For stand-alone FS-Masters the entry of unique and unambiguous values per FS-Master is required per machine or production center if there is a possibility to misconnect FS-Devices amongst different FS-Masters.

10.4.3.4 FSP protocol parameter record

FSP protocol parameters are specified in Clause A.1. Manufacturer/vendor presets values and defines ranges within the IODD for protocol version and mode, Port mode, watchdog, and TechParCRC.

Manufacturer/vendor shall determine the preset value for the watchdog timer considering the FS-Device response time at the indicated transmission rate (COMx). The FS-Master tool can calculate and suggest a value based on the performance data of the used FS-Master and on the preset value from the IODD.

An FS-Master tool shall always update the CRC signature when changes occur and only write entire consistent records.

10.4.3.5 FS I/O data structure description

With the help of this information, the mapping process within the FSCP gateway can be controlled or monitored (see 11.7.7 and A.2.7).

The FS-Device shall check the validity of its implemented safety PDin/PDout structure via the FSP_IO_StructCRC signature provided by the FSP_VerifyRecord.

10.4.3.6 Verification record

The FS-Master takes the FSP_VerifyRecord from the SMI_PortConfiguration service (see 10.3.3 and 11.7.4).

10.5 Process Data Exchange (PDE)

The FS-Master application Process Data Exchange (PDE) provides additional features called "Splitter" and "Composer".

Figure 37 shows the mechanism of splitting the SPDUIn part and the Input Data from the complete SR and NSR data. The SR part is stored within an FSInBuffer and the NSR part within the InBuffer already specified in IEC 61131-9.

In case of PortConfiguration "OSSDE", the signal status of C/Q is stored as "OSSDe1" in Bit "0" of octet "0", and signal status of I/Q is stored as "OSSDe2" in Bit "1" of octet "0".

See IEC 61131-9 for definitions of "PQI" and "PQ". PDIn valid/invalid only refer to non-safety data.

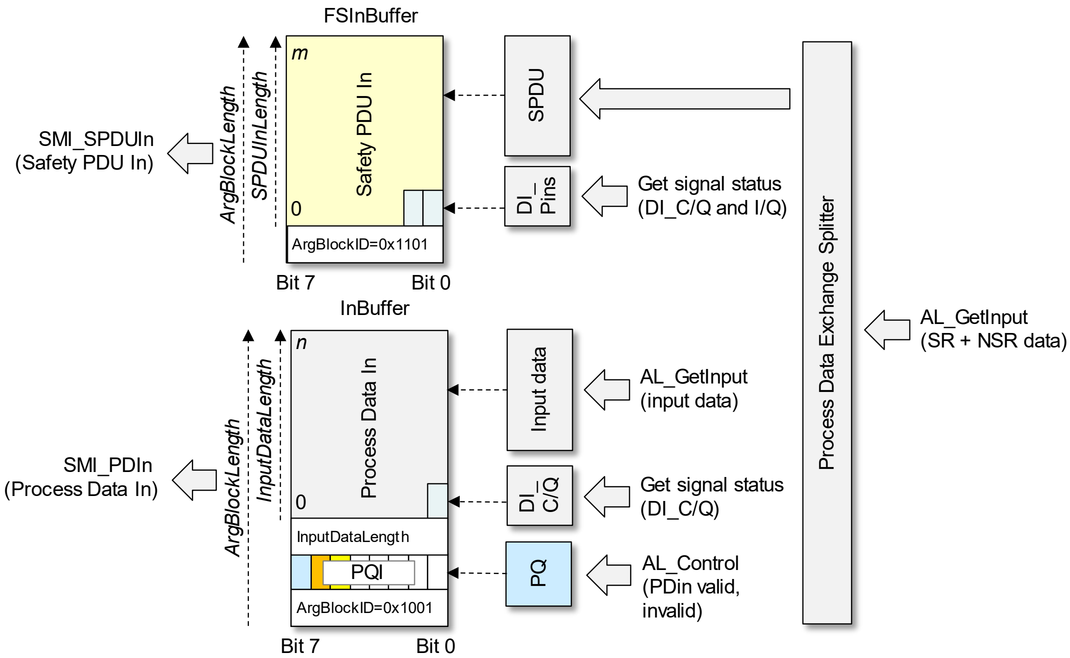


Figure 37 – PDE Splitter

Figure 38 shows the mechanism of composing the complete SR and NSR data. Both are prepared for the AL_SetOutput service out of the SPDU Out part from the FSOutBuffer and out of the Process Data Out from the OutBuffer already specified in IEC 61131-9.

See IEC 61131-9 for definitions of "OE" and "OE detect".

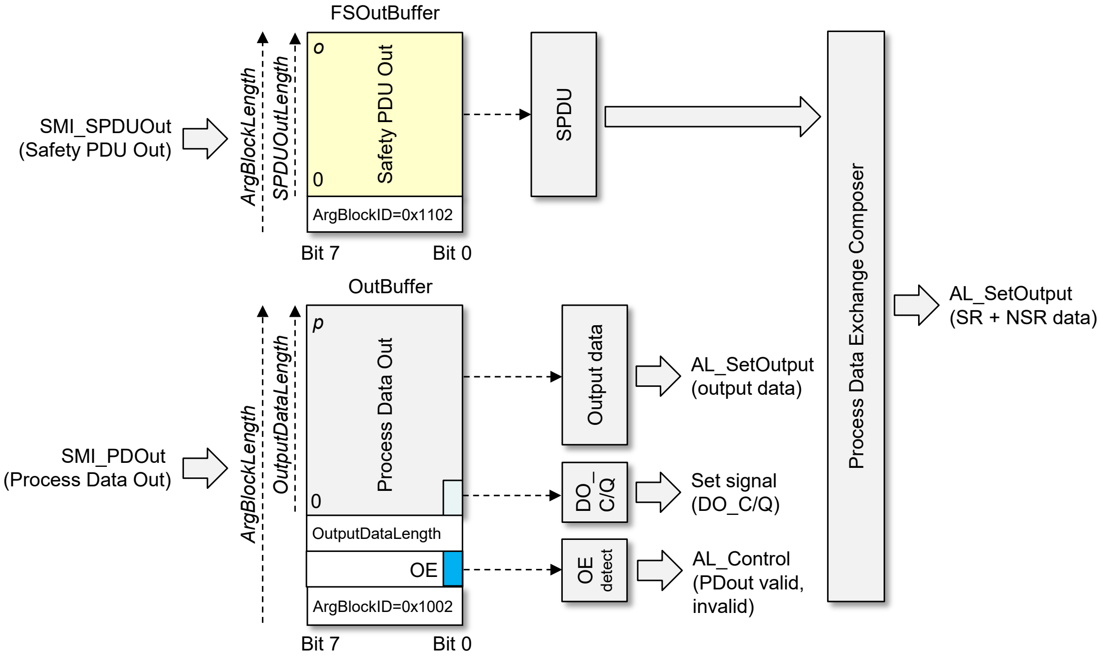


Figure 38 – PDE Composer

10.6 Data Storage (DS)

Data Storage for the purpose of parameter backup and Device/FS-Device replacement is specified in IEC 61131-9. A brief description and adaptations for SDCI-FS can be found in 9.4.

11 Safety communication layer (SCL)

11.1 Functional requirements

The functional requirements for SDCI-FS are laid down in [9]. Main application area is "safety for machinery". Usually this means operational stop of a machine until clearance or repair and restart only after an operator acknowledgement. Primarily relevant are IEC 62061 and ISO 13849.

Other major requirements are suitability for up to SIL3/PLe safety functions, Port specific passivation, and parameterization using Dedicated Tools. Safety measures and residual error rates for timeliness, authenticity, and data integrity ("TADI") of safety messages (safety PDUs) shall be compliant with IEC 61784-3:2021.

11.2 Communication errors and safety measures

The point-to-point communication basis of SDCI allows for a very lean protocol type and a hardware independent safety communication layer stack with a small memory footprint. Table 27 shows the communication errors to be considered and the chosen safety measures:

- (Sequence) counter / inverted counter;
- Watchdog timer and receipt messages;
- Connection validation at commissioning, start-up, and repair, and
- Cyclic redundancy check for data integrity.

Table 27 – Communication errors and safety measures

Communication error	Protocol safety measures			
	Counter/Inverted counter	Timeout with receipt	PortNum + Connection validation [a]	Cyclic redundancy check (CRC)
Corruption	–	–	–	X
Unintended repetition	X	X	–	–
Incorrect sequence	X	–	–	–
Loss	X	X	–	–
Unacceptable delay	–	X	–	–
Insertion	X	–	–	–
Masquerade	–	–	X	X
Addressing	–	–	X	–
Loop-back of messages	X	–	X	–
a Connection validation comprises an FSCP authenticity (see A.2.1) and the FS-Master Port number.				

It is assumed, that there are no storing elements within the SDCI communication path between FS-Master and FS-Device. Thus, a three-bit counter is sufficient as a safety measure. A value 0b000 of this counter on FS-Master side indicates a start or reset position of this counter. In cyclic mode it counts to 0b111 and returns to 0b001.

The message "send" and "receive" concept of SDCI allows for a simple watchdog timer and message receipt safety measure concept corresponding to the "de-energize to trip" principle.

It is assumed that an FS-Master is the owner of a functional safety connection ID of the upper-level FSCP communication system (FSCP authenticity) similar to an FS-DI-Module within a remote I/O. A customer is required to perform a validation procedure, whenever a change occurred with the connected safety devices. SDCI-FS relies on such a concept. Additionally, due to the standard "data storage" mechanism of SDCI and the functional safety nature of the FS-Master, it is possible to provide a more convenient mechanism.

A CRC signature is used for the data integrity check of transmitted safety PDUs. Two options can be configured. A 16-bit CRC signature for safety I/O data up to 3 octets or a 32-bit CRC signature for safety IO data up to 25 octets can be chosen.

11.3 SCL services

11.3.1 Positioning of safety communication layers (SCL)

Figure 39 shows the positioning of the SDCI-FS Safety Communication Layer (SCL).

For each Port with a connected FS-Device an instance of the SCL is required. The SCLs are exchanging safety PDUs consisting of output Process Data (PDout) together with safety code to the FS-Device and input Process Data (PDin) together with safety code from the FS-Device. The SCLs are using standard SDCI communication as a "black channel".

Sufficient availability through for example correct installations, low-noise power supplies, and low interferences are preconditions for this "black channel" to avoid so-called nuisance/spurious trips. These trips cause production stops and subsequently may cause management to remove safety equipment.

This document does not specify implementation related safety measures such as redundant microcontrollers, RAM testing, etc. It is the responsibility of the manufacturer/vendor to take appropriate measures against component failures or errors according to IEC 61508 (all parts).

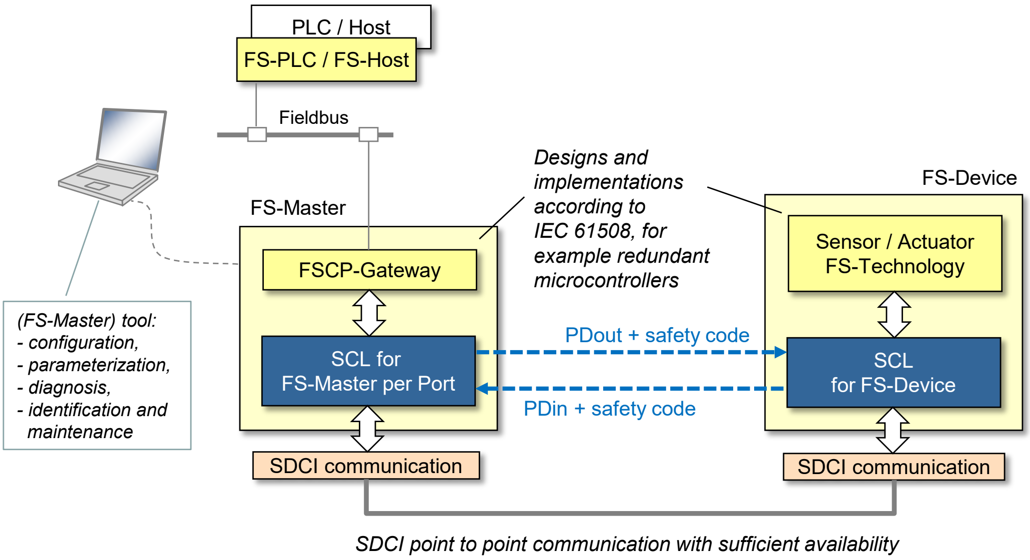


Figure 39 – Positioning of the SDCI-FS Safety Communication Layer (SCL)

11.3.2 FS-Master SCL services

SDCI-FS applications include (but are not limited to) connections to upper-level FSCP fieldbus systems. FSCPs usually also provide safety codes and control/monitoring services (signals).

Figure 40 shows the FS-Master Safety Communication Layer signals (services) depicted by arrows in the upper part of the figure. For each FSCP to be connected to, a mapping or emulation of corresponding SCL services is required.

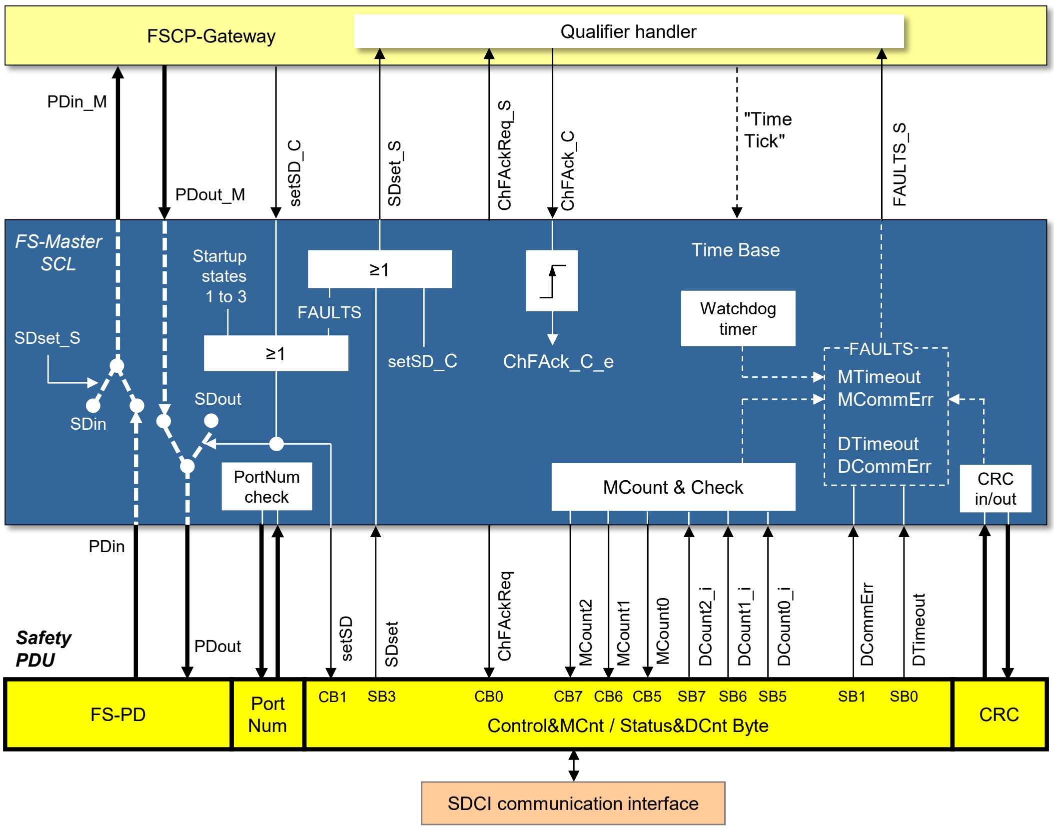


Figure 40 – FS-Master Safety Communication Layer services

A service name carries either an extension "_C" (Control), if it controls the safety communication activities or an extension "_S" (Status), if it is reporting on the activities.

Some of the service names correspond to the signal names of the Control Byte or Status Byte (see lower part of the figure and 11.4.5). That means they are correlated, but there is some control logic of the SCL in between. This control logic is time discrete and not continuous even if it is depicted as logic OR ("≥") box. Definitive are the state charts and the state transition tables of the SCL (see 11.5.2).

The following services in Table 28 shall be available to the FSCP gateway or to a programmer of an FS-Master system.

Table 28 – SCL services of FS-Master

Service/signal	Definition
PDin_M, PDout_M	These services carry the actual Process Data values, both SDin (all bits "0") and SDout (all bits "0") in case of safe state or the real process values from or to the FS-Device.
SDin, SDout	These services carry Process Data values all zero.
setSD_C	In case of emergency, safety control programs usually set output Process Data (PDout_M) for an actuator to "0". However, in some cases, for example burner ventilators, shut down may not be a safe state. This service, if set to "1", is additional information allowing an FS-Device to establish a safe state no matter what the values of Process Data are. Independent from PDout_M, this service causes the SCL to send SDout values to the FS-Device and to send SDin to the FSCP gateway (PDin_M) via SDset_S.
SDset_S	This service, if set to "1", causes the qualifier handler to set the qualifier bit for the Process Data of the connected FS-Device (see 11.11.4). In addition, it causes the SCL to send SDin to the FSCP gateway (PDin_M).
ChFAckReq_S	The FS-Master SCL sets this service to "1" in case of disappeared FAULTS or FS-Master timeouts. It shall be propagated via FSCP and indicated to the operator.
ChFAck_C	After check-up and/or repair, the operator is requested to acknowledge a "ChFAckReq_S" service via a "1". This is a precondition for the SCL to resume regular operation after 3 transmission cycles with SDin and SDout values. The operator shall release the pressed acknowledgment button to enable further acknowledgments. See "Internal Items" in Table 38.
FAULTS_S	Any communication error (counter mismatch or CRC signature error) and/or timeouts cause the qualifier handler to set the qualifier bit for the Process Data of the connected FS-Device (see 11.11.4).

The lower part of the figure shows a combined input and output safety PDU specified in 11.4.3 and 11.4.5.

11.3.3 FS-Device SCL services

Figure 41 shows the FS-Device Safety Communication Layer services depicted by arrows in the upper part of the figure.

A service name carries either an extension "_DC" (Device Control) if it controls the FS-Device technology or an extension "_DS" (Device Status) if it is reporting its status.

Some of the service names correspond to the signal names of the Control Byte or Status Byte (see lower part of the figure and 11.4.5). That means they are correlated, but there is some control logic of the SCL in between. This control logic is time discrete and not continuous even if it is depicted as logic OR ("≥") box. Definitive are the state charts and the state transition tables of the SCL (see 11.5.3).

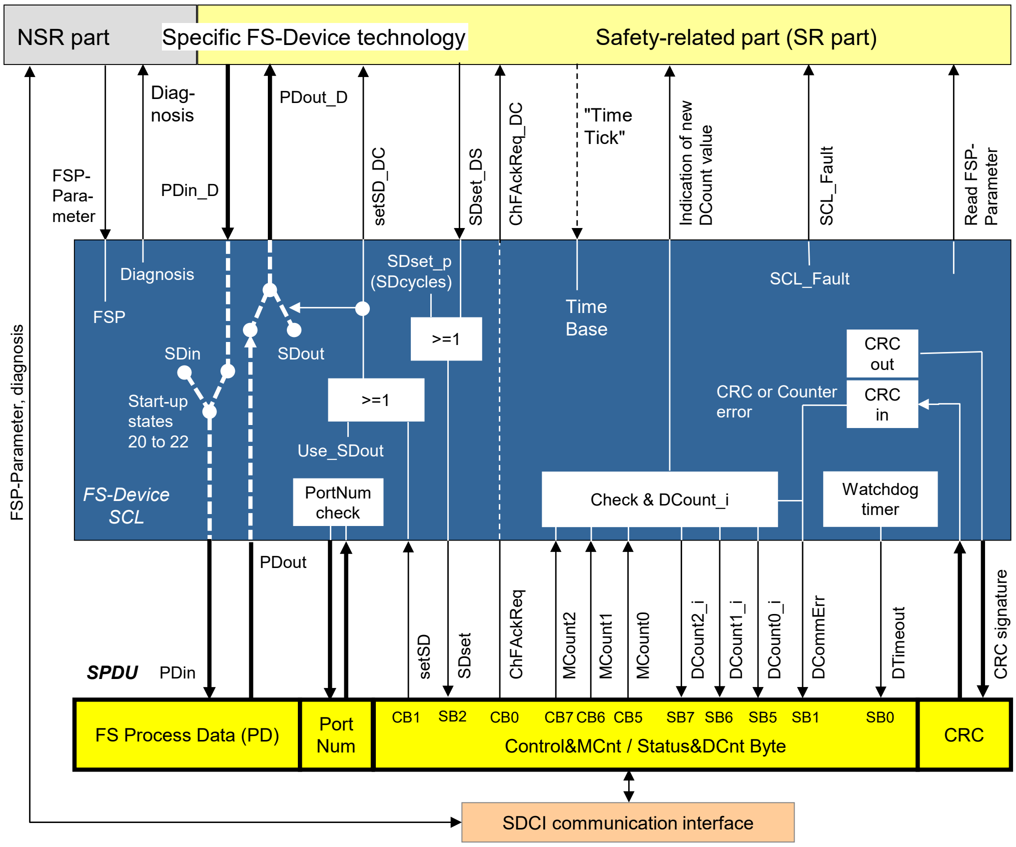


Figure 41 – FS-Device Safety Communication Layer services

The following services in Table 29 shall be available to the safety-related part of the FS-Device technology. Some services are non-safety-related and shall be available to the non-safety-related part of the FS-Device.

Table 29 – SCL services of FS-Device

Service/signal	Definition
PDin_D, PDout_D	These services carry the actual Process Data values. Real process values from the FS-Device and SDout (all bits "0") in case of safe state or the real process values to the FS-Device.
SDin, SDout	These services carry Process Data values all zero. Signal Use_SD indicates the usage of Process Data all zero.
setSD_DC	In case of emergency, safety control programs usually set output Process Data (PDout) for an actuator to "0". However, in some cases, for example burner ventilators, shut down may not be a safe state. This service, if set to "1", is additional information allowing an FS-Device to establish a safe state no matter what the values of Process Data are. Independent from PDout, this service causes the SCL to send SDout values to the FS-Device.
SDset_DS	This service, if set to "1", indicates that the FS-Device either reacts on a setSD_DC = "1" when the safe state is established or has been forced to establish safe state due to error or failure and delivers input Process Data values "0" (PDin_D).
ChFAckReq_DC	This service, if set to "1", indicates a pending operator acknowledgment. This signal is not safety-related and can be used to control an indicator, for example LED (light emitting diode).
Time tick	The SCL can be designed totally hardware independent if a periodic service call controls a time base inside the SCL.
Indication of new DCount value	Short demands of FS-Devices may not trip a safety function due to its chain of independent communication cycles across the network. Therefore, a demand shall last for at least two SCL cycles. This service provides the necessary information to implement the demand extension if required.

Service/signal	Definition
SCL_Fault	This service provides faults (errors) of the SCL software.
Read_FSP_Parameter	This service allows the FS-Device technology for reading the current FSP (protocol) parameter
Non-safety-related services:	
FSP_Parameter	The FS-Master transmits the FSP parameter record (block) at each start-up during PREOPERATE to the FS-Device. These parameters are propagated to the SCL using this service.
Diagnosis	SCL diagnosis information can be propagated to the SDCI Event system using this service.

The lower part of Figure 41 shows a combined input and output safety PDU specified in 11.4.3 and 11.4.5.

11.4 SCL protocol

11.4.1 Protocol phases to consider

Figure 42 shows the principle protocol phases to consider for the design according IEC 61784-3:2021.

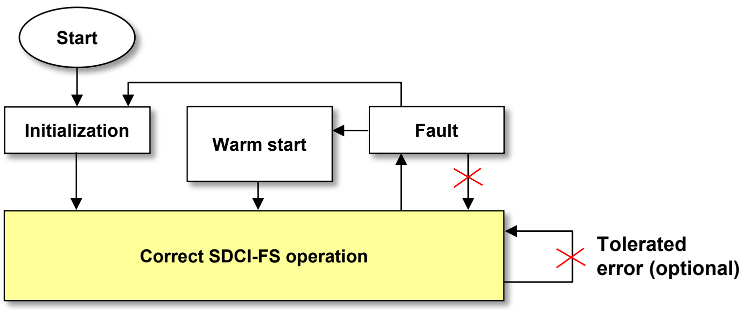


Figure 42 – Protocol phases to consider

The principle protocol phases and the corresponding requirements are listed in Table 30.

Table 30 – Protocol phases to consider

Phase	Activities	Requirements
Initialization	Establish communication, transfer FSP parameter to FS-Device, SD cycles	- Actuator shall be de-energized - SDout values shall be used during the first 3 SCL communication cycles
Setup or change	Commissioning, FST parameter backup	- As long as the FSP_TechParCRC is set to "0", cyclic data exchange of PD values is enabled.
Operation	Process Data exchange, power-down of FS-Device	- It is the responsibility of the FS-Device technology to detect undervoltages and to set SD values.
Restart after transition from fault	Timeout, operator acknowledgment	- Operator acknowledgment is required prior to a restart - MCounter reset (resynchronization) - SDout values shall be used during the first 3 SCL communication cycles
Warm start after transition from fault	CRC or counter error, operator acknowledgment	- Operator acknowledgment is required prior to a restart - SCL communication is not reset - SDout values shall be used during the first 3 SCL communication cycles
Shutdown	Contact bouncing, EMC voltage dips/changes	- It is the responsibility of the FS-Device technology to detect undervoltages and to set SD values.

11.4.2 FS-Device faults

The SCL protocol copes with faults occurring during transmission of safety PDUs such as CRC errors or timeouts. It is the responsibility of the designer of the FS-Device to cope with FS-Device faults and to make sure that the necessary functional safety actions will take place, for example setting of safety Process Data and the SDset_DS service.

11.4.3 Safety PDU (SPDU)

Figure 43 shows the structure of SPDUs of the FS-Master and FS-Device together with standard input and output data. The design follows the concept of explicit transmission of the safety measures for timeliness and authenticity according to IEC 61784-3:2021 in contrast to the implicit transmission via inclusion in the overall CRC signature calculation.

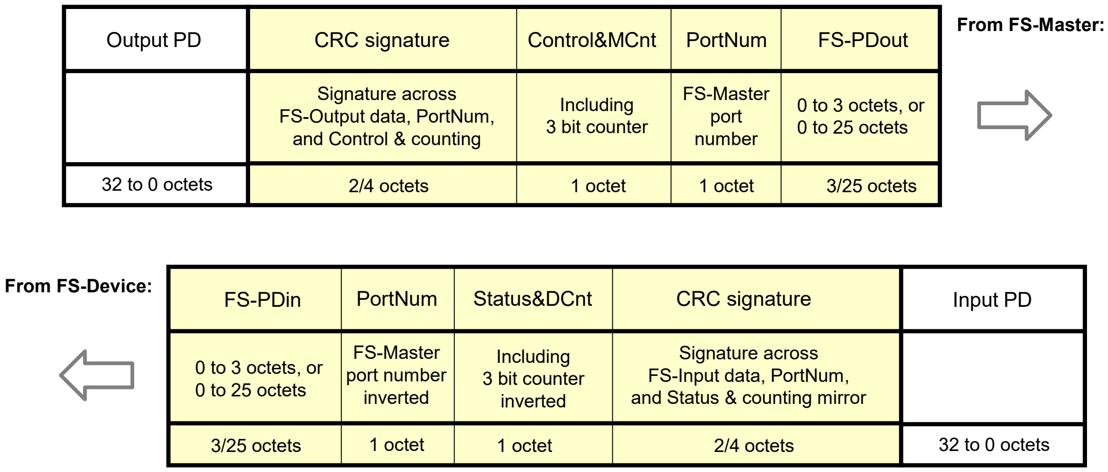


Figure 43 – Safety PDUs of FS-Master and FS-Device

The timeliness measure is represented by a 3-bit counter within the protocol management octets (see 11.4.6).

With respect to authenticity, only the FS-Master Port number is included in cyclic checking due to requested usage of unchanged SDCI implementations ("Black channel"). However, complete authenticity checking is performed during commissioning and at start-up.

The design is an enhancement of the original "de-energize to trip" principle. In case of a timeout, or a CRC error, or a counter error, or a PortNum error, the associated qualifier bit will be set. It will be only released after an explicit operator acknowledgment on the FS-Master side. After a CRC error a warm start is possible.

11.4.4 FS-Input and FS-Output data

The maximum possible size of the FS-Input and FS-Output data reaches from 0 to 25 octets depending on the amount of required standard SDCI data. See 11.4.7 for optimization issues and trade-offs. The possible data types are listed in Table 34.

NOTE Currently the safety code consists of only 4 or 6 octets and theoretically 26 octets could be available.

11.4.5 Port number

One octet carries the FS-Master Port number or value of FSP_Port respectively (see A.2.2). FS-Device returns the inverted value of the Port number.

11.4.6 Status and control

One octet is used in both transmission directions for the protocol flow of SDCI-FS. Table 31 shows the signals to control the protocol layer of an FS-Device and a counter value for the timeliness check together with a local watchdog timer adjusted through the "FSP_Watchdog" parameter (see A.2.6).

Table 31 – Control and counting (Control&MCnt)

CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0
Sequence counter, bit 2	Sequence counter, bit 1	Sequence counter, bit 0	Reserved ("0")	Reserved ("0")	Reserved ("0")	Activate safe state	Channel fault acknowledge request (indication)
MCount2	MCount1	MCount0	–	–	–	SetSD	ChFAckReq

Table 32 shows the feedback of the protocol layer of an FS-Device and the inverted counter value for the timeliness check. The counter values are inverted to prevent from undetected loop-back errors.

Table 32 – Status and counting mirror (Status&DCnt)

SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
Sequence counter, bit 2; inverted	Sequence counter, bit 1; inverted	Sequence counter, bit 0; inverted	Reserved ("0")	Reserved ("0")	Safe state activated	Communication error: CRC or counter /Port incorrect	Communi-cation fault: Timeout
DCount2_i	DCount1_i	DCount0_i	–	–	SDset	DCommErr	DTimeout

Table 33 shows the values of MCount and DCount_i during protocol operation.

Table 33 – MCount and DCount_i values

Phase	MCount		DCount_i	
	Dec	Bin	Dec	Bin
Initial or after timeout	0	000	7	111
Cyclic	1	001	6	110
	2	010	5	101
	3	011	4	100
	4	100	3	011
	5	101	2	010
	6	110	1	001
	7	111	0	000

11.4.7 CRC signature

For the design of the CRC mechanism and the calculation of the residual error probability/rate several parameters and assumptions are required:

- No multi-drop, multi-channel, or encrypted transmission in SDCI. Thus, the Binary Symmetric Channel (BSC) model can be applied.
- Explicit transmission of safety measures as opposed to implicit transmission. In this case, formulas are available within IEC 61784-3:2021.
- The sampling rate of safety PDUs is assumed to be a maximum of 1000 sampled safety PDUs per second.
- The monitoring times for errors in safety PDUs are listed in Table 41. Any detected CRC error within the safety communication layer shall trip the corresponding safety function (safe state). During the monitoring time only one nuisance trip is permitted. Maintenance is required.
- The generator polynomials in use shall be proven to be proper within the SPDU range.

- The seed value to be used for the CRC signature calculation is "1" (see D.3.6).
- In case the result of the CRC signature calculation leads to a "0", a "1" shall be sent and evaluated at the receiver side correspondingly.
- The assumed bit error probability for calculations is 10^{-2}.

Figure 44 shows the so-called 1 % share rule of the IEC 61784-3:2021. For SDCI-FS it means, the residual error rate of an SDCI-FS logical connection shall not exceed 1 % of the average frequency of a dangerous failure (PFH) of that safety function with the highest SIL the safety communication is designed for, which is SIL3. This value is 10^{-9}/h.

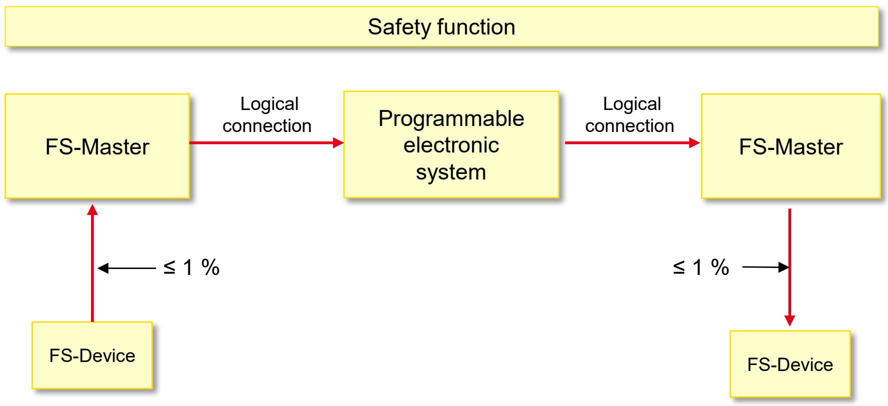


Figure 44 – The 1 % share rule of IEC 61784-3:2021

Calculations under the above conditions have shown the following possibilities (see Annex D):

– For a CRC-16 proper polynomial (*0x4EAB*) 3 octets of Process Data (safety PDU length = 7 octets),

– For a CRC-32 proper polynomial (*0xF4ACFB13*) 25 octets of Process Data (safety PDU length = 31 octets).

Thus, support of two variants is provided: CRC-16 with up to 3 octets of safety I/O data and CRC-32 with up to 25 octets of safety I/O data.

11.4.8 TADI safety considerations (informative)

In order for the SCL protocol to be compliant with IEC 61784-3:2021 (see 11.1), considerations and calculations shall be performed to prove that the overall PFH for

– timeliness,

– authenticity (masquerade, loopback), and

– data integrity,

is less than 10^{-9}/h (Figure 44).

The measures for timeliness are specified in 11.4.6 and consist of the 3-bit counter in combination with a watchdog timer within the SCL. This provides an added measure of data integrity and SCL state checking. However, in accordance with IEC 61784-3:2021, 5.8.8, due to the point-to-point nature of the SDCI-FS and no storage elements are allowed, the residual error rate for timeliness, RR_T is 0.

The measure for authenticity and masquerade checking is provided by the explicit transmission of a Port number in the SPDU that is checked by the receiving SCL endpoint. This provides an added measure of data integrity checking. However, in accordance with IEC 61784-3:2021, 5.8.7, due to the point-to-point nature of the SDCI-FS, the rate of occurrence for misdirected SPDUs (R_A) is 0, and therefore, the residual error rate for authenticity, RR_A is 0. Similarly, the rate of occurrence for masqueraded SPDUs (R_M) is 0, and therefore, the residual error rate for masquerade, RR_M is 0.

The measures for data integrity are provided by the CRC signature across safety process data, Port number, and Control&MCnt or Status&DCnt respectively as shown in 11.4.3. The calculation of the residual error probability can be performed at a bit error probability of 10^{-2} using the information in Annex D of this document. Designers shall observe the maximum sample rate specified in 11.4.7. The PFH monitor limits the maximum number of detected corrupted SPDUs for a given time interval (see 11.5.5).

11.4.9 Data types for SDCI-FS

11.4.9.1 General

The cyclically exchanged functional safety data structures between FS-Device and FS-Master comprise FS process I/O data and the SDCI-FS protocol trailer. They are transmitted in Safety PDUs.

Acyclically exchanged functional safety data structures are transmitted in SDCI On-request Data (OD) containers either from a Dedicated Tool or from a user program within an FS-PLC. In this case additional securing mechanisms (e.g. CRC signature) are required at each and every transfer or after a parameter block.

11.4.9.2 FS process I/O data (PDin and PDout)

For the FS process I/O data a well-defined set of data types and a corresponding description is defined for both FS-Device and FS-Master for correct processing and mapping to the upper-level FSCPs. Table 34 lists the three permitted data types (see Annex C).

Table 34 – FS process I/O data types

Data type	Coding	Length	See IEC 61131-9:2022	Device example
BooleanT/bit	BooleanT ("packed form" for efficiency, no WORD structures); assignment of signal names to bits is possible.	1 bit	F.2.2; Table F.22, and Figure F.9	Proximity switch
IntegerT(16)	IntegerT (enumerated or signed)	2 octets	F.2.4; Table F.4, Table F.7, and Figure F.3	Protection fields of laser scanner
IntegerT(32)	IntegerT (enumerated or signed)	4 octets	F.2.4; Table F.4, Table F.6, and Figure F.3	Encoder or length measurement (≈ ± 2 km, resolution 1 µm)

11.4.9.3 Qualifier

FS-Devices normally do not require qualifiers (see 11.11.2). The qualifier bits are configured together with the Process Data (or Safe Data = SD) during the mapping to the upper-level FSCP system. The data structures depend on the rules of these FSCP systems.

In case of FS-Terminals (see 11.11.3) the rules in Table 35 for the layout of binary and digital data and their qualifier bits apply.

Table 35 – Rules for the layout of values and qualifiers

No.	Rule
1	Only Boolean (DI, DO) and IntegerT(16) or IntegerT(32) (AI, AO) data types shall be used. Any value shall be assigned to one of these categories.
2	Boolean values precede Integer values
3	IntegerT(16) precedes IntegerT(32) values
4	Values precede qualifier in an octet-wise manner
5	Qualifiers follow directly input values. In case of no input values only the qualifiers for output values are placed.
6	Qualifier for input values precede qualifier for output values
7	Qualifiers for each category (DI, DO, AI, AO) are packed separately in an octet-wise manner
8	If data types are missing the remaining data types catch up

Table 36 shows the ranking of values and qualifiers.

Table 36 – Order of values and qualifier

Order	To FS-Master	To FS-Device
1	Value DI	Value DO
2	Value AI	Value AO
3	Qualifier DI	–
4	Qualifier AI	–
5	Qualifier DO	–
6	Qualifier AO	–

11.4.9.4 SDCI-FS protocol trailer

The data types for the protocol trailer ("safety code") are specified in Clause C.5.

11.4.9.5 FSP and FST parameter

No particular data type definitions are required.

11.5 SCL behavior

11.5.1 General

The state machines for the FS-Master and the FS-Device safety communication layer are designed using the chosen safety measures in Table 27 and the protocol signals in 11.4.5.

11.5.2 SCL state machine of the FS-Master

Figure 45 shows the FS-Master state machine for wired SDCI point-to point communication.

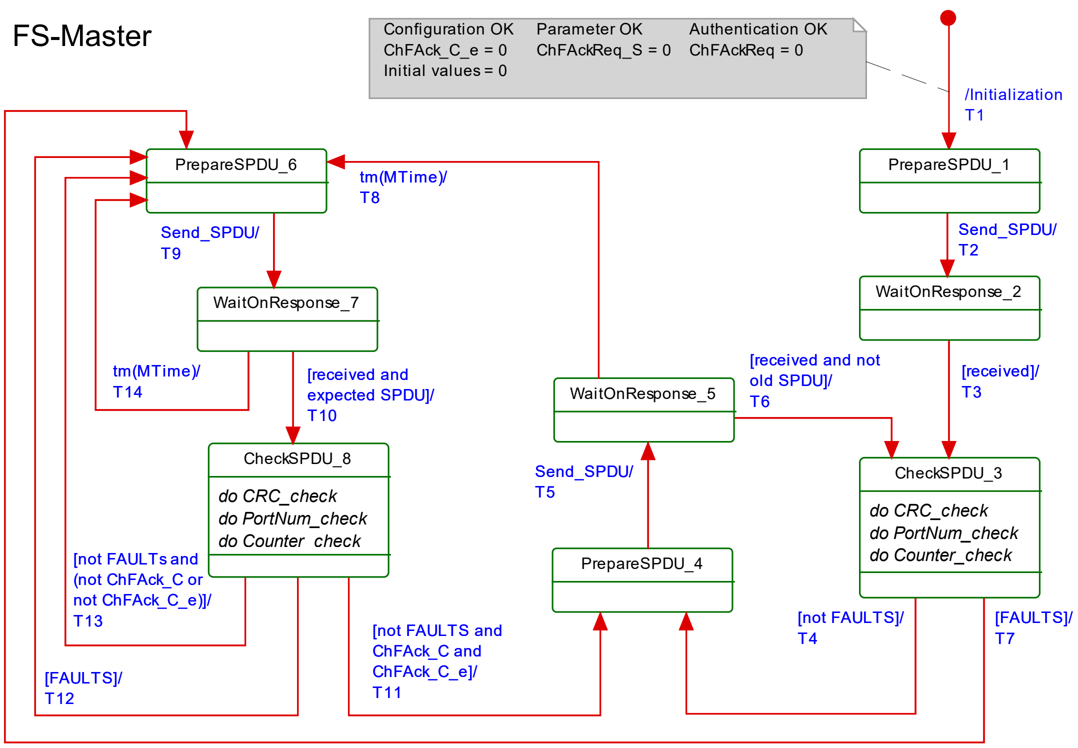


Figure 45 – SCL state machine of the FS-Master

The terms used in Figure 45 are defined in Table 37.

1892 **Table 37 – Definition of terms used in SCL state machine of the FS-Master**

Term	Definition
ChFAck_C	Operator acknowledgment for the safety function via the FS-Gateway
FAULTS	MTimeout: FS-Master timeout when waiting on an FS-Device SPDU response, or MCommErr: FS-Master detects a corrupted FS-Device SPDU response (incl. counter/Port error), or DTimeout: FS-Device reported a timeout of its SCL via Status&DCnt Byte, or DCommErr: FS-Device reported a CRC (incl. counter/Port error) by its SCL via Status&DCnt Byte

1893 **Table 38 – FS-Master SCL states and transitions**

STATE NAME	STATE DESCRIPTION
Initialization	Initial state of the FS-Master SCL instance upon power-on (one per Port).
1 PrepareSPDU	Preparation of a (*regular)* SPDU for the FS-Device. Send SPDU when prepared.
2 WaitOnResponse	SCL is waiting on SPDU from FS-Device. SPDU with all octets "0" shall be ignored.
3 CheckSPDU	Check received SPDU for not FAULTS (→ T4). In case of FAULTS: errors within the Status&DCnt Byte (DCommErr, DTimeout, SDset) → T7
4 PrepareSPDU	Preparation of a (*regular)* SPDU for the FS-Device. Send SPDU when prepared.
5 WaitOnResponse	SCL is waiting on next SPDU from FS-Device not carrying the previous DCount_i. SPDU with all octets "0" shall be ignored.
6 PrepareSPDU	Preparation of an (*exceptional*) SPDU for the FS-Device (due to MTimeout, missing OpAck, or FAULTS).
7 WaitOnResponse	SCL is waiting on next SPDU from FS-Device not carrying the previous DCount_i. When received → T10, after MTimeout → T14.
8 CheckSPDU	Check received SPDU for a CRC error (MCommErr) and for potential FS-Device faults within the Status&DCnt Byte (DTimeout, DCommErr). Once a fault occurred, no automatic restart of a safety function is permitted unless an operator acknowledgement signal (ChFAck_C) arrived (see Figure 40). Hint: A delay time may be required avoiding the impact of an occasional system shutdown.

1894

TRAN-SITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	use SD, setSD =1, SDset_S =1 MCount = 0
T2	1	2	–
T3	2	3	–
T4	3	4	MCount = MCount + 1 **if** MCount = 8 **then** MCount = 1 **if** SDset =1 or setSD_C =1 **then** use SDin, SDset_S =1 **else** use PDin, SDset_S =0 **if** setSD_C =1 **then** use SDout, setSD =1 **else** use PDout, setSD =0
T5	4	5	restart MTimer
T6	5	3	–
T7	3	6	use SD, setSD =1, SDset_S =1 MCount = MCount + 1 **if** MCount = 8 **then** MCount = 1
T8	5	6	use SD, setSD =1, SDset_S =1 MCount = 0
T9	6	7	restart MTimer
T10	7	8	–
T11	8	4	ChFAckReq =0, ChFAckReq_S =0, ChFAck_C_e =0, MCount = MCount + 1

TRAN-SITION	SOURCE STATE	TARGET STATE	ACTION
			if MCount = 8 **then** MCount = 1 **if** SDset =1 or setSD_C =1 **then** use SDin, SDset_S =1 **else** use PDin, SDset_S =0 **if** setSD_C =1 **then** use SDout, setSD =1 **else** use PDout, setSD =0
T12	8	6	ChFAckReq =0, ChFAckReq_S =0, ChFAck_C_e =0, use SD, setSD =1, SDset_S =1 MCount = MCount + 1 **if** MCount = 8 **then** MCount = 1
T13	8	6	ChFAckReq =1, ChFAckReq_S =1, /*set qualifier/acknowledgment request*/ **if** ChFAck_C = 0 **then** ChFAck_C_e =1 use SD, setSD =1, SDset_S =1 MCount = MCount + 1 **if** MCount = 8 **then** MCount = 1
T14	7	6	ChFAckReq =0, ChFAckReq_S =0, ChFAck_C_e =0, use SD, setSD =1, SDset_S =1 MCount = 0

INTERNAL ITEMS	TYPE	DEFINITION
MTimer	Timer	This timer checks the arrival of the next valid SPDU from the FS-Device in time. The FS-Master tool is responsible to define this watchdog time. Value range is 1 to 65 535 ms.
ChFAck_C_e	Flag	By means of this auxiliary variable (bit) it is ensured that the safe state will be left only after a signal change of ChFAck_C from 0 → 1 (edge). Without this mechanism an operator could overrule safe states by permanently actuating the ChFAck_C signal.
FAULTS	Flags	Permanent storage of the following errors or failures can be omitted within the FS-Master, if it can be assumed that the upper-level FSCP system prevents from automatic restart of safety functions (no FS-Device persistence): - MCommErr or MTimeout - DCommErr, including counter/Port error (Status&DCnt Bit 1 and PortNum) - DTimeout (Status&DCnt Bit 0)
Expected SPDU	Guard	Mirrored inverted counter (DCount_i = inverted MCount)
Not old SPDU	Guard	Counter value ≠ value of previous SPDU
do CRC_check	Activity	SCL calculates CRC signature across received SPDU while signature value = "0" and compares with received CRC signature
do PortNum_check	Activity	SCL checks whether PortNum carries the correct FS-Master Port number
do Counter_check	Activity	SCL checks whether DCount carries an expected value (mirror)
NOTE Variables within ACTIONs are defined in 11.3		

11.5.3 SCL state machine of the FS-Device

Figure 46 shows the corresponding FS-Device state machine.

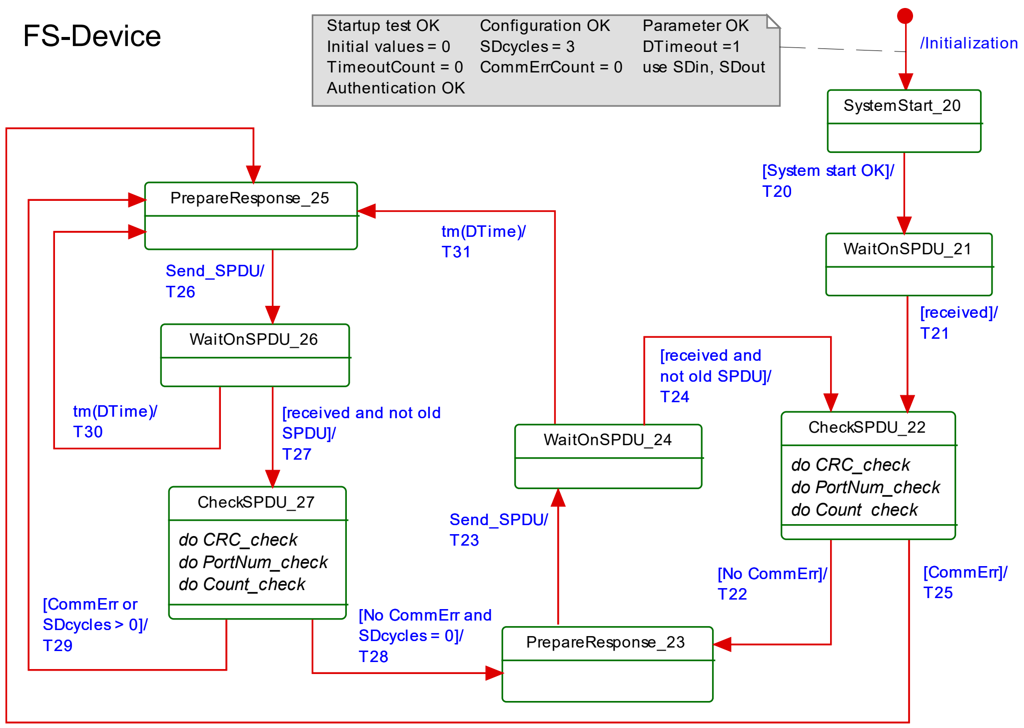


Figure 46 – SCL state machine of the FS-Device

The terms used in Figure 46 are defined in Table 39.

Table 39 – Definition of terms used in SCL state machine of the FS-Device

Term	Definition
CommErr	The SCL of the FS-Device detected a CRC or counter/Port error in the received SPDU
CommErrCount	See INTERNAL ITEM in Table 40
SDcycles	See INTERNAL ITEM in Table 40
DTimeout	FSP_WatchdogTime expired
TimeoutCount	See INTERNAL ITEM in Table 40

Table 40 – FS-Device SCL states and transitions

STATE NAME	STATE DESCRIPTION
Initialization	Initialization of the FS-Device upon power-on. Upon power-on, the FS-Device (actuator) sets the PDout to "0". Upon power-on the FS-Device (sensor) is sending "0".
20 SystemStart	Immediately after FSP parameterization the FS-Device sets PDout to SDout values. Immediately after FSP parameterization it is sending Process Data (PD).
21 WaitOnSPDU	SCL is waiting on next SPDU from FS-Master. SPDU with all octets "0" shall be ignored.
22 CheckSPDU	Check received SPDU from FS-Master for CRC errors; set ChFAckReq_DC = ChFAckReq. When guard "No CommErr" = true → T22. When guard "CommErr " = true → T25
23 PrepareResponse	Preparation of (*regular)* SPDU response for the FS-Master (response message)
24 WaitOnSPDU	SCL is waiting on next (*regular*) SPDU from FS-Master not carrying the previous MCount. After FSP_WatchdogTime expired → T31. When SPDU received and guard "MCounter_incremented" = true → T24 (*regular* cycle)
25 PrepareResponse	Preparation of (*exceptional*) SPDU response for the FS-Master (due to DTimeout or DCommErr = error report bits in Status&DCnt Byte)
26 WaitOnSPDU	SCL is waiting on next SPDU from FS-Master not carrying the previous MCount. SPDU with all octets "0" shall be ignored. After FSP_WatchdogTime expired → T30. When SPDU received and guard "MCounter_incremented" = true → T27

STATE NAME	STATE DESCRIPTION
27 CheckSPDU	Check received SPDU from FS-Master for CRC errors; set ChFAckReq_DC = ChFAckReq. When guard "No CommErr and SDcycles =0" = true → T28. When guard "CommErr or SDcycles >0" = true → T29

TRAN-SITION	SOURCE STATE	TARGET STATE	ACTION
T20	20	21	–
T21	21	22	–
T22	22	23	use PDin_D, DCommErr = 0, /*Status&DCnt, Bit 1*/ DTimeout = 0, /*Status&DCnt, Bit 0*/ DCount_i = MCount_inv, restart DTimer **if** SDcycles <> 0 **then** use SDout, setSD_DC=1, SDset =1, /*during SDcycles: SDset_p =1*/ SDcycles = SDcycles - 1 **else** use PDout, setSD_DC=0, SDset = 0 **if** setSD =1 /*use_SD =1*/ **then** use SDout, setSD_DC=1,
T23	23	24	**if** SDset_DS = 1 /* FS-Device fault*/ **then** SDset = 1
T24	24	22	–
T25	22	25	use PDin_D, use SDout, setSD_DC=1, SDset = 1, DCommErr =1, /*Status&DCnt, Bit 1*/ CommErrCount = 1, DCount_i = MCount_inv, SDcycles = 3, restart DTimer
T26	25	26	–
T27	26	27	–
T28	27	23	use PDin_D, use SDout, setSD_DC=0, SDset = 0, DCount_i = MCount_inv, DCommErr =0, /*Status&DCnt, Bit 1*/ DTimeout =0, /*Status&DCnt, Bit 0*/ restart DTimer,
T29	27	25	use PDin_D, use SDout, setSD_DC=1, SDset = 1, DCount_i = MCount_inv, restart DTimer **if** CommErr **then** DCommErr = 1, /*Status&DCnt, Bit 1*/ CommErrCount = 1, SDcycles = 3, **else** SDcycles = SDcycles -1 **if** CommErrCount = 1 **then** DCommErr = 1, /*Status&DCnt, Bit 1*/ CommErrCount = 0 **else** DCommErr = 0 /*Status&DCnt, Bit 1*/ **if** TimeoutCount = 1 **then** DTimeout = 1, /*Status&DCnt, Bit 0*/ TimeoutCount = 0 **else** DTimeout = 0 /*Status&DCnt, Bit 0*/
T30	26	25	use PDin_D, use SDout, setSD_DC=1, SDset =1, DTimeout =1, /*Status&DCnt, Bit 0*/ TimeoutCount =1, SDcycles = 3,

TRAN-SITION	SOURCE STATE	TARGET STATE	ACTION
			restart DTimer, DCount_i = MCount_inv
T31	24	25	use PDin_D, use SDout, setSD_DC=1, SDset =1, DTimeout =1, /*Status&DCnt, Bit 0*/ TimeoutCount =1, SDcycles = 3, restart DTimer, DCount_i = MCount_inv

INTERNAL ITEM	TYPE	DEFINITION
MCount_inv	Variable	Inverse value of current MCount value
SDcycles	Counter	This decremental counter is used to cause the SCL setting SDout and SDset for at least 3 cycles during start-up and after a fault. Value range is 3 to 0.
CommErrCount	Counter	This decremental counter is used to guarantee the bit "DCommErr" within the Status&DCnt Byte is being set at least for 1 cycle or for a maximum of 2 cycles. Value range is 1 to 0.
TimeoutCount	Counter	This decremental counter is used to guarantee the bit "DTimeout" within the Status&DCnt Byte is being set at least for 1 cycle or for a maximum of 2 cycles. Value range is 1 to 0.
do CRC_check	Activity	SCL calculates CRC signature across received SPDU while signature value = "0" and compares with received CRC signature
do PortNum_check	Activity	SCL checks whether PortNum carries the correct FS-Master Port number
do Counter_check	Activity	SCL checks whether MCount carries "0" (first SPDU) or either "0" or the expected subsequent value (all other SPDUs)
NOTE Variables within ACTIONs are defined in 11.3		

It is very unlikely for an FS-Device to receive SPDUs with all octets "0". The SCL within the FS-Device shall ignore such an SPDU. Normally, at least the CRC signature will be "1" if Process Data and Control Byte are "0" according to the rules in 11.4.7.

11.5.4 Sequence charts for several use cases

11.5.4.1 FS-Master and FS-Device both with power ON

Figure 47 shows the sequence chart of a regular start-up of both FS-Master and FS-Device.

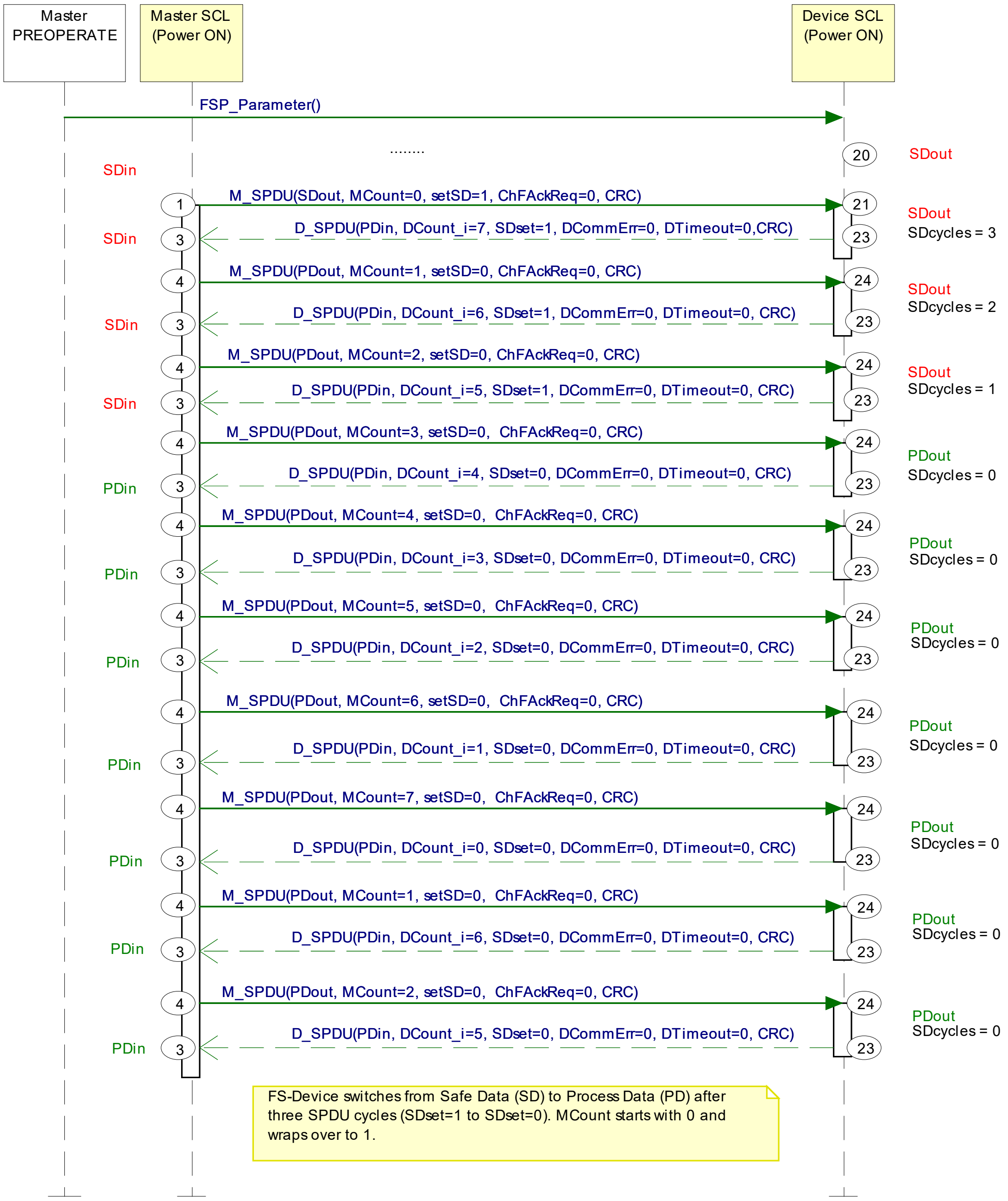


Figure 47 – FS-Master and FS-Device both with power ON

Upon power-on both FS-Master and FS-Device are providing SDin (PDin = "0") and SDout (PDout = "0") respectively. Both keep these values for 3 communication cycles (SDcycles) before switching to the regular mode, where only the MCounter and DCounter values are changing.

11.5.4.2 FS-Master with power OFF → ON

Figure 48 shows the sequence chart of regular operation while FS-Master has been switched OFF and ON again.

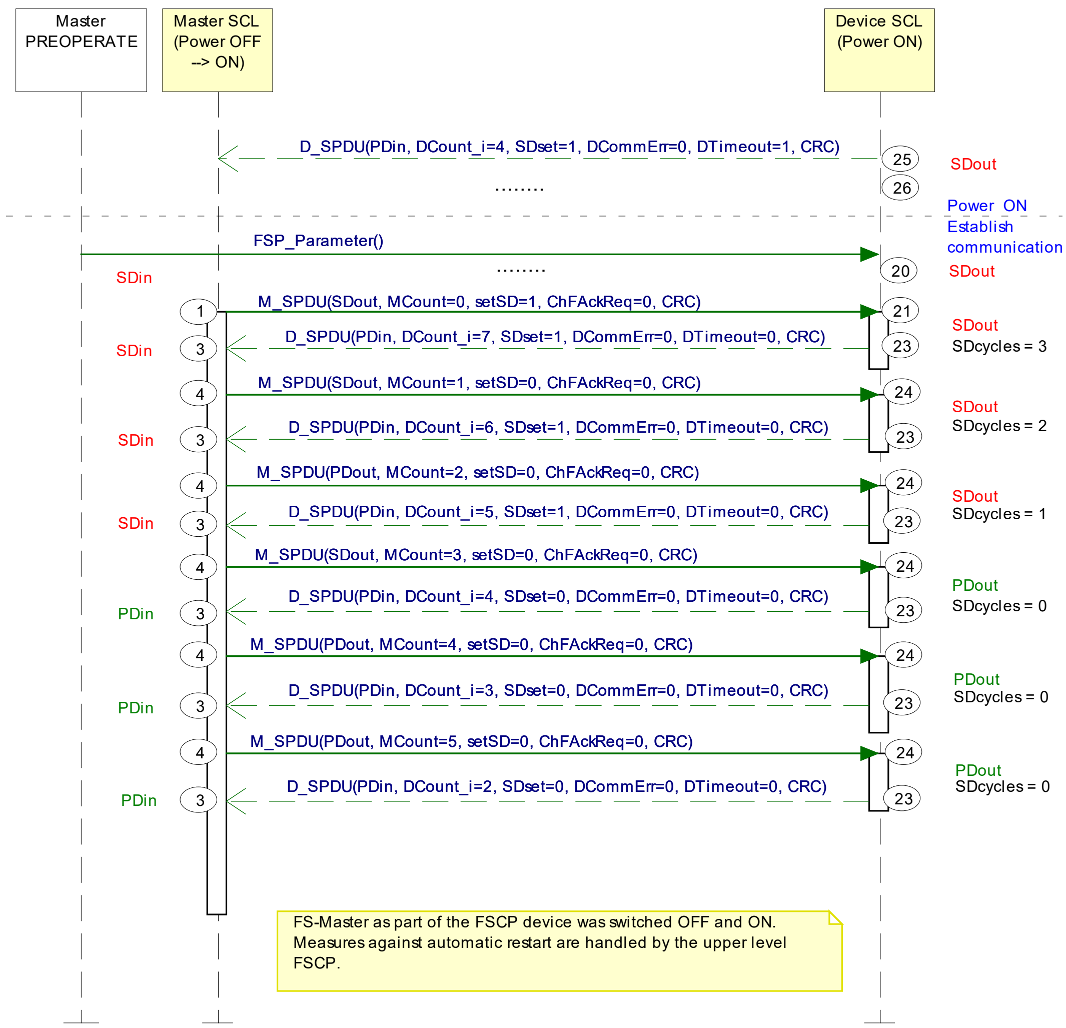


Figure 48 – FS-Master power OFF → ON

The FS-Device communication part is always powered by the FS-Master. Thus, if the FS-Master is switched OFF and ON, the FS-Device is just following, and a regular start-up occurs. Since the FS-Master is part of an upper-level FSCP system, this FSCP system is responsible to prevent from automatic restart of safety functions in this case.

11.5.4.3 FS-Device with delayed SCL start

Figure 49 shows the sequence chart when the SCL start within the FS-Device is delayed.

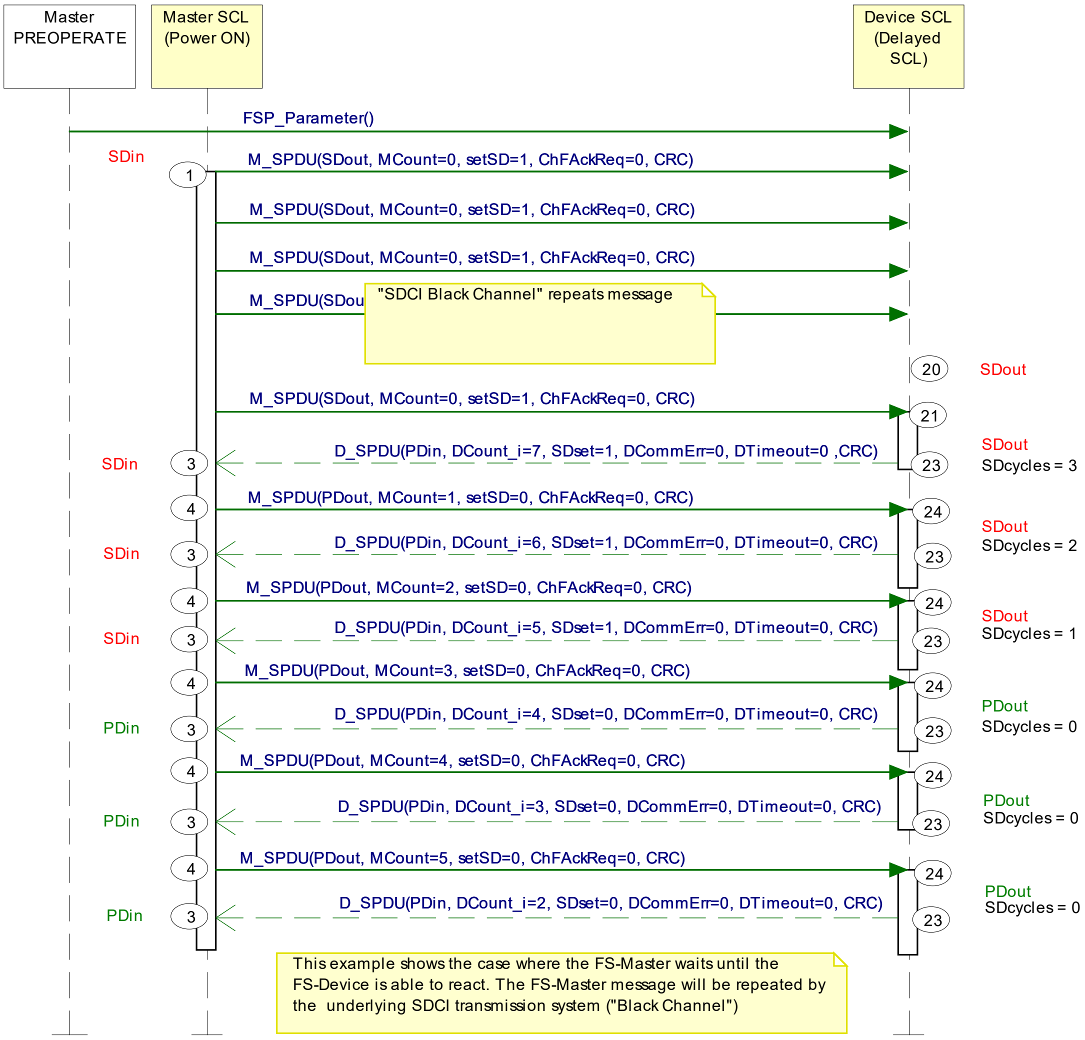


Figure 49 – FS-Device with delayed SCL start

This diagram shows how an FS-Master SCL waits on the SCL of the FS-Device in case of delays. The initial SPDU of the FS-Master is repeated by the SDCI transmission system (black channel) until the SCL of the FS-Device is ready to process in state 21.

As long as the SCL of the FS-Device is not ready, the response SPDU contains all "0" and the FS-Master SCL will ignore such an SPDU. PDvalid/invalid of SDCI is reserved for the non-safety part of the entire message.

11.5.4.4 FS-Device with power OFF and ON

Figure 50 shows the sequence chart when the FS-Device switches power OFF and ON again.

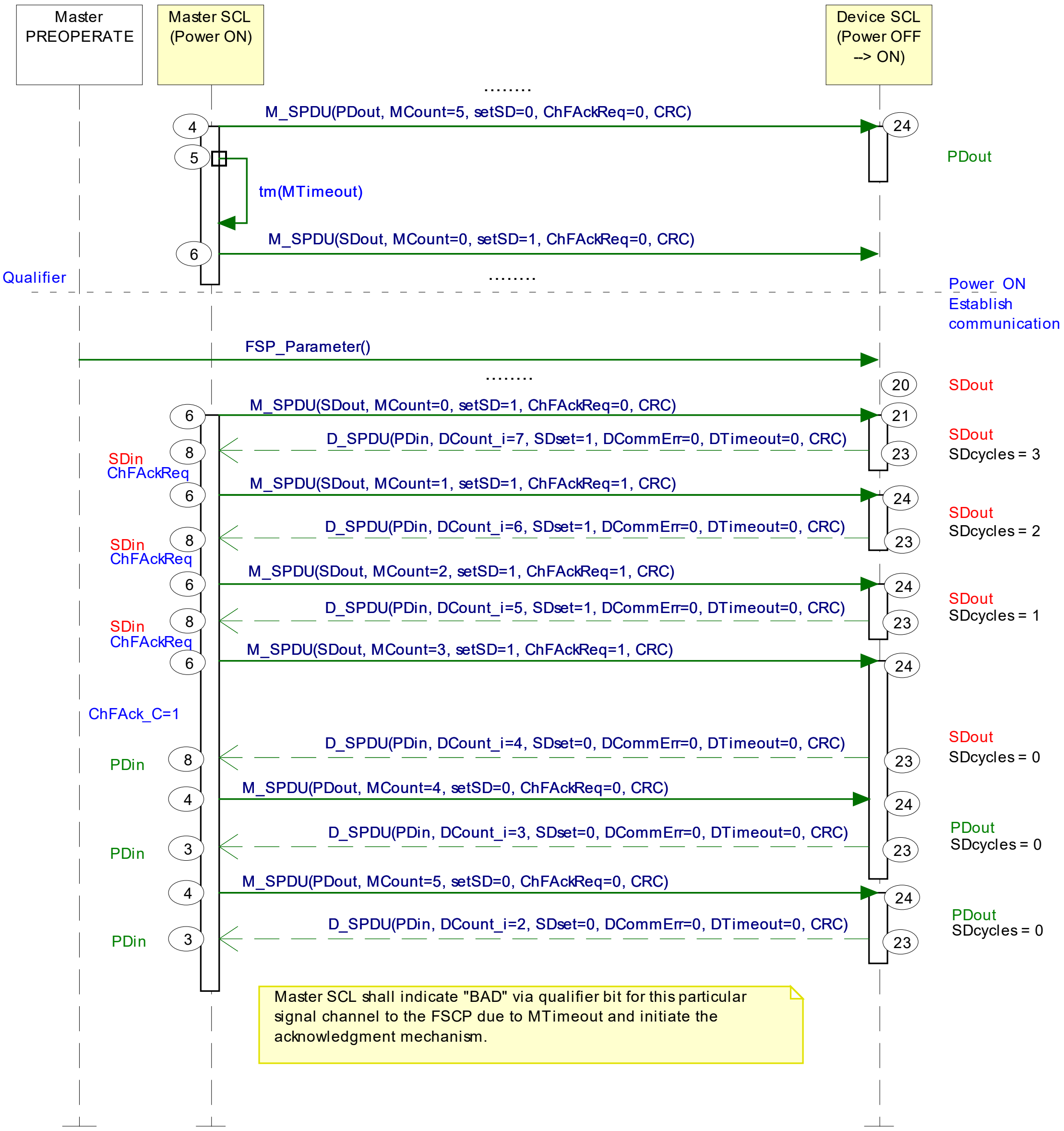


Figure 50 – FS-Device with power OFF and ON

This case assumes for example a short unplug and plug of the FS-Device causing a FAULT (MTimeout) on the FS-Master side. This FAULT causes a Qualifier bit to be set that requires via ChFAckReq (=1) an acknowledgment via ChFAck_C (=1). FS-Master and FS-Device keep SDin and SDout until this acknowledgment arrived.

11.5.4.6 FS-Master detects CRC signature error

Figure 51 shows the sequence chart when the FS-Master detects a CRC signature error.

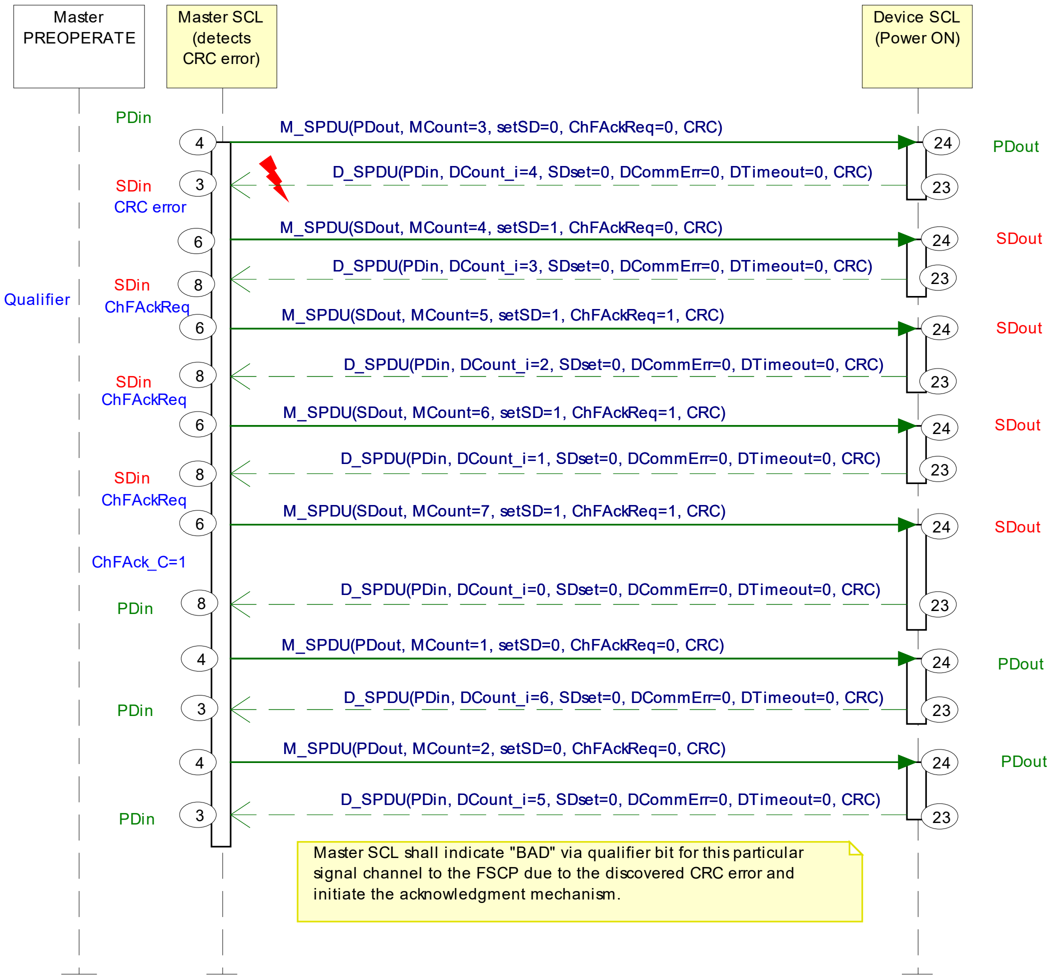


Figure 51 – FS-Master detects CRC signature error

FS-Master received an SPDU with falsified data or falsified CRC signature which results in a "CRC error" (MCommErr). Both FS-Master and FS-Device switch to SDin and SDout respectively and the FS-Master/Gateway creates a qualifier bit and indicates a ChFAckReq signal. This signal is indicated also to the FS-Device via ChFAckReq (=1) for indication via LED (light emitting diode) to the user.

FS-Master and FS-Device keep SDin and SDout until the acknowledgment ChFAck_C (=1) arrived.

11.5.4.7 FS-Device detects CRC signature error

Figure 52 shows the sequence chart when the FS-Device detects a CRC signature error.

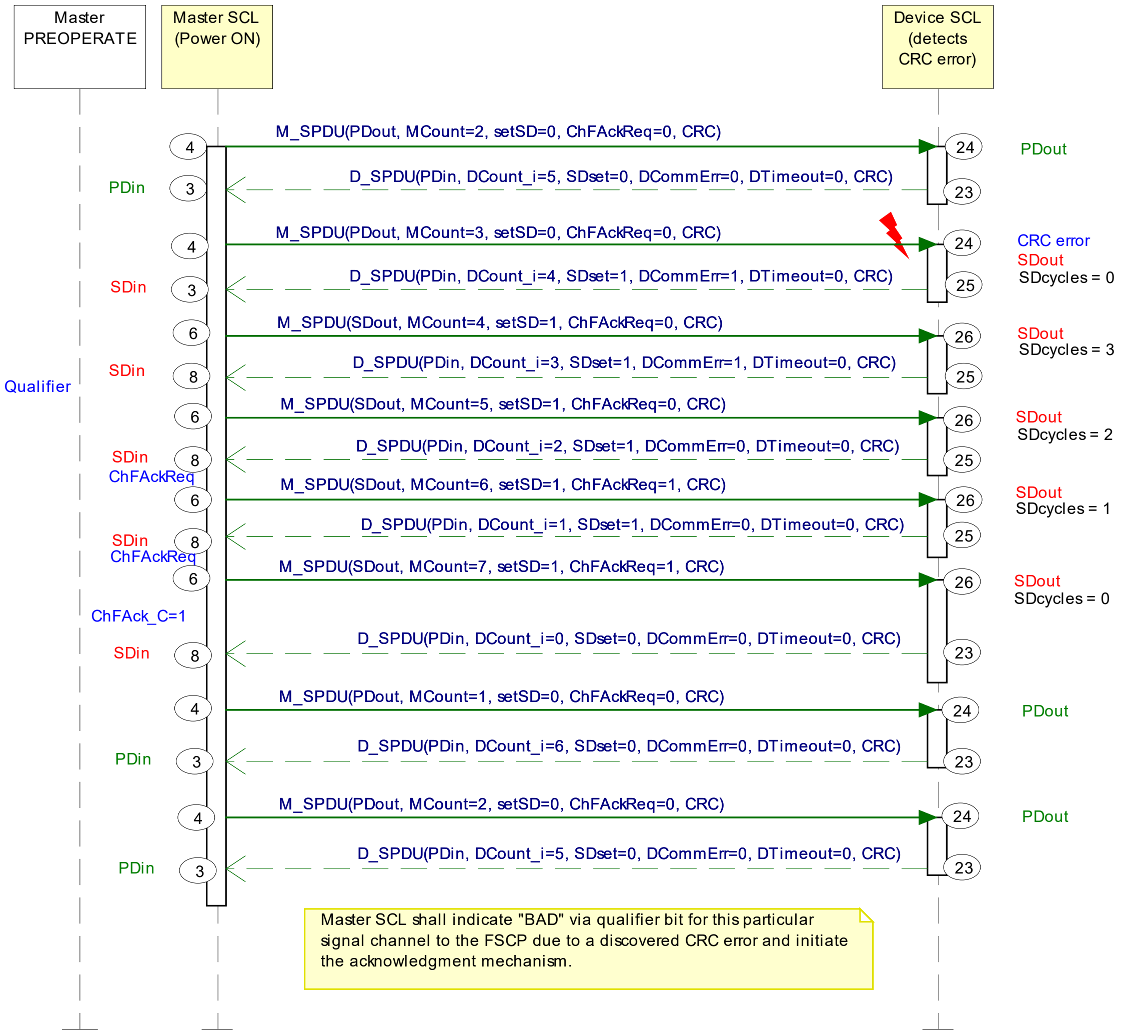


Figure 52 – FS-Device detects CRC signature error

FS-Device received an SPDU with falsified data or falsified CRC signature which results in a "CRC error" (DCommErr). Both FS-Master and FS-Device switch to SDin and SDout respectively caused by FS-Device Status Byte information (SDset=1 and DCommErr=1). The FS-Master/Gateway creates a qualifier bit and indicates a ChFAckReq signal. This signal is indicated also to the FS-Device via ChFAckReq (=1) for indication via LED (light emitting diode) to the user.

The FS-Device runs through 3 SDcycles and afterwards FS-Master and FS-Device keep SDin and SDout until the acknowledgment ChFAck_C (=1) arrived.

11.5.5 Monitoring of safety times

Figure 53 illustrates SDCI times and safety times.

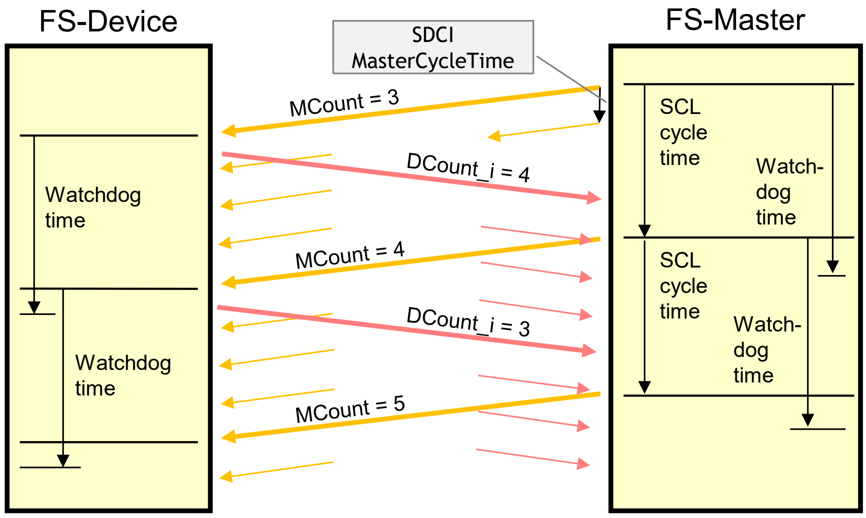


Figure 53 – Monitoring of the SCL cycle time

The base SDCI system ("black channel") transmits SPDUs within the MasterCycleTime (see :) from the FS-Master to the FS-Device and back. The same SPDU, for example with MCount = 3, may be sent several times before the Safety Communication Layer (SCL) starts the next SCL cycle with MCount = 4. In the meantime, the FS-Master received the response SPDU from the FS-Device with DCount_i = 4.

Table 41 shows timing constraints.

Table 41 – Timing constraints

<table>
<tr><th>Item</th><th colspan="2">Constraints</th></tr>
<tr><td>Synchronization</td><td colspan="2">At each start-up and after an FS-Master timeout, the FS-Master SCL uses MCount = 0</td></tr>
<tr><td>SCL cycle time</td><td colspan="2">The SCL cycle time comprises the transmission time of the FS-Master SPDU, the FS-Device processing time, the transmission time of the FS-Device response SPDU, and the FS-Master processing time until the next FS-Master SPDU (see Figure 53)</td></tr>
<tr><td>Watchdog time</td><td colspan="2">The entire SCL cycle time is monitored by the watchdog timer, whose time value is defined by the parameter FSP_Watchdog (see A.2.6).</td></tr>
<tr><td>Counter check</td><td colspan="2">The counter values are included in the cyclic CRC signature calculation. An incorrect CRC signature value will already lead immediately to a safe state. The immediate counter check in some states is used for discarding "outdated SPDUs".</td></tr>
<tr><td>Repetition</td><td colspan="2">Repetition in case of detected incorrect CRC signatures is not provided</td></tr>
<tr><td>PFH-Monitor</td><td colspan="2">The FS-Master holds the information about the reliability of both SPDU transmissions from the FS-Device and to the FS-Device (see Table 32, bit 1). Thus, the FS-Master monitors the average frequency of a dangerous failure within a given time frame (PFH-Monitor time). The FS-Master state machine is designed such that any corrupted SPDU leads always to a safe state. Whenever the unlikely event of a detected corrupted SPDU occurs during the shift of production or operation, the responsible operator is assigned to play the role of the PFH-Monitor and can tolerate the indication and acknowledge it. In case of frequent indications more often than once per PFH-Monitor time, a check of the installation or the transmission quality should be performed (see Clause H.6).</td></tr>
<tr><td rowspan="2">PFH-Monitor time (h)</td><td>10</td><td>FSP_ProtMode = 0x01; CRC-16, see A.2.5</td></tr>
<tr><td>10</td><td>FSP_ProtMode = 0x02; CRC-32, see A.2.5</td></tr>
</table>

11.5.6 Reaction in the event of a malfunction

11.5.6.1 General

Subclauses 11.5.6.2 to 11.5.6.10 specify possible communication errors. They are derived from 5.3 in IEC 61784-3:2021 and refer to Table 27 in this document. Additional notes are provided to indicate the typical behavior of the SDCI black channel.

11.5.6.2 Corruption

Messages may be corrupted due to errors within a communication participant, due to errors on the transmission medium, or due to message interference.

NOTE 1 Bit falsifications within messages during transfer is a normal phenomenon for any standard communication system. Such errors are detected at receivers with high probability by use of a hash function, in case of SDCI a checksum (CKT or CKS), and the message is ignored (see IEC 61131-9:2022, Clause A.1). After two retries the Master initiates a complete restart with wake-up.

NOTE 2 If the recovery or repetition procedures take longer than a specified deadline, a message is classed as "Unacceptable delay" (see 11.5.6.6).

Countermeasures:

The CRC signature as specified in 11.4.7 detects the bit errors in messages between FS-Master and FS-Device to the extent required for SIL3 applications. The CRC signature is generated across the SPDU including the PD or SD data, the Port number, and the Control&MCnt or Status&DCnt octet for cyclic communication.

At start-up, the FSP parameters are sent once to the FS-Device via ISDU services. They are secured by the 16-bit FSP_ProtParCRC signature. The frequency of its occurrence is assumed to be 1/day as parameter for the calculation of the residual error rate.

11.5.6.3 Unintended repetition

Due to an error, fault or interference, messages are repeated.

NOTE Repetition by the sender is a normal procedure when an expected acknowledgment/response is not received from a target station, or when a receiver station detects a missing message and asks for it to be resent.

Countermeasures:

The data within the black channel are transferred cyclically. Thus, an incorrect message/SPDU with the latest received counter value that is inserted once will be ignored. The thereby possible delay of a demand can be one DTime or MTime respectively.

11.5.6.4 Incorrect sequence

Due to an error, fault or interference, the predefined sequence (for example natural numbers, time references) associated with messages from a particular source is incorrect.

NOTE In SDCI only one sequence is active from one source, the message handler.

Countermeasures:

The receiver will detect any incorrect sequence due to the stringently sequential expectation of the MCount and DCount values.

11.5.6.5 Loss

Due to an error, fault or interference, a message or acknowledgment is not received.

Countermeasures:

Lost information will be detected by stringently changing and examining the MCount/DCount and/or MTime/DTime within the safety communication layer of the respective receiver.

11.5.6.6 Unacceptable delay

Messages may be delayed beyond their permitted arrival time window, for example due to bit falsifications in the transmission medium, congested transmission lines, interference, or due to communication participants sending messages in such a manner that services are delayed or denied (for example FIFOs in switches, bridges, routers).

NOTE 1 SDCI provides a point-to-point communication interface with defined message sequences and thus the probability for congestion and storage of messages is extremely low.

Countermeasures:

A consecutive counter in each message (MCount/DCount) together with a watchdog timer (MTime/DTime) will detect unacceptable delays.

11.5.6.7 Insertion

Due to a fault or interference, a message is received that relates to an unexpected or unknown source entity.

NOTE 1 These messages are additional to the expected message stream, and because they do not have expected sources, they cannot be classified as Correct, Unintended repetition, or Incorrect sequence.

NOTE 2 SDCI provides a point-to-point communication interface (Port) and thus the probability for insertion of messages is extremely low.

Countermeasures:

The receiver will detect any incorrect sequence due to the stringently sequential expectation of the MCount and DCount values.

11.5.6.8 Masquerade

Due to a fault or interference, a message is inserted that relates to an apparently valid source entity, so a misdirected non-safety related message may be received by a safety related participant, which then treats it as safety related correct message.

NOTE 1 Communication systems used for safety-related applications can use additional checks to detect Masquerade, such as authorised source identities and passphrases or cryptography.

NOTE 2 SDCI provides a point-to-point communication interface (Port) and thus the probability for insertion of messages is extremely low.

Countermeasures:

In case of NSR data instead of a regular SPDU, the CRC signature mechanism of the SCL will detect this incident.

After changes of wiring, the FS-Devices can detect misconnections through the FSP_Authenticity1/2 and FSP_Port parameters (see A.2.1 and A.2.2) at start-up.

11.5.6.9 Addressing

Due to a fault or interference, a safety related message is delivered to the incorrect safety related participant, which then treats reception as correct. This includes the so-called loopback error case, where the sender receives back its own sent message.

NOTE 1 The probability of not detecting a misdirected non-safety related message is lower than the probability of not detecting a misdirected safety related message since the SPDU structures are similar due to the shared protocol procedures.

NOTE 2 SDCI provides a point-to-point communication interface (Port) and thus the probability for insertion of messages is extremely low. However, FS-Master may use internal bus mechanisms to address Ports.

Countermeasures:

Port addressing errors can be detected by the Port number (PortNum) within the SPDU.

After changes of wiring, the FS-Devices can detect misconnections through the FSP_Authenticity1/2 and FSP_Port parameters (see A.2.1 and A.2.2) at start-up.

11.5.6.10 Loop-back

A special addressing error is the so-called loopback error case, where the sender receives back its own sent message.

Countermeasures:

SDCI-FS provides for inverted values of MCount as DCount and inverted values of the Port number (PortNum) returned from the FS-Device.

11.5.7 Start-up (communication)

An FS-Device starts always after an FS-Master since the FS-Master shall be the only one to power-up at least the communication part of the FS-Device. Both devices usually require time for safety self-tests that may exceed the standard timings defined in IEC 61131-9.

Due to the initial behavior of an FS-Device as an OSSDe, the start-up is coordinated and specified in 5.7, 7.2, and 7.3.

The start-up of the underlying SDCI communication system is specified in IEC 61131-9 and illustrated in an abstract and simplified manner in Figure 57 for easier comprehension. Any deviating FSP authenticity or protocol parameter CRC signature shall lead to a safe state of the particular FS-Master Port and prevent the SCL from being started.

11.6 SCL management

11.6.1 Parameter overview (FSP and FST)

Annex A specifies several functional safety related parameters for communication protocol services (FSP) as well as for the handling and integrity purposes of FS-Device technology parameters (FST).

The parameters are subdivided into 4 groups:

- Authenticity
- Safety communication
- FS-I/O structure description
- Auxiliary parameters

The authenticity parameters combine the safety connection ID ("A-Code") of the FS-Master (assigned by the upper-level FSCP system) with the Port number of the connected FS-Device. Due to the point-to-point nature of the FS-Device communication with its Master, a one-time check at start-up is sufficient to ensure authenticity (see 11.7.4).

The Safety Communication Layers (SCL) require parameters for protocol versions, protocol modes such as CRC-16 or CRC-32, watchdog for timeliness, CRC signature to secure technology parameters, and a CRC signature to secure the safety communication parameters.

The next group contains a description of the FS I/O data structure, the FS-Device wants to exchange with the FSCP-Host. This description facilitates the mapping to the description which some FSCP systems require for set-up. During the mapping process the FS-Master tool appends the qualifier bits, which are necessary for Port-selective passivation.

Auxiliary parameters are specified for several purposes. For example, to secure the functional safety parameter description within the IODD, to support the automatic calculation of approximate values of safety function response times, and to inform about the start-up self-testing time of an FS-Device until the Ready pulse appears.

Figure 54 shows an overview of the components and the activities around parameterization.

An FS-Master as a gateway comes with an associated description file for the upper-level system (FSCP). With the help of an engineering tool and these parameters, the FS-Master receives during commissioning for example its FSCP connection ID ("A-Code" for authenticity) and its FSCP watchdog time ("T-Code" for timeliness). Thus, the FSCP communication cycles are independent from the SDCI-FS communication cycles between FS-Master and FS-Device.

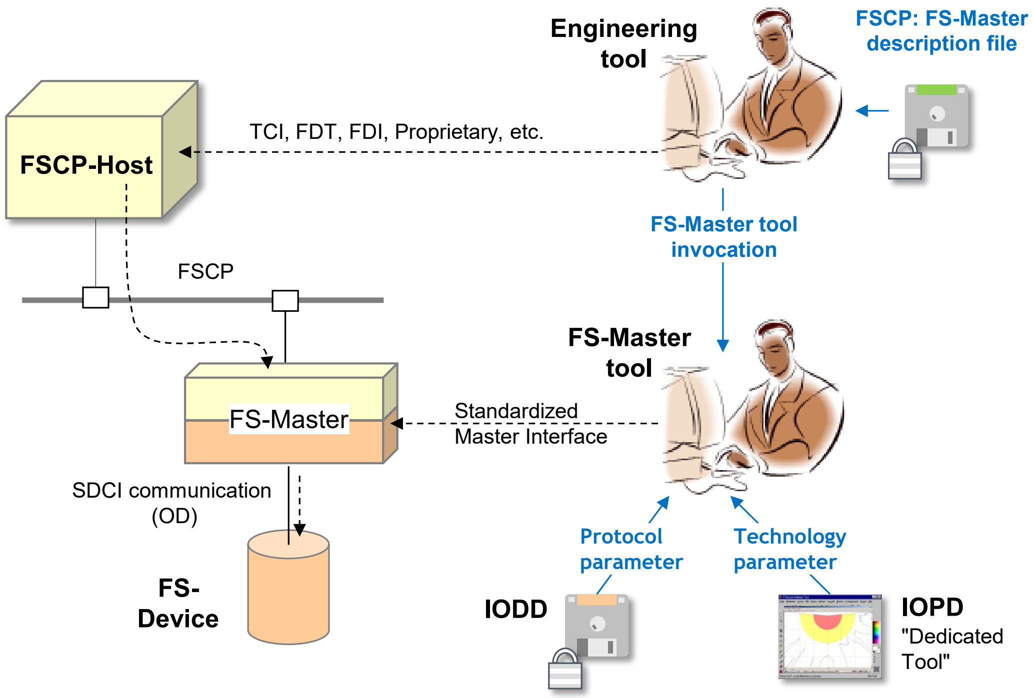


Figure 54 – Parameter types and assignments

An FS-Master with its SDCI side can be configured and parameterized with the help of its FS-Master tool. The IODD of an FS-Device contains besides the non-safety parameters also the safety section with the parameters in Annex A. The parameters can be set-up off-site or online the same way as with a non-safety system during "commissioning-test" (see Table G.1). The FSCP authenticity parameter can be copied from the engineering tool display to the FS-Master tool display (see A.2.1).

It is possible to describe a small set of technology parameters (FST) in a non-safety manner, thus allowing the usage of the SDCI standard Data Storage mechanism as described in 9.4.

However, a separate Dedicated (IOPD) Tool, developed according to IEC 61508-3 shall be used to calculate a CRC signature across the instances of the FST parameters. This CRC signature shall be entered into the respective FSP parameter (see A.2.8).

The IOPD tool uses a new standardized IOPD communication interface (DTI, see Annex F) and the same path to the FS-Device as the FS-Master tool itself.

11.6.2 Parameterization approaches

11.6.2.1 FS-Master-centric

The configuration and parameterization of a stand-alone SDCI-FS system corresponds mainly to the approach described in 11.6.1. The authenticity uses a default value in this case (see A.2.1).

Figure 54 shows a loosely coupled system, where the parameterization is performed within the SDCI-FS part. Within the FSCP system, predefined FS I/O data structures are available and can be selected during commissioning.

11.6.2.2 FSCP-Host-centric

Some automation application areas prefer an FSCP-Host-centric approach. In this case, all parameters are expected to be stored within the FSCP-Host and downloaded at start-up into the FS-Master (FSCP-subsystem) and further down into the FS-Device.

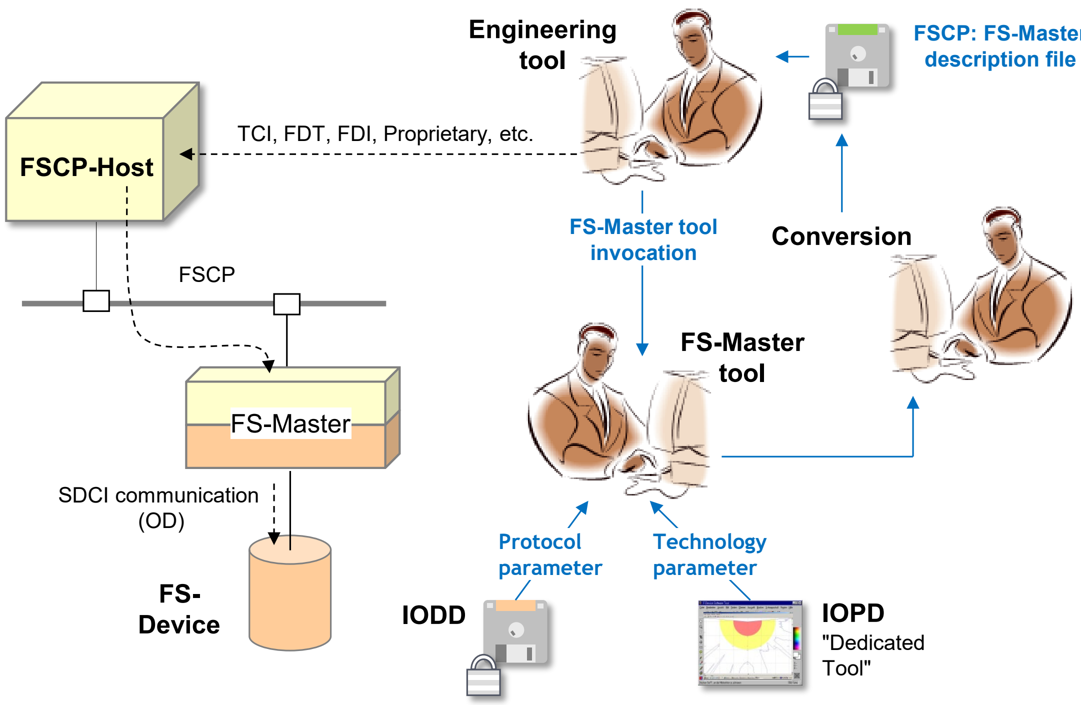


Figure 55 – FSCP-Host-centric system

Due to the fieldbus-independent design of SDCI and SDCI-FS, all parameters can for example be converted into the fieldbus device description file. It is one of the objectives of SDCI-FS to optimize the design of safety parameters such that an efficient conversion is possible.

11.7 Integrity measures

11.7.1 IODD integrity

The parameters specified in Annex A are coded in an IODD file using XML. In order to protect the safety parameter description within this file, a CRC signature ("FSP_ParamDescCRC") shall be calculated across its safety-related parts (see Annex D and Annex E.5.6). Usually, the IODD file travels many ways and the CRC signature helps detecting potentially scrambled bits.

11.7.2 Tool integrity

When opening the IODD, the FS-Master tool (interpreter of the IODD file) shall calculate the CRC signature across the safety-related parts and compare the result with the parameter "FSP_ParamDescCRC".

During the data manipulations within the FS-Master tool as well as within Device Tools/IOPDs ("Dedicated Tools") such as display, intended modification, storage/retrieval, and down/upload, parameter values could become incorrect. It is the responsibility of the designer to develop the software tools according to the software safety level requested in ISO 13849-1 or IEC 61508-3.

11.7.3 Transmission integrity

Since communication between the FS-Master tool and the FS-Device is proprietary, it is the responsibility of the FS-Master tool to ensure transmission integrity and authenticity, for example through CRC signatures and/or read back (see Table 27 and D.3.1).

11.7.4 Verification record

In either the FS-Master-centric or in the FSCP-Host-centric approach an FSP_VerifyRecord of parameter data is stored in the FS-Master per Port/FS-Device as shown in Figure 56.

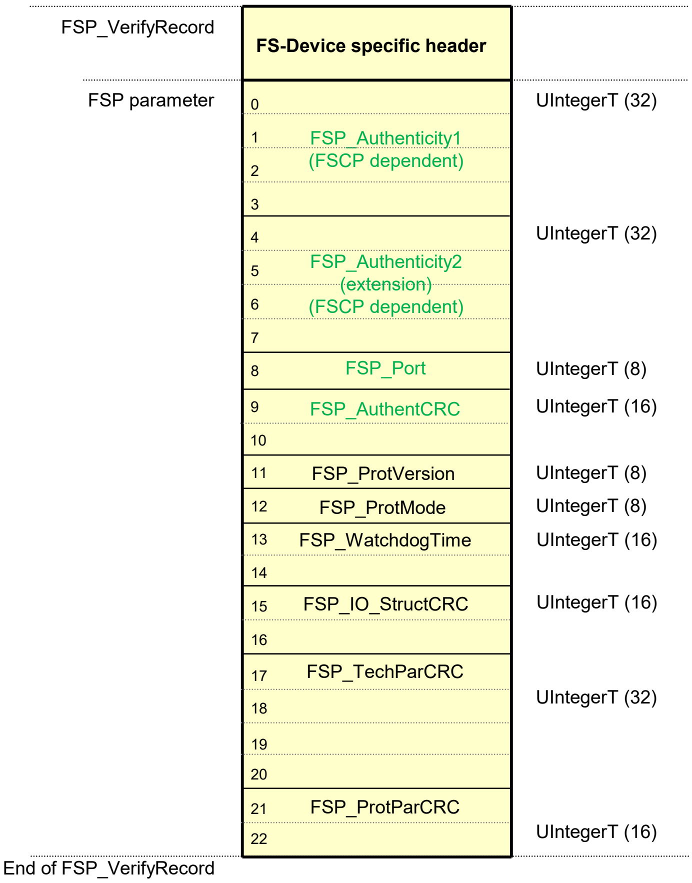


Figure 56 – Structure of the FSP_VerifyRecord

The authenticity parameters are secured by FSP_AuthentCRC for transmission from FS-Master tool to FS-Master and further to the FS-Device. The procedure of the FSCP authenticity acquisition from the FSCP gateway and subsequent handling of the FSP authenticity record is described in 10.4.3.3. FSP_ProtParCRC secures protocol parameters as described in 10.4.3.4.

11.7.5 Authentication

The SLM of the FS-Master uses the FSP_VerifyRecord received from Configuration Manager. Thus, the FSP_Authenticity codes within the record can be compared with the actual FSCP Authenticity values in the safety part of the Gateway.

11.7.6 Storage integrity

Both records (authenticity and protocol) of Figure 56 are stored in both FS-Master and FS-Device and may fail over time (see also Table A.1).

At each regular start-up, the Configuration Manager transfers the FSP_VerifyRecord to the FS-Device during PREOPERATE as shown in Figure 57 and described in 10.4.3.1 and A.2.10. Figure 57 is derived from the Master message handler in IEC 61131-9 and provides an abstract and simplified picture of the essential states and transitions.

The FS-Device will detect a potential mismatch between the downloaded authenticity parameter set and the already stored values in the FS-Device, for example if FS-Devices are misconnected to a different Port or even to a different FS-Master (see Figure 36).

The protocol parameters are propagated to the Safety Communication Layer at each start-up.

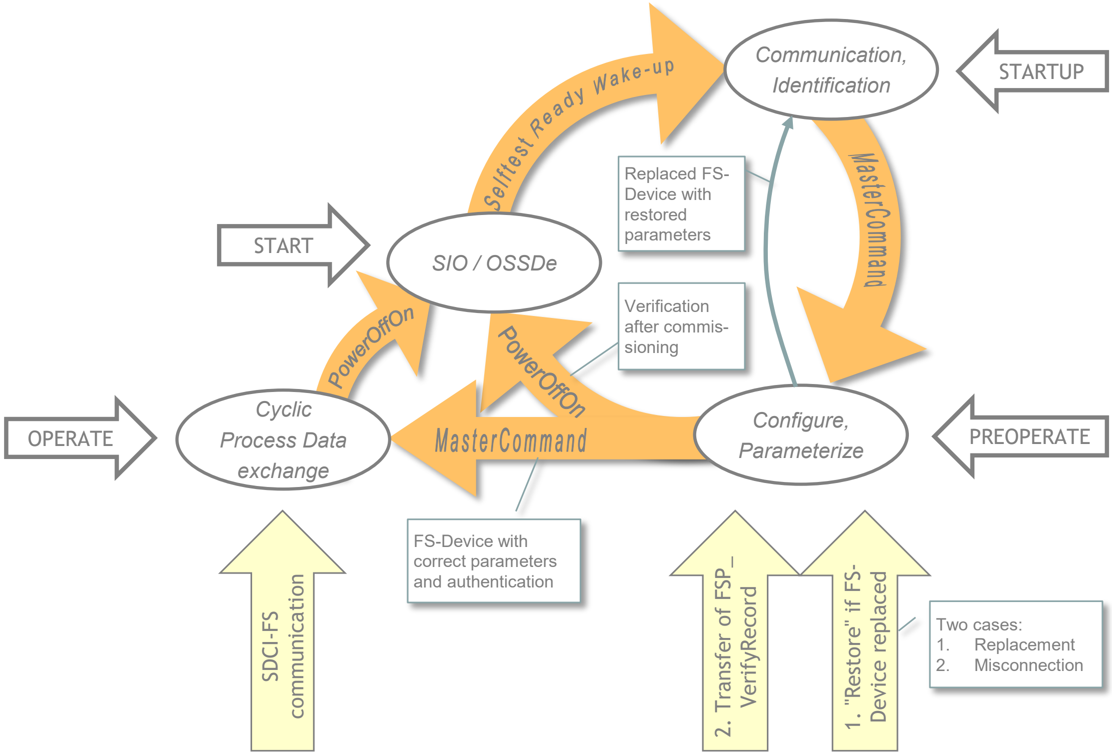


Figure 57 – Start-up of SDCI-FS

In case the FS-Device has been replaced due to a failure, the technology specific parameters (FST) and the FSP parameters are "restored" from Data Storage if the FS-Device carries all authenticity parameters = "0". If Authenticity is not "0", the FS-Device shall ignore them and keep the existing (see 9.4, E.5.7, and step 1. in Figure 57). In this case a misconnection can be assumed, or the FS-Device has already been in use and requires testing and a reset of the authenticity parameters (see Annex G).

11.7.7 FS I/O data structure integrity

All I/O data of the connected FS-Devices should be mapped in an efficient manner into the FSCP I/O data as shown in 12.1.

Due to the additional qualifier bits required for Port-selective passivation, the original FS-Device specific data structure is not directly visible within the FSCP I/O data structure exchanged with the FSCP-Host.

The safety-related interpreter of the FS-Master tool transfers the entire instance data together with the CRC signature to the FS I/O data mapper as shown in 10.4.3.1 (see also A.2.7).

11.7.8 Technology parameter (FST) based on IODD

One of the objectives of SDCI-FS is FS-Device exchange without tools by using the original data storage mechanism of SDCI. As a precondition, the FST-parameter description is required within the IODD (see E.5.7).

The FST parameters are displayed in this case within the FS-Master tool (see Figure 58, FST-Parameters section). Values can be assigned as for non-safety parameters only during "commissionning-test" (see Table G.1).

The user is responsible for correct values within the FS-Device using adequate validation procedures. The FS-Master tool can assist for example via read back and display of the parameters.

Securing of the FST parameter via signature shall not be performed by the FS-Master tool. A separate "Dedicated Tool" (Device tool) provided by the FS-Device manufacturer shall be used instead as shown in Figure 59 and explained in the following.

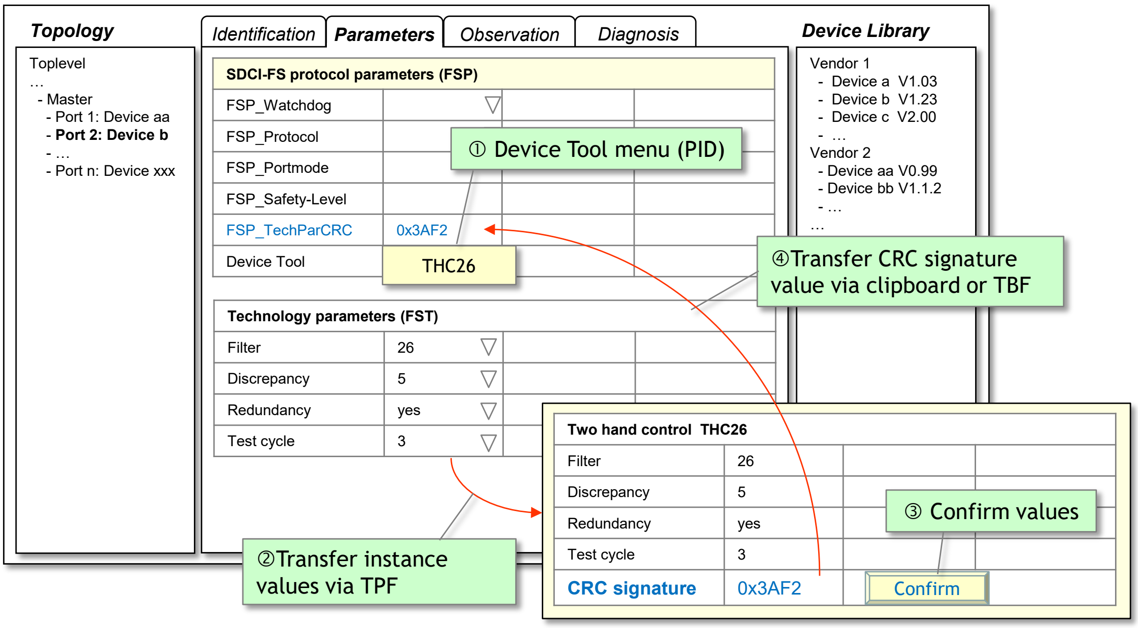


Figure 58 – Securing of FST parameters via dedicated tool

After test and validation, the Device tool is invoked via menu (step①). Instance values are transferred via TPF (step②) and displayed again. The user compares the instance values and confirms the correctness via the "Confirm" button (step③). The Device tool then calculates the CRC signature across the instance data of the FST parameters (see "CRC signature" in Figure 58), which can be copied and pasted or transferred via TBF into the "FSP_TechParCRC" field of the FSP parameters (step ④).

Since this parameter is part of the FSP parameter block, the FS-Device can check the integrity of these FST parameters together with the protocol parameters.

11.7.9 Technology parameter (FST) based on existing Dedicated Tool (IOPD)

In cases, where existing safety devices already have their PC program with password protection, wizards, teach-in functions, verification instructions, online monitoring, diagnosis, special access to device history for the manufacturer, etc., an FST parameter description may not be available. Figure 59 shows an example.

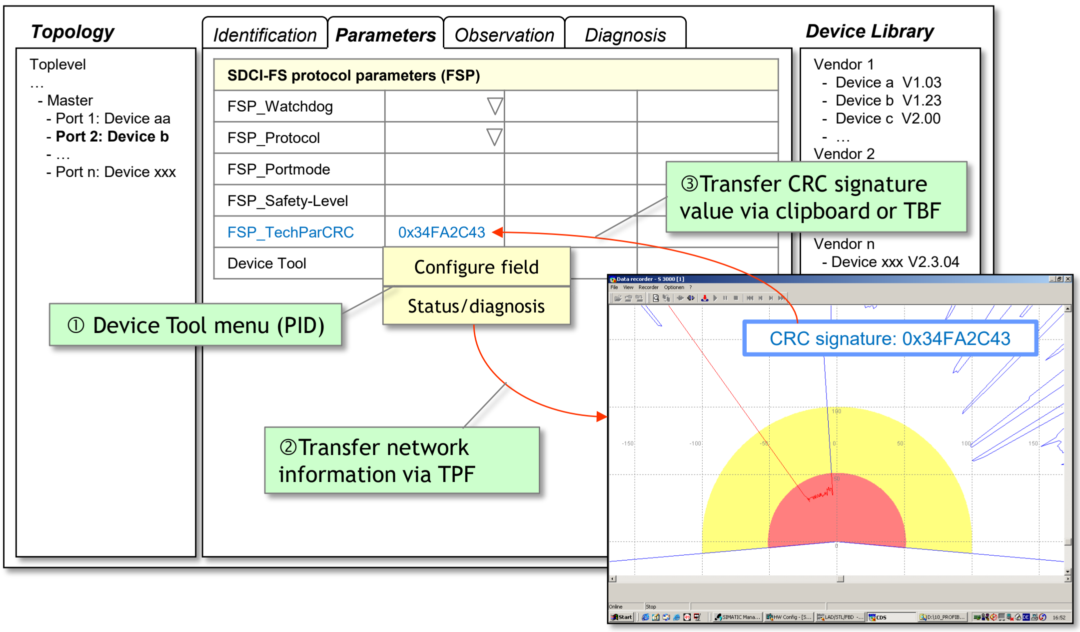


Figure 59 – Modification of FST parameters via Device tool

Such a Device tool requires communication with its particular FS-Device and therefore access to a Communication Server (see Annex F.5). It can be invoked via menu entries (step①) and thus jump directly into for example configuration or status/diagnosis functions. Network information is transferred via TPF (step②). After test and validation, it shall provide a display of the calculated CRC signature across the instance data, which can be copied and pasted into the "FSP_TechParCRC" field of the FSP parameters (step③).

These FS-Devices shall be supported by the data storage mechanism of SDCI and an FS-Device replacement without tools is possible.

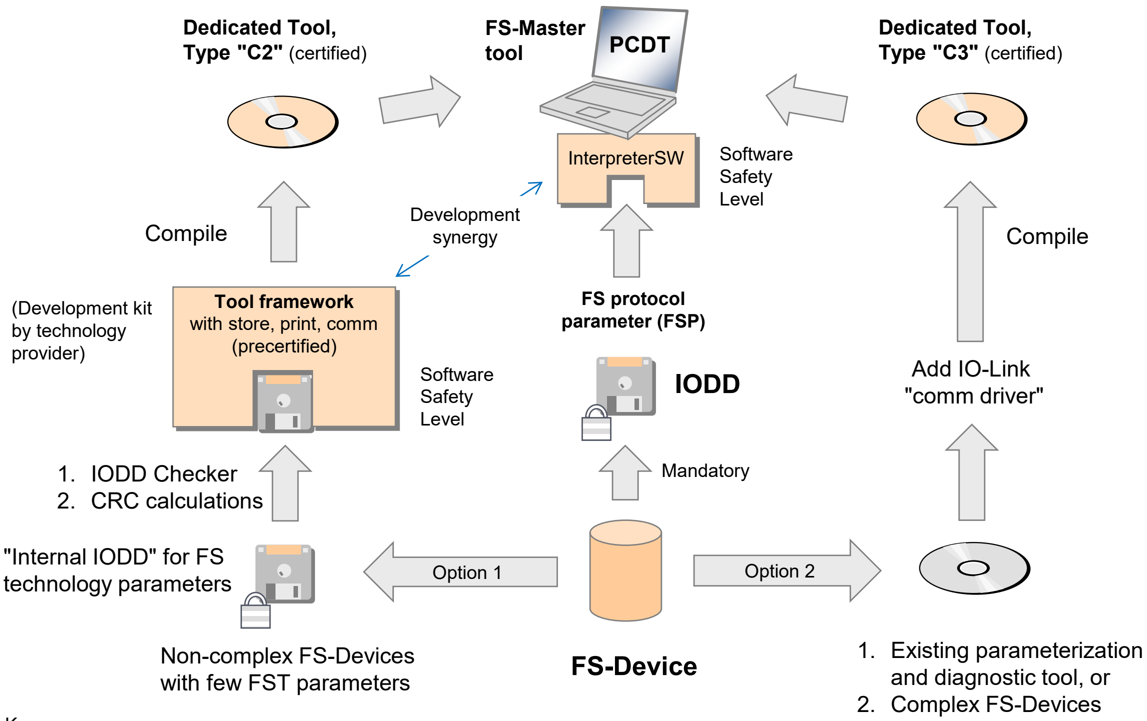
The Data Storage limit per FS-Device is 2048 octets according to IEC 61131-9.

11.8 Creation of FSP and FST parameters

Standards for "Safety-for-Machinery" such as ISO 13849-1 and IEC 62061 require "Dedicated Tools" for the parameterization of safety devices. For the ease of development and logistics of software tools it is recommended to use the process described in Figure 60.

For FS-Devices with only a few FST parameters, no business logic, and no wizard and help systems, one particular "Dedicated Tool Framework" could be developed in a safe manner according to IEC 61508-3 and equipped with the necessary communication interfaces.Technology provider can provide such a framework for the FST parameters of a particular FS-Device (Option 1 in Figure 60). FS-Device developers can individualize the framework using the brand name, company name, and FS-Device identifiers to one dedicated tool (IOPD). This executable Dedicated Tool software can be certified by assessment bodies.

For FS-Devices with more complex FST parameters, the IOPD can be developed individually, or existing tools can be used. In both cases the tools can be equipped with the necessary communication interfaces (Option 2 in Figure 60).



Key
IOPD = IO Parameterization and Diagnostic
Software Safety Level according ISO 13849 and IEC 61508-3

Figure 60 – Creation of FSP and FST parameters

In any case, the dedicated tool (IOPD) shall calculate and display the CRC signature across all FST parameters. This signature can be copied into the entry field of the FSP parameter "FSP_TechParCRC", such that an FS-Device can verify the correctness of locally stored FST parameters after start-up and download of the FSP parameter set to the FS-Device.

For each and every FS-Device the same set of FSP (protocol) parameters shall be created in an extended IODD for SDCI-FS. This IODD is mandatory and contains the usual conventional parameters and additionally the FSP parameters.

11.9 Integration of Dedicated Tools (IOPD)

11.9.1 IOPD interface

Usually, a so-called Master tool (PCDT) provides engineering support for a Master and its Devices via Device descriptions in form of XML files (IODD). In principle, this is the same for FS-Master and FS-Device. For functional safety besides an extended IODD it is necessary for an FS-Device vendor to provide an additional Dedicated Tool (IOPD) as shown in Figure 60.

In order for the IOPD to communicate with its FS-Device a new standardized communication interface is required.

11.9.2 Standard interfaces

Usually, Master tools are integrated using existing standards such as FDT, the FDI (see [10]), or proprietary solutions. Such a variety is not acceptable for FS-Devices and therefore, easy, and proven-in-use technology has been selected and adopted for SDCI-FS. It is called "Device Tool Interface" (DTI).

Annex F provides the specification for this interface.

Figure 61 illustrates the communication hierarchy of FDT and others for the fieldbus as well as the connection via the "Device Tool Interface" and the underlying SDCI communication.

The FS-Device tool (IOPD) does not have to know about the fieldbus environment it is connected to. The example in Figure 61 illustrates how it sends a "Read Index 0x4231" service and how the FS-Master tool packs this service into a fieldbus container and passes it to the fieldbus communication server.

The addressed FS-Master is connected to the fieldbus and receives this container. It unpacks the SDCI Read service and performs it with the addressed FS-Device connected to a Port.

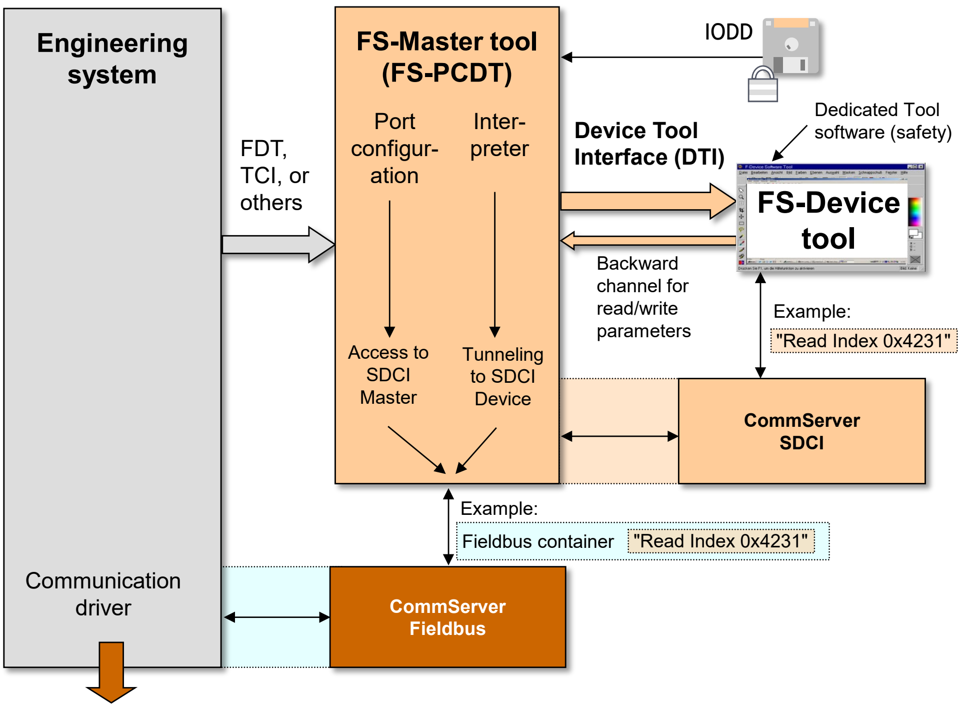


Figure 61 – Example of a communication hierarchy

11.9.3 Backward channel

An FS-Master vendor does not know in advance which FS-Devices a customer wants to connect to the FS-Master Ports. As a consequence, the fieldbus device description of such an FS-Master can only provide predefined "containers" for the resulting I/O data structure of the FS-Device ensembles connected to the Ports. In functional safety this is even more complicated since the description of the data structures shall be coded and secured.

Because of the variety of different configurations and parameterizations of a particular FS-Device, it usually for example

- requires different I/O data structures to exchange with PLCs or hosts,
- has different reaction times due to configured high or low resolutions, and
- can have different SIL, PL, category, or PFH values impacting the overall safety level of a safety function.

The classic "fieldbus device description" to inform an engineering system is not flexible in this respect. Its advantage is the testability and certification for its interoperability with engineering tools.

Nevertheless, a "backward channel" within the tool interfaces allows for nowadays flexible manufacturing and ease of engineering and commissioning. An example is specified in [11] Clause 4.15.5.

F.3.5 and F.9.4 specify the features for this "backward channel".

11.10 Validation

It is the responsibility of the FS-Device designer to specify the necessary verification and validation steps (for example tests; see H.6) within the user/safety manual and/or within the "Dedicated Tool" (IOPD).

11.11 Passivation

11.11.1 Motivation and means

Figure 62 illustrates the motivation for Port selective passivation. Usually for efficiency reasons, the signals 0 to 7 of FS-Devices connected to Ports are not mapped individually to an FSCP SPDU, but rather packed into one FSCP SPDU. Each of these signals can be assigned to a separate safety function n to n+7. If a fault occurs in one of the signal channels, a collective passivation for the entire FSCP SPDU would be necessary causing all safety functions to trip.

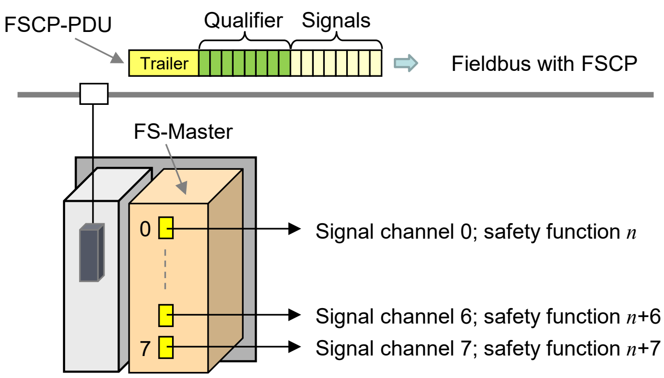


Figure 62 – Motivation for Port selective passivation

FSCPs usually provide so-called qualifier bits associated to the signal process data, which enable selectively passivating that particular signal channel and the associated safety function.

Safety of machinery usually requires an operator acknowledgment after repair of a defect signal channel to prevent from automatic restart of a machine. It is not necessary to provide the acknowledgment for each signal channel and it can be one for all channels.

11.11.2 Port selective (FS-Master)

In 11.11.1 a use case is described where the signal channel corresponds directly to a particular FS-Device. The qualifier and acknowledgment mechanism shall be implemented within the FS-Master in accordance with the specifications of the particular FSCP.

It can be helpful for the user to provide an indication in each FS-Device that an operator acknowledgment is required prior to a restart of a safety function. CB0 (ChFAckReq) within the Control&MCnt octet (see Table 31) shall be used for that purpose. It is not safety related.

Optionally, in case of FS_PortMode "OSSDe" (see 10.4.2), the signal ChFAckReq can be connected separately to the corresponding FS-Device indication (see Clause H.1).

11.11.3 Signal selective (FS-Terminal)

Figure 13 shows the use case of an FS-Terminal where an FS-Device provides several signal channels to switching devices such as E-Stop buttons.

For those FS-Devices the design rules in 11.4.9.3 apply. The acknowledgment mechanisms shall be implemented within the safety Process Data.

11.11.4 Qualifier settings in case of communication

Figure 63 illustrates the embedding of the qualifier handler in case of FS_PortModes "SafetyCom" and "MixedSafetyCom" (see 10.4.2). The services/signals "FAULT_S", "SDset_S", "ChFAckReq_S", and "ChFAck_C" are specified in 11.3.2 and 11.5.2.

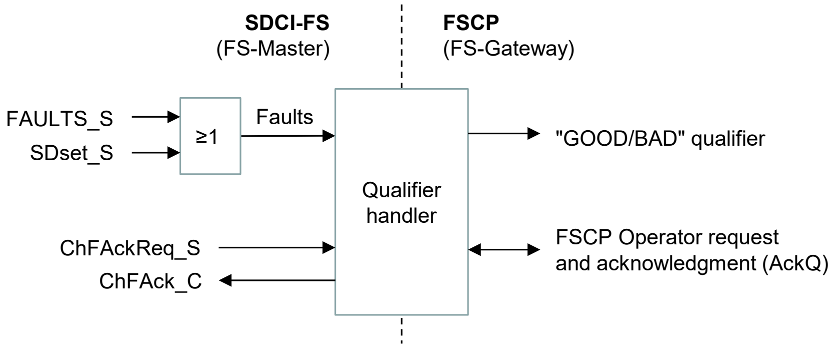


Figure 63 – Qualifier handler (communication)

The qualifier bits "GOOD/BAD" shall be set according to the definitions in Table 42 during the FSCP mapping procedure.

Table 42 – Qualifier bits "GOOD/BAD"

Faults	Qualifier	Signal state
0	GOOD	1
1	BAD	0

11.11.5 Qualifier handling in case of OSSDe

Figure 64 illustrates the embedding of the qualifier handler in case of FS_PortModes "OSSDe" (see 10.4.2). Definitions of Table 42 apply.

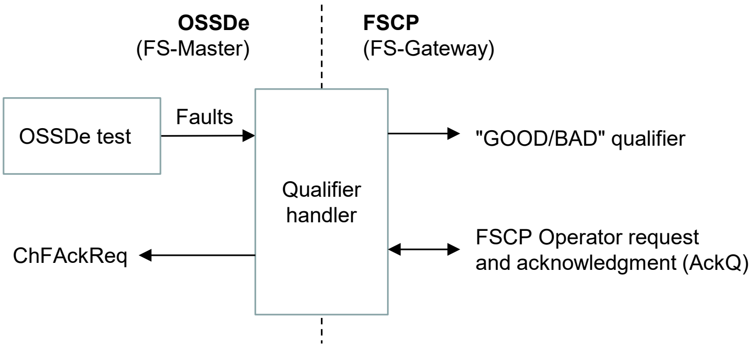


Figure 64 – Qualifier handler (OSSDe)

Figure 65 shows the state machine for the behavior of the qualifier handler (OSSDe).

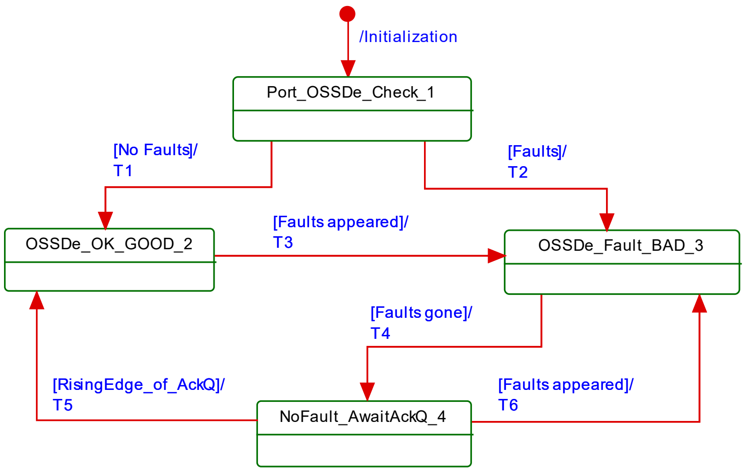


Figure 65 – Qualifier behavior per FS-Master Port

Table 43 shows the state and transition table for the qualifier behavior.

Table 43 – State transition table for the qualifier behavior

STATE NAME	STATE DESCRIPTION
Initialization	Use SD, Qualifier = BAD, ChFAckReq =0
1 Port_OSSDe_Check	Perform Port diagnosis to detect Faults
2 OSSDe_OK_GOOD	Perform Port diagnosis cyclically to detect Faults
3 OSSDe_Fault_BAD	Perform Port diagnosis cyclically to detect Faults
4 NoFault_AwaitAckQ	Wait on rising edge of AckQ

TRAN-SITION	SOURCE STATE	TARGET STATE	ACTION
T1	1	2	Use PD, Qualifier = GOOD, AckQ = 0, ChFAckReq =0
T2	1	3	Use SD, Qualifier = BAD, AckQ = 0, ChFAckReq =0
T3	2	3	Use SD, Qualifier = BAD, AckQ = 0, ChFAckReq =0
T4	3	4	Use SD, Qualifier = BAD, AckQ = 0, ChFAckReq =1
T5	4	2	Use PD, Qualifier = GOOD, AckQ = 1, ChFAckReq =0
T6	4	3	Use SD, Qualifier = BAD, AckQ = 0, ChFAckReq =0

INTERNAL ITEMS	TYPE	DEFINITION
RisingEdge_of_AckQ	Flag	Means to prevent from permanently actuating the AckQ signal.
AckQ	Flag	Flag depending on the individual upper-level FSCP system and its mapping.

INTERNAL ITEMS	TYPE	DEFINITION
Faults	Flag	Any detected fault such as a wire break, short circuit.
ChFAckReq	Flag	Signal set by qualifier handler (see 11.11.2 and H.1)

11.12 SCL diagnosis

The Safety Communication Layer can create its own EventCodes such as CRC error, counter error, or timeout as listed in Clause B.1.

12 Functional safe processing (FS-P)

12.1 Recommendations for efficient I/O mappings

Figure 66 shows how efficiency can be increased when packing I/O data from the connected safety devices into one FSCP SPDU instead of several individual FSCP SPDUs. On the left, the bits of safety devices (OSSD) are packed into one FSCP SPDU by the FS-DI module. On the right, the FS-Devices use each an FSCP SPDU through the FS-Master/Gateway (FS-M/G) to transmit a bit. In the middle, an FS-M/G packs several bits into one FSCP SPDU similar to an FS-DI.

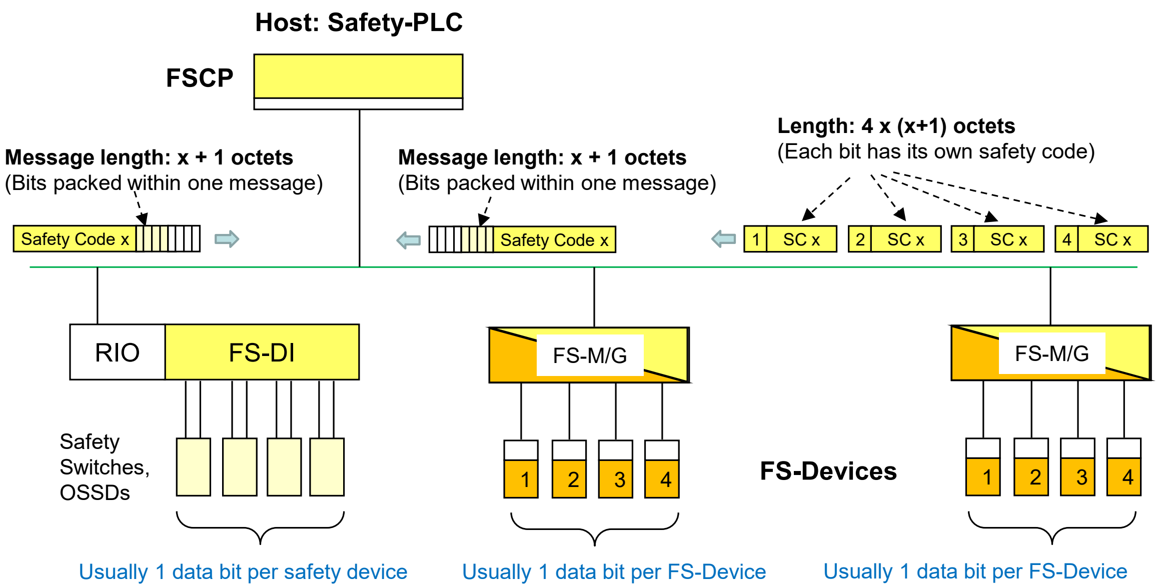


Figure 66 – Mapping efficiency issues

The FS I/O data structure shall be created as a multiple of 16 bits.

12.2 Embedded FS controller

Specification and implementation of an embedded FS controller as described in 4.2 and Figure 1 to provide "local" safety functions on SDCI-FS level are manufacturer's responsibility and not standardized.

Annex A
(normative)

Extensions to parameters

A.1 Indices and parameters for SDCI-FS

The Index range reserved for SDCI-FS includes 255 Indices from 0x4200 to 0x42FF.

Table A.1 shows the specified Indices for SDCI profiles, the protocol parameters (FSP) of SDCI-FS, comprising authenticity, protocol, I/O data structure records, and auxiliary parameters as well as the total reserved range for SDCI-FS, and the second range of Indices for SDCI profiles.

Table A.1 – Indices for SDCI-FS

Index (dec)	Sub index	Object name	Access	Length	Data type	M/O/C	Purpose/reference
...							
0x4000 to 0x41FF		Profile specific Indices					For example: Smart sensors
Authenticity (11 octets)							
0x4200 (16896)	1	FSP_Authenticity_1	R/W	4 octets	UIntegerT	M	"A-Code" from the upper-level FSCP system; see A.2.1
	2	FSP_Authenticity_2	R/W	4 octets	UIntegerT	M	Extended "A-Code" from the upper-level FSCP system
	3	FSP_Port	R/W	1 octet	UIntegerT	M	PortNumber identifying the particular FS-Device; see A.2.2
	4	FSP_Authent CRC	R/W	2 octets	UIntegerT	M	CRC-16 across authenticity parameters; see A.2.3
Protocol (12 octets)							
0x4201 (16897)	1	FSP_ ProtVersion	R/W	1 octet	UIntegerT	M	Protocol version: 0x01; see A.2.4
	2	FSP_ ProtMode	R/W	1 octet	UIntegerT	M	Protocol modes, e.g. 16/32 bit CRC; see A.2.5
	3	FSP_ Watchdog	R/W	2 octets	UIntegerT	M	Monitoring of I/O update; 1 to 65 535 ms; see A.2.6
	4	FSP_IO_ StructCRC	R/W	2 octets	UIntegerT	M	CRC-16 signature across I/O structure description block; see A.2.7
	5	FSP_TechParCRC	R/W	4 octets	UIntegerT	M	Securing code across FST (technology specific parameter); see A.2.8
	6	FSP_ProtParCRC	R/W	2 octets	UIntegerT	M	CRC-16 across protocol parameters; see A.2.9
Verification Record (23 octets)							
0x4202 (16898)		FSP_VerifyRecord	W	23 octets	RecordT	M	FS-Master sends this verification record consisting of authenticity and protocol parameters at PREOPERATE. This Index is hidden to the user; see A.2.10
Auxiliary parameters							
0x4210 (16912)		FSP_TimeToReady	R	2 octets	UIntegerT	M	Time to Ready pulse; 1 to 32 767 (in 10 ms), see A.2.11
0x4211 (16913)		FSP_MinShutDown Time	R	2 octets	UIntegerT	M	Minimum time for the FS-Device to shut down after Port Power off; 100 to 1 000 (in 10 ms), see A.2.12

Index (dec)	Sub index	Object name	Access	Length	Data type	M/O/C	Purpose/reference
0x4212 (16914)		FSP_ ParamDescCRC	R	4 octets	UIntegerT	M	CRC-32 signature securing authenticity, protocol, and FS I/O structure description within IODD; see A.2.15
0x4213 (16915)		FSP_WCDT	R	2 octets	UIntegerT	M	Worst-case delay time; 1 to 32 767 (in ms), see A.2.13 and H.6
0x4214 (16916)		FSP_OFDT	R	2 octets	UIntegerT	M	One fault delay time; 1 to 32 767 (in ms), see A.2.14 and H.6
0x4215 (16917) to 0x42FF (17151)		Reserved for SDCI-FS					
0x4300 to 0x4FFF		Profile specific Indices					For example: BLOB and Firmware update
...							
Key	M = mandatory; O = optional; C = conditional						

A.2 Parameters in detail

A.2.1 FSP_Authenticity

During off-line commissioning of an SDCI-FS project, the default value of this parameter is "0". During on-line commissioning, the user acquires the FSCP authenticity ("A-Code") from the FS-Master via SMI service and propagates it to the FS-Device within an entire record as described in 10.4.3.1. The FS-Master tool shall only transfer entire authenticity blocks to the FS-Device with correct CRC signature values such that the FS-Device can check plausibility and correctness (see A.2.3).

In case the system is armed (FSP_TechParCRC ≠ "0") the FS-Device compares at each start-up (DS_Change or PortPowerOffOn) its locally stored values with the values of the FSP_VerifyRecord to detect any misconnection (incorrect Port or FS-Master), see Annex G.

Some FSCPs provide extended authenticity. In those cases, the extended code shall be included in this parameter.

Padding bits and octets shall be filled with "0".

A.2.2 FSP_Port

The FS-Master tool identifies the FS-Master PortNumber of the attached FS-Device and stores it in this parameter. Storage and checking of the parameter by the FS-Device corresponds to A.2.1 and A.2.3. Numbering starts at "1". Thus, the FS-Device shall not accept a "0".

Default PortNumber in IODD is "0" and means PortNumber of a particular Device has not been assigned yet.

A.2.3 FSP_AuthentCRC

The FS-Master tool shall only transfer entire authenticity blocks to the FS-Device including FSP_Authenticity and FSP_Port (see Table A.1).

For the CRC signature calculation of the entire authenticity block, the CRC-16 in Table D.1 shall be used. This CRC polynomial has a Hamming distance of ≥ 6 for lengths ≤ 16 octets. A seed value "0" shall be used (see D.3.6).

A.2.4 FSP_ProtVersion

Table A.2 shows the coding of FSP_ProtVersion.

Table A.2 – Coding of protocol version

Value	Definition
0x00	Not permitted
0x01	This protocol version
0x02 to 0xFF	Reserved

A.2.5 FSP_ProtMode

Table A.3 shows the coding of FSP_ProtMode. The "test mirrors" are used by testers.

Table A.3 – Coding of protocol mode

Value	Definition
0x00	Not permitted
0x01	0 to 3 octets of FS I/O Process Data; CRC-16
0x02	0 to 25 octets of FS I/O Process Data; CRC-32
0x03 to 0xF8	Reserved
0xF9	Test mirror in case of CRC-32 (reserved for test)
0xFA	Test mirror in case of CRC-16 (reserved for test)
0xFB to 0xFF	Reserved

A.2.6 FSP_Watchdog

The FS-Device designer determines the I/O update time and uses it as default value of this parameter within the IODD. The I/O update time is the time period between two safety PDUs with subsequent counter values (I/O samples) including possible repetitions within the SDCI communication layer (black channel; see 11.5.5).

With the help of the parameter default value (I/O update time), the transmission times of the safety PDUs, and FS-Master processing times, the FS-Master tool can estimate the total time and suggest the value of the "FSP_Watchdog" parameter.

The value range is 1 to 65 535 (measured in ms). A value of "0" is not permitted. The SCL of the FS-Device is responsible to check the validity at start-up and to create an error in case (see Table B.1).

A.2.7 FSP_IO_StructCRC

An IODD-based non-safety viewer can be used to calculate this 16-bit CRC signature across the FS I/O structure description within the IODD during the development phase. The algorithm for the calculation is shown in Annex D. A seed value "0" shall be used (see D.3.6).

The safety-related interpreter of the FS-Master tool transfers the entire instance data together with the CRC signature to the FS I/O data mapper as shown in 10.4.3.1.

Table A.4 shows Version "1" of the generic FS I/O data structure description for FS-Devices. With the help of this table, individual instances of FS-Device I/O Process Data can be created via IODD and, amongst others, used for an automatic mapping of SDCI-FS data to FSCP safety data.

Table A.4 – Generic FS I/O data structure description

Item name	Item length	Definition
IO_DescVersion	1 octet	Version of this generic structure description: 0x01
InputDataRange	1 octet	Length in octets of the entire FS input Process Data including the 4 or 6 octets respectively for the safety code (Control/Status, PortNumber, and CRC-16/32)
TotalOfInBits	1 octet	Number of the entire set of input BooleanT (bits)
TotalOfInOctets	1 octet	Number of octets with input BooleanT (including unfilled octets)
TotalOfInInt16	1 octet	Number of input IntegerT(16)
TotalOfInInt32	1 octet	Number of input IntegerT(32)
OutputDataRange	1 octet	Length in octets of the entire FS output Process Data including the 4 or 6 octets respectively for the safety code (Control/Status, PortNumber, and CRC-16/32)
TotalOfOutBits	1 octet	Number of the entire set of output BooleanT (bits)
TotalOfOutOctets	1 octet	Number of octets with output BooleanT (including unfilled octets)
TotalOfOutInt16	1 octet	Number of output IntegerT(16)
TotalOfOutInt32	1 octet	Number of output IntegerT(32)
FSP_IO_StructCRC	2 octets	CRC-16 signature value across all items (see Annex D.1)

Figure A.1 shows the instance of the FS I/O data description of the example in Figure A.2.

```
IO_DESCVERSION         01      0x01
INPUT_DATA_RANGE       10      0x0A
TOTAL_OF_INBITS        13      0x0D
TOTAL_OF_INOCTETS      02      0x02
TOTAL_OF_ININT16       01      0x01
TOTAL_OF_ININT32       00      0x00
OUTPUT_DATA_RANGE      06      0x06
TOTAL_OF_OUTBITS       00      0x00
TOTAL_OF_OUTOCTETS     00      0x00
TOTAL_OF_OUTINT16      00      0x00
TOTAL_OF_OUTINT32      00      0x00
FSP_IO_STRUCTCRC    39464      0x9A28
```

Figure A.1 – Instance of an FS I/O data description

Figure A.2 shows an example with FS input Process Data and no FS output Process Data.

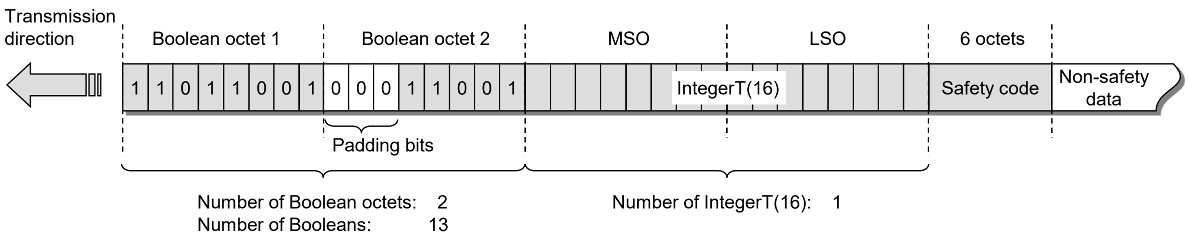


Figure A.2 – Example FS I/O data structure with non-safety data

A.2.8 FSP_TechParCRC

This document specifies two basic methods for the assignment of technology specific parameters (FST). The FS-Device designer is responsible for the selection of the securing method.

The method in 11.7.8 is based on IODD and suggests using one of the CRC generator polynomials in Table D.1. If calculation of the CRC signature value results in "0", a "1" shall be used.

The method in 11.7.9 depends on an existing FS-Device tool (Dedicated Tool). Whatever method is used, the tool shall display a securing code after verification and validation that can be copied and pasted into the FSP_TechParCRC parameter entry field.

During commissioning a value of "0" can be entered to allow for certain behavior at start-ups of the FS-Device (see 10.4.3.1). During production, this value shall be ≠ "0".

For technology specific parameter block transfers > 232 octets, the SMI_PortCmd service CMD = "0" (DeviceParBatch) specified in IEC 61131-9 can be used.

A.2.9 FSP_ProtParCRC

The FS-Master tool shall only transfer entire protocol blocks to the FS-Device including all protocol parameters (see Table A.1). For the CRC signature calculation of the entire protocol block, the CRC-16 in Table D.1 shall be used. This CRC polynomial has a Hamming distance of ≥ 6 for lengths ≤ 16 octets. A seed value "0" shall be used (see D.3.6).

A.2.10 FSP_VerifyRecord

A record consisting of the authenticity and protocol parameters is transferred via the service "SMI_PortConfiguration" (see 10.2.1 and 10.3.3) and stored within the Configuration Manager of an FS-Master. At start-up during PREOPERATE, the FS-Master forwards this verification record in write only manner to a "hidden" Index in the FS-Device (see 11.7.4). The FS-Device uses this diversly handled record for verification of authenticity, protocol, I/O structure, and technology parameters. This takes place during PREOPERATE after a "DS_Change" (see Figure 35 and IEC 61131-9) whenever an FS-Device has been replaced and parameter have been restored through Data Storage mechanisms. It also takes place after Port power OFF/ON during commissioning through SMI_PortPowerOffOn (see IEC 61131-9:2022, 11.2.14 and Clause E.9).

The record shall be transferred as an entity. Subindex access is not permitted. Index 0x4202 (16898) shall be "hidden" to the user; that is, it shall not be described within the IODD.

A.2.11 FSP_TimeToReady

The FS-Device designer measures/determines the time from power-on to the appearance of the Ready pulse (see 5.3.3 and t_{2R} in Table 7) and assigns the value to FSP_TimeToReady in the IODD of the FS-Device (see E.5.8).

NOTE The value is related to the parameter "Time delay before availability" in [8] or [12].

Values greater than 5 s leads to restricted FS-Master behavior, for example no automatic Device detection.

A.2.12 FSP_MinShutDownTime

The FS-Device designer measures/determines the minimum time required to shut down after Port power off prior to a restart and assigns the value to FSP_MinShutDownTime in the IODD of the FS-Device (see Annex E.5.8).

A.2.13 FSP_WCDT

The FS-Device designer measures/determines the "Worst-Case Delay Time" values as defined in H.6. Several different values are possible since FS-Devices can be configured via technology parameters, for example high resolution may lead to longer and low resolution to shorter times. When reading this parameter, the FS-Device will provide a value corresponding to its current parameterization.

A.2.14 FSP_OFDT

The FS-Device designer measures/determines the "One Fault Delay Time" values as defined in H.6. Several different values are possible. When reading this parameter, the FS-Device will provide a value corresponding to its current parameterization.

A.2.15 FSP_ParamDescCRC

The purpose of this parameter is to secure the relevant descriptions of safety parameters within the IODD (see E.5.6) against data falsification as shown in Figure A.3.

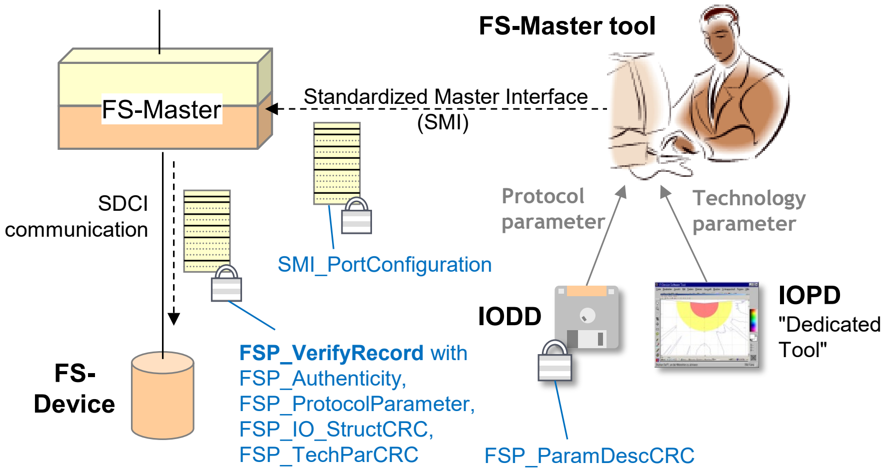


Figure A.3 – Securing of safety parameters

Annex B
(normative)

Extensions to EventCodes

B.1 Additional FS-Device EventCodes

The Safety Communication Layer (SCL) within an FS-Device can create its own EventCodes as shown in Table B.1. They are conveyed by the SMI_DeviceEvent service.

Table B.1 – FS-Device SCL specific EventCodes

EventCode	Definition and recommended maintenance action	FS-Device status value	TYPE
0xB000	Transmission error (CRC signature)	2	Notification
0xB001	Transmission error (Counter)	2	Notification
0xB002	Transmission error (Timeout)	3	Error
0xB003	Unexpected authentication code	3	Error
0xB004	Unexpected authentication Port	3	Error
0xB005	Incorrect FSP_AuthentCRC	3	Error
0xB006	Incorrect FSP_ProtParCRC	3	Error
0xB007	Incorrect FSP_TechParCRC	3	Error
0xB008	Incorrect FSP_IO_StructCRC	3	Error
0xB009	Watchdog time out of specification (e.g. "0")	3	Error
0xB00A	No FSP_VerifyRecord received (triggered after transition to OPERATE)	3	Error
0xB00B to 0xB0FF	Reserved: do not use number; do not evaluate number	–	–

Usually, "CRC signature" and/or "Counter" transmission errors are caused by seriously falsified SDCI messages with SPDUs due to heavy interferences. There is nothing to repair and an operator acknowledgment is sufficient. This very unlikely warning should inform the operator and the responsible production manager about possible changes within a machine requiring an inspection according to the safety manual (see Clause H.6).

B.2 Additional Port EventCodes

The Safety Communication Layer (SCL) within an FS-Master can create its own EventCodes as shown in Table B.2. They are conveyed by the SMI_PortEvent service (see IEC 61131-9).

Table B.2 – FS-Master SCL specific EventCodes

EventCode	Definition and recommended maintenance action	Status value	TYPE
0x2000	Transmission error (CRC signature)	2	Notification
0x2001	Transmission error (Counter)	2	Notification
0x2002	Transmission error (Timeout)	3	Error
0x2003	Unexpected authentication code	3	Error
0x2004	Reserved	–	–
0x2005	Incorrect FSP_AuthentCRC	3	Error
0x2006	Incorrect FSP_ProtParCRC	3	Error
0x2007 to 0x2008	Reserved	–	–
0x2009	Watchdog time out of specification (e.g. "0")	3	Error
0x200A to 0x20EF	Reserved: do not use number; do not evaluate number	–	–

EventCode	Definition and recommended maintenance action	Status value	TYPE
0x20F0	OSSD locked – signals beyond specified discrepancy times	3	Error
0x20F1 to 0x20FF	Reserved: do not use number; do not evaluate number	–	–

Annex C
(normative)

Extensions to Data Types

C.1 Data types for SDCI-FS

Table C.1 shows the available data types in SDCI-FS for cyclic exchange of Process Data for safety functions (see 11.4.9.2).

Table C.1 – Data types for SDCI-FS

Data type	Coding	Length	See IEC 61131-9:2022	Device example
BooleanT/bit	BooleanT ("packed form" for efficiency, no WORD structures); assignment of signal names to bits is possible.	1 bit	F.2.2; Table F.22 and Figure F.9	Proximity switch
IntegerT(16)	IntegerT (enumerated or signed)	2 octets	F.2.4; Table F.4, Table F.7, and Figure F.3	Protection fields of laser scanner
IntegerT(32)	IntegerT (enumerated or signed)	4 octets	F.2.4; Table F.4, Table F.6, and Figure F.3	Encoder or length measurement (≈ ± 2 km, resolution 1 µm)

C.2 BooleanT (bit)

A BooleanT represents a data type that can have only two different values i.e. TRUE and FALSE. The data type is specified in Table C.2.

Table C.2 – BooleanT for SDCI-FS

Data type name	Value range	Resolution	Length
BooleanT	TRUE / FALSE	-	1 bit

SDCI-FS uses solely the so-called packed form via RecordT as shown in Table C.3.

Table C.3 – Example of BooleanT within a RecordT

Subindex	Offset	Data items	Data Type	Name/symbol
1	0	TRUE	BooleanT	Proximity_1
2	1	FALSE	BooleanT	Proximity_2
3	2	FALSE	BooleanT	EmergencyStop_1
4	3	TRUE	BooleanT	EmergencyStop_2
5	4	TRUE	BooleanT	EmergencyStop_3

Figure C.1 demonstrates an example of a BooleanT data structure. Padding bits are "0".

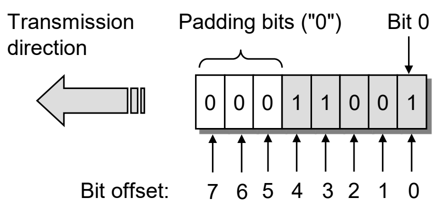


Figure C.1 – Example of a BooleanT data structure

Only RecordT data structures of 8-bit length are permitted. Longer data structures shall use multiple RecordT data structures (see Clause C.5).

NOTE Data structures longer than 8 bit can cause mapping problems with upper-level FSCP systems (see 3.5.2).

C.3 IntegerT (16)

An IntegerT(16) is representing a signed number depicted by 16 bits. The number is accommodated within the octet container 2 and right aligned and extended correctly signed to the chosen number of bits. The data type is specified in Table C.4 for singular use. SN represents the sign with "0" for all positive numbers and zero, and "1" for all negative numbers. Padding bits are filled with the content of the sign bit (SN).

Table C.4 – IntegerT(16)

Data type name	Value range	Resolution	Length
IntegerT(16)	-2^{15} to $2^{15}-1$	1	2 octets
NOTE 1 High order padding bits are filled with the value of the sign bit (SN). NOTE 2 Most significant octet (MSO) sent first (lowest respective octet number in Table C.5).			

The coding of IntegerT(16) is shown in Table C.5.

Table C.5 – IntegerT(16) coding

Bit	7	6	5	4	3	2	1	0	Container
Octet 1	SN	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2 octets
Octet 2	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

C.4 IntegerT (32)

An IntegerT(32) is representing a signed number depicted by 32 bits. The number is accommodated within the octet container 4 and right aligned and extended correctly signed to the chosen number of bits. The data type is specified in Table C.6 for singular use. SN represents the sign with "0" for all positive numbers and zero, and "1" for all negative numbers. Padding bits are filled with the content of the sign bit (SN).

Table C.6 – IntegerT(32)

Data type name	Value range	Resolution	Length
IntegerT(32)	-2^{31} to $2^{31}-1$	1	4 octets
NOTE 1 High order padding bits are filled with the value of the sign bit (SN). NOTE 2 Most significant octet (MSO) sent first (lowest respective octet number in Table C.7).			

The coding of IntegerT(32) is shown in Table C.7

Table C.7 – IntegerT(32) coding

Bit	7	6	5	4	3	2	1	0	Container
Octet 1	SN	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	4 octets
Octet 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

C.5 Safety Code

Size of the Safety Code as shown in Figure C.2 and Figure C.3 can be identified by the

- Parameter "FSP_ProtMode" (see Table A.1), and
- FS I/O structure description (see Table A.1).

Thus, the overall I/O data structure can be identified even if there are non-safety related I/O data associated with the SPDU.

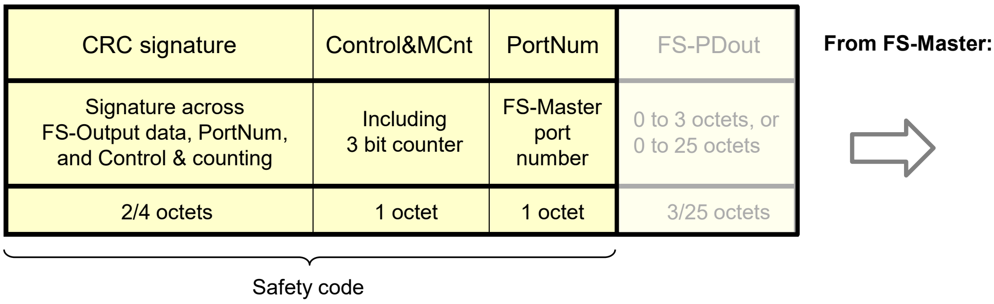


Figure C.2 – Safety Code of an output message

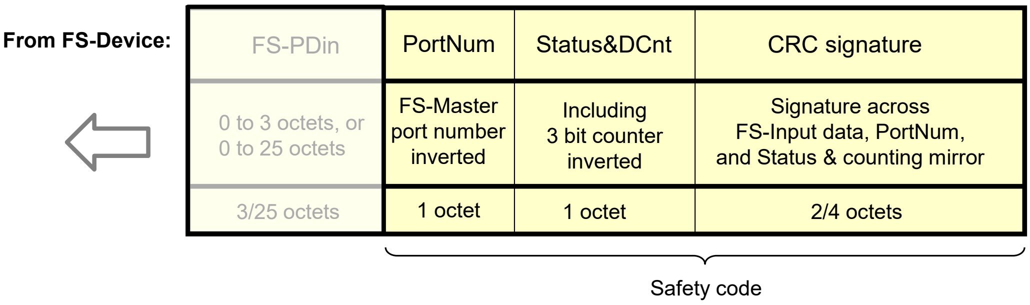


Figure C.3 – Safety Code of an input message

Annex D
(normative)

CRC generator polynomials

D.1 Overview of CRC generator polynomials

Hamming distance and properness for all required data lengths are important characteristics to select a particular generator polynomial.

If the generator polynomial g(x) = p(x)*(1 + x) is used, where p(x) is a primitive polynomial of degree (r – 1), then the maximum total block length is $2^{(r-1)}$ - 1, and the code is able to detect single, double, triple and any odd number of errors (see [13]).

If properness is approved, the residual error probability for the approved data length is 2^{-r}.

It shall be prohibited that the CRC generator polynomial used in the underlying transmission systems, for example SDCI, matches the CRC generator polynomial used for SDCI-FS.

Table D.1 shows the CRC-16 and CRC-32 generator polynomials in use for SDCI-FS:

Table D.1 – CRC generator polynomials for SDCI-FS

CRC-16/32 polynomial ("Normal" representation)	Data length (bits)	Hamming distance	Properness	Reference	Remark
0x4EAB	≤ 128	≥ 6	≤ 7 octets	[14]	Suitable for functional safety
0xF4ACFB13	≤ 32768	≥ 6	≤128 octets	[14]	
	≤ 65534	≥ 4			
NOTE Representation: "Normal": high order bit omitted					

The CRC-16 can be used

- to secure cyclic Process Data exchange with a total safety PDU length of up to 7 octets, i.e. 3 octets for safety Process Data and 4 octets of safety code, and
- to secure the transfer of up to 16 octets of FSP parameters at start-up or restart.

The CRC-32 can be used

- to secure cyclic Process Data exchange with a total safety PDU length of up to 32 octets, i.e. 25 octets for safety Process Data (see 11.4.4), and 6 octets of safety code, and
- to secure the transfer and data integrity of the entire FST parameter set.

Additional parameters and assumptions for the calculation of residual error probabilities/rates can be found in 11.4.7.

D.2 Residual error probabilities

Figure D.1 shows the results of residual error probability (REP) calculations over bit error probabilities (BEP) for safety PDU lengths from 3 to 7 octets.

The REP is less than $0,9 \times 10^{-9}$ for BEPs less than the required 10^{-2} at a length of 7 octets.

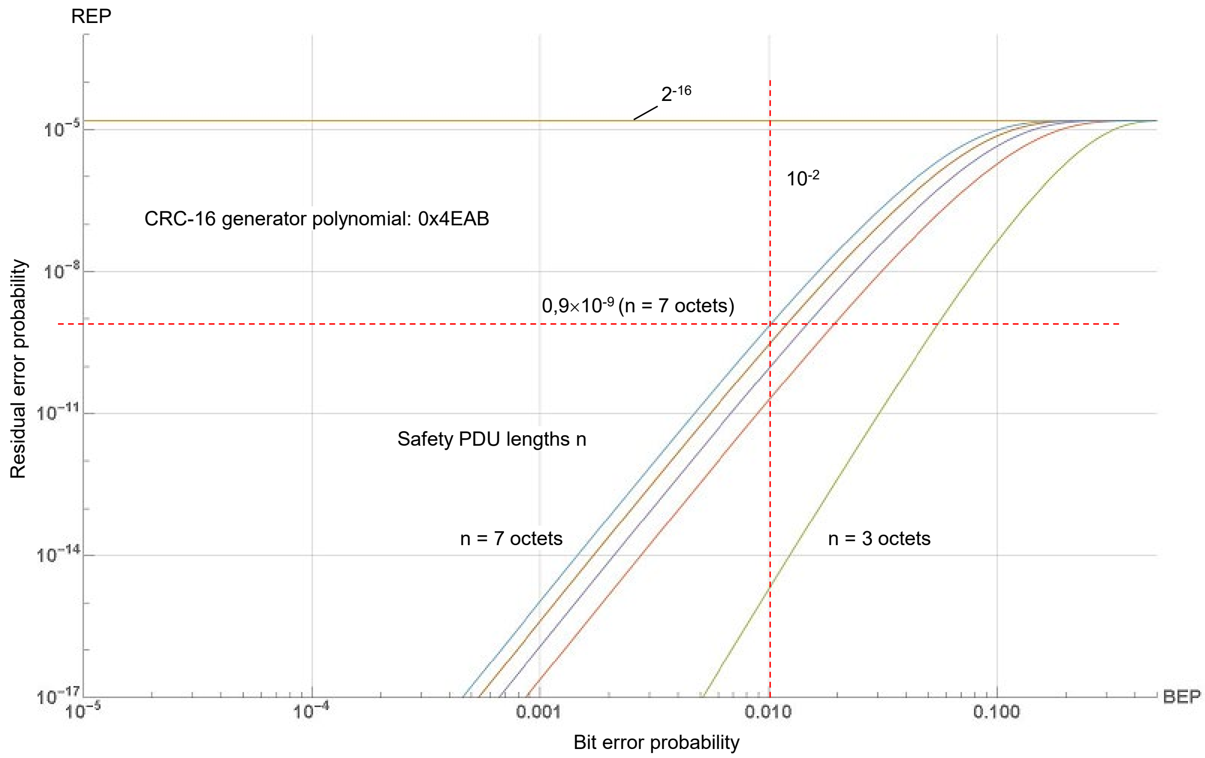


Figure D.1 – CRC-16 generator polynomial

Figure D.2 shows the results of residual error probability (REP) calculations over bit error probabilities (BEP) for safety PDU lengths from 5 to 128 octets.

The REP is less than $0{,}5\times10^{-10}$ for BEPs less than the required 10^{-2} at a length of 26 octets.

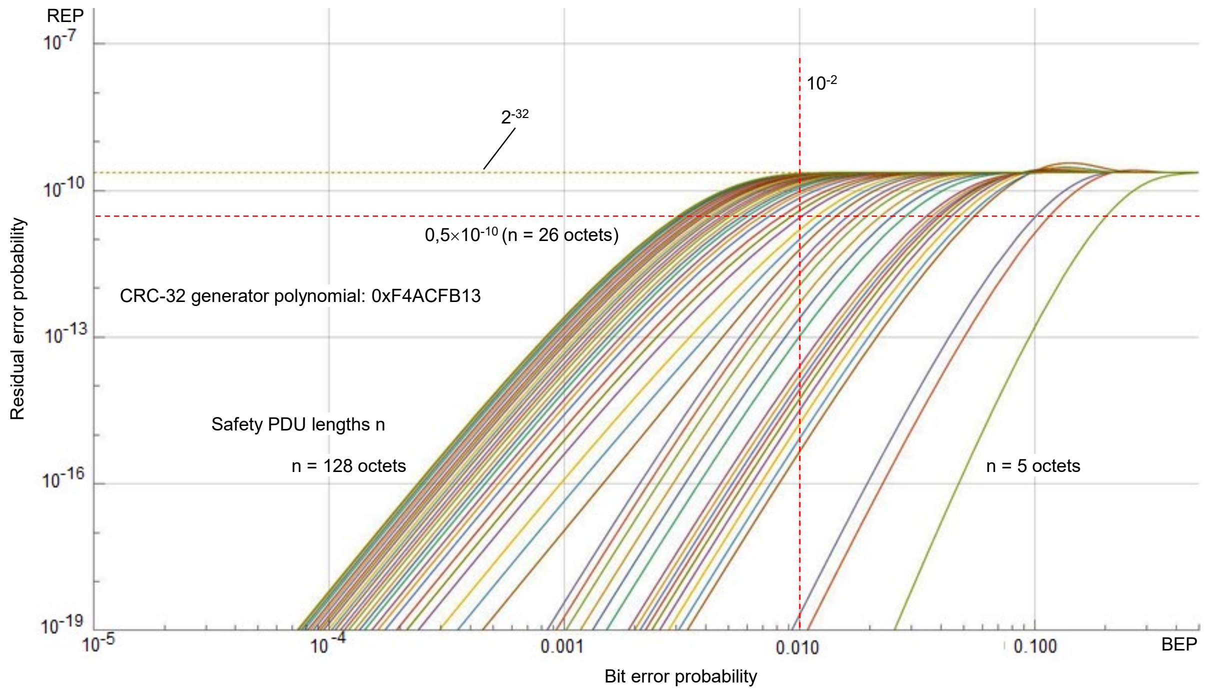


Figure D.2 – CRC-32 generator polynomial

D.3 Implementation considerations

D.3.1 Overview

The designer has two choices to implement the CRC signature calculation. One is based on an algorithm using XOR and shift operations while the other is faster using octet shifts and lookup tables.

D.3.2 Bit shift algorithm (16 bit)

For the 16-bit CRC signature, the value 0x4EAB is used as the generator polynomial. The number of data bits may be odd or even. The value generated after the last octet corresponds to the CRC signature to be transferred.

Figure D.3 shows the algorithm for the innermost loop in "C" programming language.

```
void crc16_calc(unsigned char x, unsigned long *r)
  int i;
  for (i = 1; i <= 8; i++)
    if ((bool)(*r & 0x8000) != (bool)(x & 0x80))
      /* XOR = 1 → shift and process polynomial */
      *r = (*r << 1) ^ 0x4EAB;
      else
      /* XOR = 0 → shift only */
      *r = *r << 1;
    x = x << 1;
  /* for */
```

Figure D.3 – Bit shift algorithm in "C" language (16 bit)

The variables used in Figure D.3 are specified in Table D.2.

Table D.2 – Definition of variables used in Figure D.3

Variable	Definition
x	Data bits including 16-bit CRC signature with "0"
*r	Dereferenced pointer to CRC signature
i	Bitcount 1 to 8

D.3.3 Lookup table (16 bit)

The corresponding function in "C" language is shown in Figure D.4. This function is faster. However, the lookup table requires memory space.

```
r = crctab16 [((r >> 8) ^ *q++) & 0xff] ^(r << 8)
```

Figure D.4 – CRC-16 signature calculation using a lookup table

The variables used in Figure D.4 are specified in Table D.3.

Table D.3 – Definition of variables used in Figure D.4

Variable	Definition
r	CRC signature
q	q represents the pointer to the actual octet value requiring CRC calculation. After reading the value this pointer shall be incremented for the next octet via q++.

The function in Figure D.4 uses the lookup in Table D.4.

Table D.4 – Lookup table for CRC-16 signature calculation

CRC-16 lookup table (0 to 255)							
0000	4EAB	9D56	D3FD	7407	3AAC	E951	A7FA
E80E	A6A5	7558	3BF3	9C09	D2A2	015F	4FF4
9EB7	D01C	03E1	4D4A	EAB0	A41B	77E6	394D
76B9	3812	EBEF	A544	02BE	4C15	9FE8	D143
73C5	3D6E	EE93	A038	07C2	4969	9A94	D43F
9BCB	D560	069D	4836	EFCC	A167	729A	3C31
ED72	A3D9	7024	3E8F	9975	D7DE	0423	4A88
057C	4BD7	982A	D681	717B	3FD0	EC2D	A286
E78A	A921	7ADC	3477	938D	DD26	0EDB	4070
0F84	412F	92D2	DC79	7B83	3528	E6D5	A87E
793D	3796	E46B	AAC0	0D3A	4391	906C	DEC7
9133	DF98	0C65	42CE	E534	AB9F	7862	36C9
944F	DAE4	0919	47B2	E048	AEE3	7D1E	33B5
7C41	32EA	E117	AFBC	0846	46ED	9510	DBBB
0AF8	4453	97AE	D905	7EFF	3054	E3A9	AD02
E2F6	AC5D	7FA0	310B	96F1	D85A	0BA7	450C
81BF	CF14	1CE9	5242	F5B8	BB13	68EE	2645
69B1	271A	F4E7	BA4C	1DB6	531D	80E0	CE4B
1F08	51A3	825E	CCF5	6B0F	25A4	F659	B8F2
F706	B9AD	6A50	24FB	8301	CDAA	1E57	50FC
F27A	BCD1	6F2C	2187	867D	C8D6	1B2B	5580
1A74	54DF	8722	C989	6E73	20D8	F325	BD8E
6CCD	2266	F19B	BF30	18CA	5661	859C	CB37
84C3	CA68	1995	573E	F0C4	BE6F	6D92	2339
6635	289E	FB63	B5C8	1232	5C99	8F64	C1CF
8E3B	C090	136D	5DC6	FA3C	B497	676A	29C1
F882	B629	65D4	2B7F	8C85	C22E	11D3	5F78
108C	5E27	8DDA	C371	648B	2A20	F9DD	B776
15F0	5B5B	88A6	C60D	61F7	2F5C	FCA1	B20A
FDFE	B355	60A8	2E03	89F9	C752	14AF	5A04
8B47	C5EC	1611	58BA	FF40	B1EB	6216	2CBD
6349	2DE2	FE1F	B0B4	174E	59E5	8A18	C4B3

NOTE This table contains 16-bit values in hexadecimal representation for each value (0 to 255) of the argument a in the function crctab16 [a]. The table should be used in ascending order from top left (0) to bottom right (255).

D.3.4 Bit shift algorithm (32 bit)

For the 32-bit CRC signature, the value 0xF4ACFB13 is used as the generator polynomial. The number of data bits may be odd or even. The value generated after the last octet corresponds to the CRC signature to be transferred.

Figure D.5 shows the algorithm for the innermost loop in "C" programming language.

```
void crc32_calc(unsigned char x, unsigned long *r)
  int i;
  for (i = 1; i <= 8; i++)
    if ((bool)(*r & 0x80000000) != (bool)(x & 0x80))
      /* XOR = 1 → shift and process polynomial */
      *r = (*r << 1) ^ 0xF4ACFB13;
      else
      /* XOR = 0 → shift only */
      *r = *r << 1;
    x = x << 1;
  /* for */
```

Figure D.5 – Bit shift algorithm in "C" language (32 bit)

The variables used in Figure D.5 are specified in Table D.5.

Table D.5 – Definition of variables used in Figure D.5

Variable	Definition
x	Data bits including 32-bit CRC signature with "0"
*r	Dereferenced pointer to CRC signature
i	Bit count 1 to 8

D.3.5 Lookup table (32 bit)

The corresponding function in "C" language is shown in Figure D.6. This function is faster. However, the lookup table requires memory space.

```
r = crctab32 [((r >> 24) ^ *q++) & 0xff] ^(r << 8)
```

Figure D.6 – CRC-32 signature calculation using a lookup table

The variables used in Figure D.6 are specified in Table D.6.

Table D.6 – Definition of variables used in Figure D.4

Variable	Definition
r	CRC signature
q	q represents the pointer to the actual octet value requiring CRC calculation. After reading the value this pointer shall be incremented for the next octet via q++.

The function in Figure D.6 uses the lookup table in Table D.7.

Table D.7 – Lookup table for CRC-32 signature calculation

CRC-32 lookup table (0 to 255)							
00000000	F4ACFB13	1DF50D35	E959F626	3BEA1A6A	CF46E179	261F175F	D2B3EC4C
77D434D4	8378CFC7	6A2139E1	9E8DC2F2	4C3E2EBE	B892D5AD	51CB238B	A567D898
EFA869A8	1B0492BB	F25D649D	06F19F8E	D44273C2	20EE88D1	C9B77EF7	3D1B85E4
987C5D7C	6CD0A66F	85895049	7125AB5A	A3964716	573ABC05	BE634A23	4ACFB130
2BFC2843	DF50D350	36092576	C2A5DE65	10163229	E4BAC93A	0DE33F1C	F94FC40F
5C281C97	A884E784	41DD11A2	B571EAB1	67C206FD	936EFDEE	7A370BC8	8E9BF0DB
C45441EB	30F8BAF8	D9A14CDE	2D0DB7CD	FFBE5B81	0B12A092	E24B56B4	16E7ADA7

CRC-32 lookup table (0 to 255)							
B380753F	472C8E2C	AE75780A	5AD98319	886A6F55	7CC69446	959F6260	61339973
57F85086	A354AB95	4A0D5DB3	BEA1A6A0	6C124AEC	98BEB1FF	71E747D9	854BBCCA
202C6452	D4809F41	3DD96967	C9759274	1BC67E38	EF6A852B	0633730D	F29F881E
B850392E	4CFCC23D	A5A5341B	5109CF08	83BA2344	7716D857	9E4F2E71	6AE3D562
CF840DFA	3B28F6E9	D27100CF	26DDFBDC	F46E1790	00C2EC83	E99B1AA5	1D37E1B6
7C0478C5	88A883D6	61F175F0	955D8EE3	47EE62AF	B34299BC	5A1B6F9A	AEB79489
0BD04C11	FF7CB702	16254124	E289BA37	303A567B	C496AD68	2DCF5B4E	D963A05D
93AC116D	6700EA7E	8E591C58	7AF5E74B	A8460B07	5CEAF014	B5B30632	411FFD21
E47825B9	10D4DEAA	F98D288C	0D21D39F	DF923FD3	2B3EC4C0	C26732E6	36CBC9F5
AFF0A10C	5B5C5A1F	B205AC39	46A9572A	941ABB66	60B64075	89EFB653	7D434D40
D82495D8	2C886ECB	C5D198ED	317D63FE	E3CE8FB2	176274A1	FE3B8287	0A977994
4058C8A4	B4F433B7	5DADC591	A9013E82	7BB2D2CE	8F1E29DD	6647DFFB	92EB24E8
378CFC70	C3200763	2A79F145	DED50A56	0C66E61A	F8CA1D09	1193EB2F	E53F103C
840C894F	70A0725C	99F9847A	6D557F69	BFE69325	4B4A6836	A2139E10	56BF6503
F3D8BD9B	07744688	EE2DB0AE	1A814BBD	C832A7F1	3C9E5CE2	D5C7AAC4	216B51D7
6BA4E0E7	9F081BF4	7651EDD2	82FD16C1	504EFA8D	A4E2019E	4DBBF7B8	B9170CAB
1C70D433	E8DC2F20	0185D906	F5292215	279ACE59	D336354A	3A6FC36C	CEC3387F
F808F18A	0CA40A99	E5FDFCBF	115107AC	C3E2EBE0	374E10F3	DE17E6D5	2ABB1DC6
8FDCC55E	7B703E4D	9229C86B	66853378	B436DF34	409A2427	A9C3D201	5D6F2912
17A09822	E30C6331	0A559517	FEF96E04	2C4A8248	D8E6795B	31BF8F7D	C513746E
6074ACF6	94D857E5	7D81A1C3	892D5AD0	5B9EB69C	AF324D8F	466BBBA9	B2C740BA
D3F4D9C9	275822DA	CE01D4FC	3AAD2FEF	E81EC3A3	1CB238B0	F5EBCE96	01473585
A420ED1D	508C160E	B9D5E028	4D791B3B	9FCAF777	6B660C64	823FFA42	76930151
3C5CB061	C8F04B72	21A9BD54	D5054647	07B6AA0B	F31A5118	1A43A73E	EEEF5C2D
4B8884B5	BF247FA6	567D8980	A2D17293	70629EDF	84CE65CC	6D9793EA	993B68F9
NOTE This table contains 32-bit values in hexadecimal representation for each value (0 to 255) of the argument a in the function crctab32 [a]. The table should be used in ascending order from top left (0) to bottom right (255).							

D.3.6 Seed values

The algorithm for example in Figure D.3 does not mention explicitly any initial value for the CRC signature variable in "*r". It is implicitly assumed to be "0" by default. This initial value is sometimes called "seed value" in literature.

In 11.4.7 a seed value of "1" is required for the cyclic data exchange of safety PDUs. The reason for that is the possibility for the FS-PDout or FS-PDin data to become completely "0". Since it is a property of CRC-signatures for leading zeros in data strings not to be secured by CRC signatures whenever the seed value is "0", the requirement in 11.4.6 is justified. Any value instead of "0" could be used. However, a "1" is sufficient and faster since all the operations then are shifting and only the last one consists of shifting and XOR processing.

In A.2.3, A.2.9, A.2.7, A.2.11, and E.5.1, the seed value can be "0" since there are no leading zeros within the data strings.

Publicly available CRC signature calculators can be found in [15].

D.3.7 Octet order for CRC calculation

The order of octets for the CRC calculation of the SPDU shall start with the seed value followed by all other octets in the transmission order (see 3.5.2, Figure C.2, and Figure C.3).

Annex E
(normative)

IODD extensions

E.1 General

The IODD of FS-Devices requires extensions for particular FSP parameters and a securing mechanism to protect the content of IODD files from being falsified as mentioned in 11.7.1.

In addition, some of the parameters specified in IEC 61131-9 shall be mandatory instead of optional for this SDCI extension/profile (see E.3).

E.2 Schema

There are no extensions required to the existing IODD schema.

NOTE The IODD schema is outside the scope of IEC 61131-9. It is described in [16].

E.3 IODD constraints

E.3.1 General rules

Basis for the IODD in SDCI-FS are the definitions of the Common Profile in [17].

All parameters refer to IEC 61131-9. As a general rule, all parameters with Read/Write (R/W) access shall provide a default value within the IODD (for FSP parameters see E.5.2).

E.3.2 Description of the IODD structure

The structure of the IODD of an FS-Device is defined within the XML snippet file "IOL-SafetyProfile-Snippets1.1.3.xml". The following XML code shows the content of this file:

```
<?xml version="1.0" encoding="UTF-8"?>
<IODDProfileDefinitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.io-link.com/IODD/2010/10">
  <DocumentInfo version="V1.1.3" releaseDate="2022-01-21" copyright="Copyright 2022, IO-Link
Community"/>
  <SupportedProfiles profileCharacteristic="16385" profileClassName="Safety"
profilePrefixes="FSP, FST" reservedIndexRange="16896..16898, 16912, 16914..16916"
reservedEvents="45056..45066" requiredProfile="16384">
    <ProfileVariant id="PR_FSP" profileId="16385" name="IO-Link Safety"/>
  </SupportedProfiles>
  <!--
          IODD snippet content for the IO-Link Safety Profile Extension.
          The IODD Checker will verify the IODD according to these rules.
    -->
  <DatatypeCollection>
    <Datatype id="D_FSP_Authenticity" xsi:type="RecordT" bitLength="88"
subindexAccessSupported="false">
      <RecordItem subindex="1" bitOffset="56">
        <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
        <Name textId="TN_V_FSP_Authenticity_1"/>
        <Description textId="TD_V_FSP_Authenticity_1"/>
      </RecordItem>
      <RecordItem subindex="2" bitOffset="24">
        <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
        <Name textId="TN_V_FSP_Authenticity_2"/>
        <Description textId="TD_V_FSP_Authenticity_2"/>
      </RecordItem>
      <RecordItem subindex="3" bitOffset="16">
        <SimpleDatatype xsi:type="UIntegerT" bitLength="8"/>
        <Name textId="TN_V_FSP_Port"/>
        <Description textId="TD_V_FSP_Port"/>
      </RecordItem>
      <RecordItem subindex="4" bitOffset="0">
        <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
        <Name textId="TN_V_FSP_AuthentCRC"/>
        <Description textId="TD_V_FSP_AuthentCRC"/>
```

```
        </RecordItem>
      </Datatype>
      <Datatype id="D_FSP_Protocol" xsi:type="RecordT" bitLength="96"
subindexAccessSupported="false">
        <RecordItem subindex="1" bitOffset="88">
          <SimpleDatatype xsi:type="UIntegerT" bitLength="8">
            <SingleValue value="1"/>
          </SimpleDatatype>
          <Name textId="TN_V_FSP_ProtVersion"/>
          <Description textId="TD_V_FSP_ProtVersion"/>
        </RecordItem>
        <RecordItem subindex="2" bitOffset="80">
          <SimpleDatatype xsi:type="UIntegerT" bitLength="8">
            <!-- Which of these SingleValues is supported is device specific. Only one shall be
referenced. -->
            <SingleValue value="1" checkElement="minOccurs 0">
              <Name textId="TN_SV_FSP_ProtMode_crc16"/>
            </SingleValue>
            <SingleValue value="2" checkElement="minOccurs 0">
              <Name textId="TN_SV_FSP_ProtMode_crc32"/>
            </SingleValue>
          </SimpleDatatype>
          <Name textId="TN_V_FSP_ProtMode"/>
          <Description textId="TD_V_FSP_ProtMode"/>
        </RecordItem>
        <RecordItem subindex="3" bitOffset="64">
          <SimpleDatatype xsi:type="UIntegerT" bitLength="16">
            <!-- Which ValueRange is supported is device specific (but the lowerValue must be >0)
-->
            <ValueRange lowerValue="#tbd 1..65535 #" upperValue="#tbd 2..65535 #"/>
          </SimpleDatatype>
          <Name textId="TN_V_FSP_Watchdog"/>
          <Description textId="TD_V_FSP_Watchdog"/>
        </RecordItem>
        <RecordItem subindex="4" bitOffset="48">
          <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
          <Name textId="TN_V_FSP_IO_StructCRC"/>
          <Description textId="TD_V_FSP_IO_StructCRC"/>
        </RecordItem>
        <RecordItem subindex="5" bitOffset="16">
          <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
          <Name textId="TN_V_FSP_TechParCRC"/>
          <Description textId="TD_V_FSP_TechParCRC"/>
        </RecordItem>
        <RecordItem subindex="6" bitOffset="0">
          <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
          <Name textId="TN_V_FSP_ProtParCRC"/>
          <Description textId="TD_V_FSP_ProtParCRC"/>
        </RecordItem>
      </Datatype>
      <Datatype id="D_FSP_PDin" xsi:type="RecordT" bitLength="#tbd#"
subindexAccessSupported="false">
        <!-- Technology specific safety process data have subindex 1..126 -->
        <!-- Datatype order for subindices is: boolean, integer16, integer32 -->
        <!-- Boolean array -->
        <!-- There may be no gaps between the booleans, but the last octet may contain less than
eight booleans. -->
        <RecordItem subindex="#tbd 1..126 #" bitOffset="#tbd#" checkElement="minOccurs 0">
          <SimpleDatatype xsi:type="BooleanT">
            <SingleValue value="false" checkElement="minOccurs 0">
              <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
            </SingleValue>
            <SingleValue value="true" checkElement="minOccurs 0">
              <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
            </SingleValue>
          </SimpleDatatype>
          <Name textId="TN_V_FST_" checkAttributes="startsWith textId"/>
          <Description textId="TD_V_FST_" checkAttributes="startsWith textId"/>
        </RecordItem>
        <!-- Integer16 values -->
        <!-- Integer values always are aligned to byte borders. -->
        <RecordItem subindex="#tbd 1..126 #" bitOffset="#tbd#" checkElement="minOccurs 0">
          <SimpleDatatype xsi:type="IntegerT" bitLength="16">
            <SingleValue value="#tbd#" checkElement="minOccurs 0">
              <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
            </SingleValue>
            <ValueRange lowerValue="#tbd#" upperValue="#tbd#" checkElement="minOccurs 0"/>
          </SimpleDatatype>
```

```xml
        <Name textId="TN_V_FST_" checkAttributes="startsWith textId"/>
        <Description textId="TD_V_FST_" checkAttributes="startsWith textId"/>
      </RecordItem>
      <!-- Integer32 values -->
      <!-- Integer values always are aligned to byte borders. -->
      <RecordItem subindex="#tbd 1..126 #" bitOffset="#tbd#" checkElement="minOccurs 0">
        <SimpleDatatype xsi:type="IntegerT" bitLength="32">
          <SingleValue value="#tbd#" checkElement="minOccurs 0">
            <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
          </SingleValue>
          <ValueRange lowerValue="#tbd#" upperValue="#tbd#" checkElement="minOccurs 0"/>
        </SimpleDatatype>
        <Name textId="TN_V_FST_" checkAttributes="startsWith textId"/>
        <Description textId="TD_V_FST_" checkAttributes="startsWith textId"/>
      </RecordItem>
      <!-- Safety Code has fixed subindex 127 -->
      <RecordItem subindex="127" bitOffset="#tbd#">
        <SimpleDatatype xsi:type="OctetStringT" fixedLength="#tbd 4, 6 #"/>
        <Name textId="TN_V_FSP_SafetyCode"/>
        <Description textId="TD_V_FSP_SafetyCode"/>
      </RecordItem>
      <!-- Non-safety process data has subindex 128..255 -->
      <RecordItem subindex="#tbd 128..255 #" bitOffset="#tbd#" checkElement="minOccurs 0">
        <DatatypeRef datatypeId="#tbd#" checkElement="minOccurs 0"/>
        <SimpleDatatype xsi:type="tbd" checkElement="minOccurs 0" checkAttributes="option
bitLength, option fixedLength">
          <SingleValue value="false" checkElement="minOccurs 0">
            <Name textId="TN_SV_" checkAttributes="startsWith textId"/>
          </SingleValue>
          <ValueRange lowerValue="#tbd#" upperValue="#tbd#" checkElement="minOccurs 0"/>
        </SimpleDatatype>
        <Name textId="#tbd#"/>
        <Description textId="#tbd#"/>
      </RecordItem>
    </Datatype>
    <Datatype id="D_FSP_PDout" xsi:type="RecordT" bitLength="#tbd#"
subindexAccessSupported="false">
      <!-- Technology specific safety process data have subindex 1..126 -->
      <!-- Datatype order for subindices is: boolean, integer16, integer32 -->
      <!-- Boolean array  -->
      <!-- There may be no gaps between the booleans, but the last octet may contain less than
eight booleans. -->
      <RecordItem subindex="#tbd 1..126 #" bitOffset="#tbd#" checkElement="minOccurs 0">
        <SimpleDatatype xsi:type="BooleanT">
          <SingleValue value="false" checkElement="minOccurs 0">
            <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
          </SingleValue>
          <SingleValue value="true" checkElement="minOccurs 0">
            <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
          </SingleValue>
        </SimpleDatatype>
        <Name textId="TN_V_FST_" checkAttributes="startsWith textId"/>
        <Description textId="TD_V_FST_" checkAttributes="startsWith textId"/>
      </RecordItem>
      <!-- Integer16 values -->
      <!-- Imteger values always are aligned to byte borders. -->
      <RecordItem subindex="#tbd 1..126 #" bitOffset="#tbd#" checkElement="minOccurs 0">
        <SimpleDatatype xsi:type="IntegerT" bitLength="16">
          <SingleValue value="#tbd#" checkElement="minOccurs 0">
            <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
          </SingleValue>
          <ValueRange lowerValue="#tbd#" upperValue="#tbd#" checkElement="minOccurs 0"/>
        </SimpleDatatype>
        <Name textId="TN_V_FST_" checkAttributes="startsWith textId"/>
        <Description textId="TD_V_FST_" checkAttributes="startsWith textId"/>
      </RecordItem>
      <!-- Integer32 values -->
      <!-- Imteger values always are aligned to byte borders. -->
      <RecordItem subindex="#tbd 1..126 #" bitOffset="#tbd#" checkElement="minOccurs 0">
        <SimpleDatatype xsi:type="IntegerT" bitLength="32">
          <SingleValue value="#tbd#" checkElement="minOccurs 0">
            <Name textId="TN_SV_FST_" checkAttributes="startsWith textId"/>
          </SingleValue>
          <ValueRange lowerValue="#tbd#" upperValue="#tbd#" checkElement="minOccurs 0"/>
        </SimpleDatatype>
        <Name textId="TN_V_FST_" checkAttributes="startsWith textId"/>
        <Description textId="TD_V_FST_" checkAttributes="startsWith textId"/>
      </RecordItem>
```

```
        <!-- Safety Code has fixed subindex 127 -->
        <RecordItem subindex="127" bitOffset="#tbd#">
          <SimpleDatatype xsi:type="OctetStringT" fixedLength="6"/>
          <Name textId="TN_V_FSP_SafetyCode"/>
          <Description textId="TD_V_FSP_SafetyCode"/>
        </RecordItem>
        <!-- Non-safety process data has subindex 128..255 -->
        <RecordItem subindex="#tbd 128..255 #" bitOffset="#tbd#" checkElement="minOccurs 0">
          <DatatypeRef datatypeId="#tbd#" checkElement="minOccurs 0"/>
          <SimpleDatatype xsi:type="tbd" checkElement="minOccurs 0" checkAttributes="option
bitLength, option fixedLength">
            <SingleValue value="false" checkElement="minOccurs 0">
              <Name textId="TN_SV_" checkAttributes="startsWith textId"/>
            </SingleValue>
            <ValueRange lowerValue="#tbd#" upperValue="#tbd#" checkElement="minOccurs 0"/>
          </SimpleDatatype>
          <Name textId="#tbd#"/>
          <Description textId="#tbd#"/>
        </RecordItem>
      </Datatype>
    </DatatypeCollection>
    <VariableCollection checkElement="atLeast">
      <StdVariableRef id="V_SystemCommand" checkElement="minOccurs 1">
        <StdSingleValueRef value="128" checkElement="maxOccurs 0"/>
        <StdSingleValueRef value="130" checkElement="maxOccurs 0"/>
      </StdVariableRef>
      <StdVariableRef id="V_ProcessDataInput" checkElement="minOccurs 1"/>
      <StdVariableRef id="V_ProcessDataOutput" checkElement="minOccurs 1"/>
      <Variable index="16896" id="V_FSP_Authenticity" accessRights="rw"
profileConstraints="PR_FSP">
        <DatatypeRef datatypeId="D_FSP_Authenticity"/>
        <RecordItemInfo subindex="1" defaultValue="0"/>
        <RecordItemInfo subindex="2" defaultValue="0"/>
        <RecordItemInfo subindex="3" defaultValue="0"/>
        <RecordItemInfo subindex="4" defaultValue="0"/>
        <Name textId="TN_V_FSP_Authenticity"/>
        <Description textId="TD_V_FSP_Authenticity"/>
      </Variable>
      <Variable index="16897" id="V_FSP_Protocol" accessRights="rw" profileConstraints="PR_FSP">
        <DatatypeRef datatypeId="D_FSP_Protocol"/>
        <!-- FSP_ProtVersion: 1= valid -->
        <RecordItemInfo subindex="1" defaultValue="1"/>
        <!-- FSP_ProtMode: 1  (16 bit CRC) or 2 (32 bit CRC)= valid -->
        <RecordItemInfo subindex="2" defaultValue="#tbd 1 2 #"/>
        <!-- FSP_Watchdog: 1 .. 65535 = valid -->
        <RecordItemInfo subindex="3" defaultValue="#tbd 1..65535 #"/>
        <!-- FSP_IO_StructCRC: = valid -->
        <RecordItemInfo subindex="4" defaultValue="#tbd 1..65535 #"/>
        <!-- FSP_TechParCRC: 0= invalid -->
        <RecordItemInfo subindex="5" defaultValue="0"/>
        <!-- FSP_ProtParCRC: 0= invalid -->
        <RecordItemInfo subindex="6" defaultValue="0"/>
        <Name textId="TN_V_FSP_Protocol"/>
        <Description textId="TD_V_FSP_Protocol"/>
      </Variable>
      <!-- Note: Variable FSP_VerifyRecord (index="16898") shall not be described in the IODD. -->
      <Variable id="V_FSP_TimeToReady" index="16912" accessRights="ro" defaultValue="#tbd 1..65535
#" profileConstraints="PR_FSP">
        <!-- resolution 10 ms -->
        <Datatype xsi:type="UIntegerT" bitLength="16"/>
        <Name textId="TN_V_FSP_TimeToReady"/>
        <Description textId="TD_V_FSP_TimeToReady"/>
      </Variable>
      <Variable id="V_FSP_MinShutDownTime" index="16913" accessRights="ro" defaultValue="#tbd
100..1000 #" profileConstraints="PR_FSP">
        <!-- resolution 10 ms -->
        <Datatype xsi:type="UIntegerT" bitLength="16"/>
        <Name textId="TN_V_FSP_MinShutDownTime"/>
        <Description textId="TD_V_FSP_MinShutDownTime"/>
      </Variable>
      <Variable id="V_FSP_ParamDescCRC" index="16914" accessRights="ro" defaultValue="#tbd#"
profileConstraints="PR_FSP">
        <Datatype xsi:type="UIntegerT" bitLength="32"/>
        <Name textId="TN_V_FSP_ParamDescCRC"/>
        <Description textId="TD_V_FSP_ParamDescCRC"/>
      </Variable>
      <Variable id="V_FSP_WCDT" index="16915" accessRights="ro" defaultValue="#tbd 1..32767 #"
profileConstraints="PR_FSP">
```

```xml
      <!-- in ms -->
      <Datatype xsi:type="UIntegerT" bitLength="16"/>
      <Name textId="TN_V_FSP_WCDT"/>
      <Description textId="TD_V_FSP_WCDT"/>
    </Variable>
    <Variable id="V_FSP_OFDT" index="16916" accessRights="ro" defaultValue="#tbd 1..32767 #" profileConstraints="PR_FSP">
      <!-- in ms -->
      <Datatype xsi:type="UIntegerT" bitLength="16"/>
      <Name textId="TN_V_FSP_OFDT"/>
      <Description textId="TD_V_FSP_OFDT"/>
    </Variable>
  </VariableCollection>
  <ProcessDataCollection>
    <ProcessData id="P_" checkElement="minOccurs 1" checkAttributes="startsWith id">
      <Condition variableId="#tbd#" subindex="#tbd#" value="#tbd#" checkElement="minOccurs 0" checkAttributes="option subindex"/>
      <ProcessDataIn id="PI_PDin" bitLength="#tbd 32..256 #" checkAttributes="startsWith id">
        <DatatypeRef datatypeId="D_FSP_PDin" checkAttributes="startsWith datatypeId"/>
        <Name textId="TN_PI_PDin" checkAttributes="startsWith textId"/>
      </ProcessDataIn>
      <ProcessDataOut id="PO_PDout" bitLength="#tbd 32..256 #" checkAttributes="startsWith id">
        <DatatypeRef datatypeId="D_FSP_PDout" checkAttributes="startsWith datatypeId"/>
        <Name textId="TN_PO_PDout" checkAttributes="startsWith textId"/>
      </ProcessDataOut>
    </ProcessData>
  </ProcessDataCollection>
  <EventCollection checkElement="atLeast">
    <Event code="45056" type="Warning">
      <Name textId="TN_EV_FSP_TransmissionError_CRCSignature"/>
      <Description textId="TD_EV_FSP_TransmissionError_CRCSignature"/>
    </Event>
    <Event code="45057" type="Warning">
      <Name textId="TN_EV_FSP_TransmissionError_Counter"/>
      <Description textId="TD_EV_FSP_TransmissionError_Counter"/>
    </Event>
    <Event code="45058" type="Error">
      <Name textId="TN_EV_FSP_TransmissionError_Timeout"/>
      <Description textId="TD_EV_FSP_TransmissionError_Timeout"/>
    </Event>
    <Event code="45059" type="Error">
      <Name textId="TN_EV_FSP_UnexpectedAuthenticationCode"/>
      <Description textId="TD_EV_FSP_UnexpectedAuthenticationCode"/>
    </Event>
    <Event code="45060" type="Error">
      <Name textId="TN_EV_FSP_UnexpectedAuthenticationPort"/>
      <Description textId="TD_EV_FSP_UnexpectedAuthenticationPort"/>
    </Event>
    <Event code="45061" type="Error">
      <Name textId="TN_EV_FSP_Incorrect_AuthentCRC"/>
      <Description textId="TD_EV_FSP_Incorrect_AuthentCRC"/>
    </Event>
    <Event code="45062" type="Error">
      <Name textId="TN_EV_FSP_Incorrect_ProtParCRC"/>
      <Description textId="TD_EV_FSP_Incorrect_ProtParCRC"/>
    </Event>
    <Event code="45063" type="Error">
      <Name textId="TN_EV_FSP_Incorrect_TechParCRC"/>
      <Description textId="TD_EV_FSP_Incorrect_TechParCRC"/>
    </Event>
    <Event code="45064" type="Error">
      <Name textId="TN_EV_FSP_Incorrect_IO_StructCRC"/>
      <Description textId="TD_EV_FSP_Incorrect_IO_StructCRC"/>
    </Event>
    <Event code="45065" type="Error">
      <Name textId="TN_EV_FSP_WatchdogTimeOutOfSpec"/>
      <Description textId="TD_EV_FSP_WatchdogTimeOutOfSpec"/>
    </Event>
    <Event code="45066" type="Error">
      <Name textId="TN_EV_FSP_NoFSVerifyRecord"/>
      <Description textId="TD_EV_FSP_NoFSVerifyRecord"/>
    </Event>
  </EventCollection>
  <UserInterface checkElement="atLeast">
    <ProcessDataRefCollection checkElement="atLeast">
      <ProcessDataRef processDataId="PI_PDin" checkElement="atLeastSequence" checkAttributes="startsWith processDataId">
        <!-- Space for technology specific functional safety PD Input Data-->
```

```
        <ProcessDataRecordItemInfo subindex="#tbd 1..127 #" checkElement="minOccurs 0"
checkAttributes="option displayFormat, option offset, option gradient, option unitCode"/>
        <!-- Safety Code -->
        <ProcessDataRecordItemInfo subindex="127" displayFormat="Hex"/>
        <!-- Space for technology specific non-safety PD Input Data-->
        <ProcessDataRecordItemInfo subindex="#tbd 128..255 #" checkElement="minOccurs 0"
checkAttributes="option displayFormat, option offset, option gradient, option unitCode"/>
      </ProcessDataRef>
      <ProcessDataRef processDataId="PO_PDout" checkElement="atLeastSequence"
checkAttributes="startsWith processDataId">
        <!-- Space for technology specific functional safety PD Input Data-->
        <ProcessDataRecordItemInfo subindex="#tbd 1..127 #" checkElement="minOccurs 0"
checkAttributes="option displayFormat, option offset, option gradient, option unitCode"/>
        <!-- Safety Code -->
        <ProcessDataRecordItemInfo subindex="127" displayFormat="Hex"/>
        <!-- Space for technology specific non-safety PD Input Data-->
        <ProcessDataRecordItemInfo subindex="#tbd 128..255 #" checkElement="minOccurs 0"
checkAttributes="option displayFormat, option offset, option gradient, option unitCode"/>
      </ProcessDataRef>
    </ProcessDataRefCollection>
    <MenuCollection checkElement="atLeast">
      <Menu id="M_OR_Param" checkElement="atLeastSequence">
        <MenuRef menuId="M_OR_Param_" checkElement="minOccurs 0" checkAttributes="startsWith
menuId"/>
        <MenuRef menuId="M_OR_FST_Param" checkElement="minOccurs 0"/>
        <MenuRef menuId="M_OR_FSP_Param"/>
        <MenuRef menuId="M_OMSR_FSP_Param_Aux"/>
      </Menu>
      <Menu id="M_MSR_Param" checkElement="atLeastSequence">
        <MenuRef menuId="M_MSR_Param_" checkElement="minOccurs 0" checkAttributes="startsWith
menuId"/>
        <MenuRef menuId="M_MSR_FST_Param" checkElement="minOccurs 0"/>
        <MenuRef menuId="M_MSR_FSP_Param"/>
        <MenuRef menuId="M_OMSR_FSP_Param_Aux"/>
      </Menu>
      <Menu id="M_OR_FST_Param" contextConstraints="ParameterMenu" checkElement="minOccurs 0">
        <Name textId="TN_M_FST_Param"/>
        <MenuRef menuId="M_OR_FST_Param" checkElement="minOccurs 0" checkAttributes="startsWith
menuId"/>
        <VariableRef variableId="V_FST_" accessRightRestriction="ro" checkElement="minOccurs 0"
checkAttributes="startsWith variableId"/>
        <RecordItemRef variableId="V_FST_" subindex="#tbd#" accessRightRestriction="ro"
checkElement="minOccurs 0" checkAttributes="startsWith variableId"/>
      </Menu>
      <Menu id="M_MSR_FST_Param" contextConstraints="ParameterMenu" checkElement="minOccurs 0">
        <Name textId="TN_M_FST_Param"/>
        <MenuRef menuId="M_MSR_FST_Param" checkElement="minOccurs 0" checkAttributes="startsWith
menuId"/>
        <VariableRef variableId="V_FST_" checkElement="minOccurs 0" checkAttributes="startsWith
variableId, option accessRightRestriction"/>
        <RecordItemRef variableId="V_FST_" subindex="#tbd#" checkElement="minOccurs 0"
checkAttributes="startsWith variableId, option accessRightRestriction"/>
      </Menu>
      <Menu id="M_OR_FSP_Param" contextConstraints="ParameterMenu"
checkElement="atLeastSequence">
        <Name textId="TN_M_FSP_Param"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="1"
accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="2"
accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="3"
accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="4"
accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="1" accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="2" accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="3" accessRightRestriction="ro"
displayFormat="Dec.0" offset="0.0" gradient="1.0" unitCode="1056"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="4" accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="5" accessRightRestriction="ro"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="6" accessRightRestriction="ro"/>
      </Menu>
      <Menu id="M_MSR_FSP_Param" contextConstraints="ParameterMenu">
        <Name textId="TN_M_FSP_Param"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="1"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="2"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="3"/>
        <RecordItemRef variableId="V_FSP_Authenticity" subindex="4"/>
```

```xml
        <RecordItemRef variableId="V_FSP_Protocol" subindex="1"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="2"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="3" displayFormat="Dec.0" offset="0.0" gradient="1.0" unitCode="1056"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="4"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="5"/>
        <RecordItemRef variableId="V_FSP_Protocol" subindex="6"/>
      </Menu>
      <Menu id="M_OMSR_FSP_Param_Aux" contextConstraints="ParameterMenu">
        <Name textId="TN_M_FSP_Param_Aux"/>
        <VariableRef variableId="V_FSP_TimeToReady" displayFormat="Dec.0" offset="0.0" gradient="10.0" unitCode="1056"/>
        <VariableRef variableId="V_FSP_MinShutDownTime" displayFormat="Dec.0" offset="0.0" gradient="10.0" unitCode="1056"/>
        <VariableRef variableId="V_FSP_WCDT" displayFormat="Dec.0" offset="0.0" gradient="1.0" unitCode="1056"/>
        <VariableRef variableId="V_FSP_OFDT" displayFormat="Dec.0" offset="0.0" gradient="1.0" unitCode="1056"/>
      </Menu>
      <Menu id="M_OMSR_Observe">
        <VariableRef variableId="V_ProcessDataInput" contextConstraints="ObservationMenu" checkElement="minOccurs 0"/>
        <VariableRef variableId="V_ProcessDataOutput" contextConstraints="ObservationMenu" checkElement="minOccurs 0"/>
        <RecordItemRef variableId="V_ProcessDataInput" subindex="#tbd#" contextConstraints="ObservationMenu" checkElement="minOccurs 0" checkAttributes="option displayFormat, option offset, option gradient, option unitCode"/>
        <RecordItemRef variableId="V_ProcessDataOutput" subindex="#tbd#" contextConstraints="ObservationMenu" checkElement="minOccurs 0" checkAttributes="option displayFormat, option offset, option gradient, option unitCode"/>
      </Menu>
    </MenuCollection>
    <ObserverRoleMenuSet>
      <IdentificationMenu menuId="M_OR_Ident" checkElement="minOccurs 1"/>
      <ParameterMenu menuId="M_OR_Param"/>
      <ObservationMenu menuId="M_OMSR_Observe" checkElement="minOccurs 1"/>
      <DiagnosisMenu menuId="M_OR_Diag" checkElement="minOccurs 1"/>
    </ObserverRoleMenuSet>
    <MaintenanceRoleMenuSet>
      <IdentificationMenu menuId="M_MSR_Ident" checkElement="minOccurs 1"/>
      <ParameterMenu menuId="M_MSR_Param"/>
      <ObservationMenu menuId="M_OMSR_Observe" checkElement="minOccurs 1"/>
      <DiagnosisMenu menuId="M_MSR_Diag" checkElement="minOccurs 1"/>
    </MaintenanceRoleMenuSet>
    <SpecialistRoleMenuSet>
      <IdentificationMenu menuId="M_MSR_Ident" checkElement="minOccurs 1"/>
      <ParameterMenu menuId="M_MSR_Param"/>
      <ObservationMenu menuId="M_OMSR_Observe" checkElement="minOccurs 1"/>
      <DiagnosisMenu menuId="M_MSR_Diag" checkElement="minOccurs 1"/>
    </SpecialistRoleMenuSet>
  </UserInterface>
  <ExternalTextCollection>
    <PrimaryLanguage xml:lang="en" checkElement="minOccurs 1">
      <Text id="TN_P_PDin" value="PD Input"/>
      <Text id="TN_P_PDout" value="PD Output"/>
      <Text id="TN_V_FSP_SafetyCode" value="FS Safety Code"/>
      <Text id="TD_V_FSP_SafetyCode" value="Control/Status octet and CRC."/>
      <Text id="TN_V_FSP_Authenticity" value="Authenticity"/>
      <Text id="TD_V_FSP_Authenticity" value="Authenticity parameters."/>
      <Text id="TN_V_FSP_Authenticity_1" value="FSP_Authenticity_1"/>
      <Text id="TD_V_FSP_Authenticity_1" value="&quot;A-Code&quot; from the upper level FSCP system."/>
      <Text id="TN_V_FSP_Authenticity_2" value="FSP_Authenticity_2"/>
      <Text id="TD_V_FSP_Authenticity_2" value="Extended &quot;A-Code&quot; from the upper level FSCP system."/>
      <Text id="TN_V_FSP_Port" value="FSP_Port"/>
      <Text id="TD_V_FSP_Port" value="PortNumber identifying the particular FS-Device."/>
      <Text id="TN_V_FSP_AuthentCRC" value="FSP_AuthentCRC"/>
      <Text id="TD_V_FSP_AuthentCRC" value="CRC-16 across authenticity parameters."/>
      <Text id="TN_V_FSP_Protocol" value="Protocol"/>
      <Text id="TD_V_FSP_Protocol" value="Protocol parameters."/>
      <Text id="TN_V_FSP_ProtVersion" value="FSP_ProtVersion"/>
      <Text id="TD_V_FSP_ProtVersion" value="Protocol version (1=current version)."/>
      <Text id="TN_V_FSP_ProtMode" value="FSP_ProtMode"/>
      <Text id="TD_V_FSP_ProtMode" value="Protocol mode (1=16 bit CRC, 2=32 bit CRC)"/>
      <Text id="TN_SV_FSP_ProtMode_crc16" value="16 bit CRC" checkElement="minOccurs 0"/>
      <Text id="TN_SV_FSP_ProtMode_crc32" value="32 bit CRC" checkElement="minOccurs 0"/>
      <Text id="TN_V_FSP_Watchdog" value="FSP_Watchdog"/>
```

```
        <Text id="TD_V_FSP_Watchdog" value="Monitoring of IO update."/>
        <Text id="TN_V_FSP_IO_StructCRC" value="FSP_IO_StructCRC"/>
        <Text id="TD_V_FSP_IO_StructCRC" value="CRC-16 across IO structure description block."/>
        <Text id="TN_V_FSP_TechParCRC" value="FSP_TechParCRC"/>
        <Text id="TD_V_FSP_TechParCRC" value="Securing code across FST (technology specific
parameter)."/>
        <Text id="TN_V_FSP_ProtParCRC" value="FSP_ProtParCRC"/>
        <Text id="TD_V_FSP_ProtParCRC" value="CRC-16 across protocol parameters"/>
        <Text id="TN_V_FSP_ParamDescCRC" value="FSP_ParamDescCRC"/>
        <Text id="TD_V_FSP_ParamDescCRC" value="CRC-32 signature securing authenticity, protocol,
and FS I/O structure description within IODD."/>
        <Text id="TN_V_FSP_TimeToReady" value="Time To Ready"/>
        <Text id="TD_V_FSP_TimeToReady" value="The time that is required by the device after power
on in order to be ready for communication."/>
        <Text id="TN_V_FSP_MinShutDownTime" value="Minimum shut down time"/>
        <Text id="TD_V_FSP_MinShutDownTime" value="Minimum time to wait after shut down of FS-
Device."/>
        <Text id="TN_V_FSP_WCDT" value="Worst-case Delay Time"/>
        <Text id="TD_V_FSP_WCDT" value="Time from triggering an FS-Device (sensor) until the
output shows a corresponding signal change or Process Data change. For an FS-Device (actuator)
it is the time from signal change or Process Data change to the actuator's safe state."/>
        <Text id="TN_V_FSP_OFDT" value="One Fault Delay Time"/>
        <Text id="TD_V_FSP_OFDT" value="Time from occurrence of a fault within the FS-Device to a
corresponding signal change or Process Data change"/>
        <Text id="TN_EV_FSP_TransmissionError_CRCSignature" value="Transmission error (CRC
signature)"/>
        <Text id="TD_EV_FSP_TransmissionError_CRCSignature" value=""/>
        <Text id="TN_EV_FSP_TransmissionError_Counter" value="Transmission error (Counter)"/>
        <Text id="TD_EV_FSP_TransmissionError_Counter" value=""/>
        <Text id="TN_EV_FSP_TransmissionError_Timeout" value="Transmission error (Timeout)"/>
        <Text id="TD_EV_FSP_TransmissionError_Timeout" value=""/>
        <Text id="TN_EV_FSP_UnexpectedAuthenticationCode" value="Unexpected authentication code"/>
        <Text id="TD_EV_FSP_UnexpectedAuthenticationCode" value=""/>
        <Text id="TN_EV_FSP_UnexpectedAuthenticationPort" value="Unexpected authentication port"/>
        <Text id="TD_EV_FSP_UnexpectedAuthenticationPort" value=""/>
        <Text id="TN_EV_FSP_Incorrect_AuthentCRC" value="Incorrect FSP_AuthentCRC"/>
        <Text id="TD_EV_FSP_Incorrect_AuthentCRC" value=""/>
        <Text id="TN_EV_FSP_Incorrect_ProtParCRC" value="Incorrect FSP_ProtParCRC"/>
        <Text id="TD_EV_FSP_Incorrect_ProtParCRC" value=""/>
        <Text id="TN_EV_FSP_Incorrect_TechParCRC" value="Incorrect FSP_TechParCRC"/>
        <Text id="TD_EV_FSP_Incorrect_TechParCRC" value=""/>
        <Text id="TN_EV_FSP_Incorrect_IO_StructCRC" value="Incorrect FSP_IO_StructCRC"/>
        <Text id="TD_EV_FSP_Incorrect_IO_StructCRC" value=""/>
        <Text id="TN_EV_FSP_WatchdogTimeOutOfSpec" value="Watchdog time out of specification"/>
        <Text id="TD_EV_FSP_WatchdogTimeOutOfSpec" value=""/>
        <Text id="TN_EV_FSP_NoFSVerifyRecord" value="No FS_VerifyRecord received"/>
        <Text id="TD_EV_FSP_NoFSVerifyRecord" value=""/>
        <Text id="TN_M_FST_Param" value="FS Technology Parameter"/>
        <Text id="TN_M_FSP_Param" value="FS Protocol Parameter"/>
        <Text id="TN_M_FSP_Param_Aux" value="FS Auxiliary Parameter"/>
      </PrimaryLanguage>
    </ExternalTextCollection>
</IODDProfileDefinitions>
```

E.3.3 Behavior of "Reset" SystemCommands in SDCI-FS

Table E.8 shows the specific behavior of the "reset" SystemCommands in FS-Devices (see IEC 61131-9). None of these SystemCommands is accepted by the FS-Device in "armed" mode.

Table E.8 – Specific behavior of FS-Device "Reset" SystemCommands

Command (hex)	Command (dec)	Command name	H/M/O	Definition
…				
0x80	128	Device reset	–	Not permitted for implementation in FS-Device
0x81	129	Application reset	M	Permitted in commissioning mode. Authenticity and Protocol parameters shall not be changed.
0x82	130	Restore factory settings	–	Not permitted for implementation in FS-Device
0x83	131	Back-to-box	M	This command shall only be effective whenever the parameter value of FSP_TechParCRC is "0" (commissioning phase)

Command (hex)	Command (dec)	Command name	H/M/O	Definition
…				
Key H = higly recommended; M = mandatory; O = optional				

E.3.4 Profile Characteristic

The identifier for the common profile SDCI-FS is 16 385 or 0x4001 (see E.5.8). The function class 0x8020 is reserved for future use.

E.3.5 ProcessDataInput and ProcessDataOutput

These variables shall be implemented. The sample IODD in E.5.8 shows details.

E.4 IODD conventions

E.4.1 Naming

While this document and IEC 61131-9 use "parameter" for any data object of a Device and FS-Device, IODDs in [16] use "variable" instead and thus all those data objects are indicated via the prefix "V_". The following rules apply:

1) Naming of non-safety parameters shall be "V_xxx". Prefixes "V_FSP", "V_FST" shall be omitted for FS-Devices.
2) Naming of FST technology safety parameters shall be "V_FST_xxx".
3) Naming of FSP safety parameters shall be "V_FSP_xxx".

These naming conventions shall only be used for SDCI-FS.

E.4.2 Process Data (PD)

The following rules apply for Process Data:

1) PDin and PDout shall be described as record.
4) Subindices shall be used within the records to differentiate between safety PD and non-safety PD.
5) Subindices 1 to 126 shall be used to describe safety PD starting with the highest bit offset.
6) Safety Code (see C.5) shall not be described in detail within the IODD. However, Subindex 127 shall be used to describe the Safety Code by means of an OctetStringT (4 or 6 octets) as a dummy to indicate the length of the Safety Code.
7) Subindices 128 to 255 shall be used to describe non-safety PD.
8) Multiple PD structure definitions selected by conditions are not permitted. This does not impact switching of the user interface to display scaling and units, e.g. °C and °F via conditions.

E.4.3 IODD conventions for user interface

The following rules apply for user interface:

1) The IODD shall contain different headlines (menu IDs) for the parameter block types "FS Technology", "FS Protocol", and "FS Auxiliary" in this order.
2) FS Technology parameters shall only be referenced in menus marked with the menu ID prefix M_xxx_FST.
3) FS Protocol parameters shall only be referenced in menus marked with the menu ID prefix M_xxx_FSP.
4) NSR parameters shall not be referenced in menus containing FS parameters.
5) The prefixes "FSP" and "FST" shall only be used for FS variables. Corresponding menus shall be colored in yellow.

6) User roles are "Observer", "Maintenance", "Specialist". The menus are organized for the "Observer" role (prefix OR) and the combined maintenance and specialist role (prefix MSR). Menus covering all user roles are marked with the prefix OMSR. The menu IDs shall be structured and named as follows:

M_OR_FST_Param
M_MSR_FST_Param
M_OR_FSP_Param
M_MSR_FSP_Param
M_OMSR_FSP_Param_Aux

E.4.4 Master tool features

The following rules on how to present the IODD to the user are highly recommended:

1) IODD interpreter in Master tools should show headlines not only for PDin and PDout but also for SR and NSR PD. These headlines should use yellow colors.

2) In case of PD observation via ISDU access the variable names should be the same as with cyclic PDs.

E.5 Securing

E.5.1 General

An IODD-based non-safety viewer calculates the 32-bit CRC signature across the FSP parameter description within the IODD. The algorithm for the calculation is shown in this Annex. The safety-related interpreter of the FS-Master tool checks the correctness of the imported IODD data. Parameter names associated to Index/Subindex are known in the interpreter and can be checked in a safe manner.

An IODD checker is not safety-related and thus not sufficient.

Only one IODD per DeviceID is permitted. A particular FS-Device (hardware) can have two DeviceIDs for example a current DeviceID and a DeviceID of a previous software version.

Figure E.1 shows the algorithm to build the FSP_ParamDescCRC signature. The algorithm shall be used across the Authenticity and the Protocol block (see Table A.1). A seed value "0" shall be used (see D.3.6).

1. General rule: All numerical values are serialized in **big-endian octet order** (most significant octet first).
2. Serialize the **Index** (16-bit unsigned integer) of the FS parameter.
3. Serialize the **bitLength** (16-bit unsigned integer) of the RecordT.
4. Sort the **RecordItems** in ascending order by Subindex.
5. For each **RecordItem** (including the last one) serialize:
   a) The **Subindex** (8-bit unsigned integer)
   b) The **bitOffset** (16-bit unsigned integer)
   c) The **Datatype** (8-bit unsigned integer): 1 = UIntegerT(8), 2 = UIntegerT(16), 3 = UIntegerT(32)
   d) If and only if a **DefaultValue** is given in the IODD: The DefaultValue (8/16/32 bit unsigned integer according to data type).
   e) If and only if **SingleValues** or a **ValueRange** is given in the IODD: The allowed values. A list of SingleValues is serialized as a sequence of these values, in ascending order. A ValueRange is serialized as the sequence of the minimum and maximum value. Whether SingleValues and/or a ValueRange are allowed depends on the specific RecordItem. See Table E.4.
6. Calculate the 4-octet CRC across the octet stream using the CRC polynomial 0xF4ACFB13.

Figure E.1 – Algorithm to build the FSP parameter CRC signatures

E.5.2 DefaultValues for FSP

The DefaultValues for FSP_Authenticity1/2, FSP_Port, FSP_AuthentCRC, FSP_TechParCRC, and FSP_ProtParCRC shall be "0". Table E.9 demonstrates the user actions to replace the default values by actual values.

Table E.9 – User actions to replace DefaultValues

Parameter	User actions
FSP_Authenticity1/2	During commissioning, the Authenticity values can be acquired from the gateway and displayed by the Master tool. SCL will not start with the default value.
FSP_Port	The user shall replace the default "0" by an allowed number with the help of the Master tool during commissioning. SCL will not start with the default value.
FSP_AuthentCRC	Master tool calculates this CRC signature
FSP_TechParCRC	The user parameterizes the FS-Device during commissioning or maintenance and uses a Dedicated Tool to calculate the actual value (see 11.7.8 and 11.7.9)
FSP_ProtParCRC	Master tool calculates this CRC signature

E.5.3 FSP_Authenticity

The values of the authenticity parameters cannot be defined within the IODD. They are maintained by the FS-Master tool.

E.5.4 FSP_Protocol

The limited variability of the protocol parameters requires the securing mechanism specified in E.5.1.

Table E.10 lists the RecordItems of FSP_Protocol to be serialized.

Table E.10 – RecordItems of FSP_Protocol where allowed values shall be serialized

RecordItem	Serialized as
FSP_ProtVersion	List of 8-bit unsigned integer containing the allowed values, in ascending order
FSP_ProtMode	List of 8-bit unsigned integer containing the allowed values, in ascending order
FSP_Watchdog	Minimum value and maximum value of the contiguous range of allowed values
Any other	All values according to the data type are allowed, therefore nothing is serialized

E.5.5 FSP_IO_Description

The FSP_IO_Description parameters do not require a particular securing mechanism since these instance values are straight forward. The IODD designer can calculate the CRC signature already and place it into the IODD (see A.2.7).

E.5.6 Sample serialization for FSP_ParamDescCRC

Table E.11 shows a sample serialization for the calculation of the FSP_ParamDescCRC signature in E.5.7. A seed value "0" shall be used since there are no leading zeros (see D.3.6).

Table E.11 – Sample serialization for FSP_ParamDescCRC

Offset	Serialization	IODD items	Expected values
0000	42 00	index	42 00 *(≠ 0)*
0002	00 58	bitLength of index	00 58
0004	01	subindex	01 *(Authenticity 1)*
0005	00 38	bitOffset	00 38
0007	03	xsi:type=UIntegerT, bitLength=32	03
0008	00 00 00 00	RecordItemInfo/@defaultValue	00 00 00 00

Offset	Serialization	IODD items	Expected values
000C	02	subindex	02 *(Authenticity 2)*
000D	00 18	bitOffset	00 18
000F	03	xsi:type=UIntegerT, bitLength=32	03
0010	00 00 00 00	RecordItemInfo/@defaultValue	00 00 00 00
0014	03	subindex	03 *(Port)*
0015	00 10	bitOffset	00 10
0017	01	xsi:type=UIntegerT, bitLength=8	01
0018	00	RecordItemInfo/@defaultValue	00
0019	04	subindex	04 *(AuthentCRC)*
001A	00 00	bitOffset	00 00
001C	02	xsi:type=UIntegerT, bitLength=16	02
001D	00 00	RecordItemInfo/@defaultValue	00 00 *(dummy CRC)*
001F	42 01	index	42 01
0021	00 60	bitLength of index	00 60
0023	01	subindex	01 *(ProtVersion)*
0024	00 58	bitOffset	00 58
0026	01	xsi:type=UIntegerT, bitLength=8	01
0027	01	RecordItemInfo/@defaultValue	01
0028	01	SingleValue/@value	01 *(this document)*
0029	02	subindex	02 *(ProtMode)*
002A	00 50	bitOffset	00 50
002C	01	xsi:type=UIntegerT, bitLength=8	01
002D	02	RecordItemInfo/@defaultValue	*(Vendor defined)*
002E	02	SingleValue/@value	02 *(example 32 bit)*
002F	03	subindex	03 *(Watchdog)*
0030	00 40	bitOffset	00 40
0032	02	xsi:type=UIntegerT, bitLength=16	02
0033	00 64	RecordItemInfo/@defaultValue	*(Vendor defined)*
0035	00 64	ValueRange/@lowerValue	00 64 *(example: 100)*
0037	13 88	ValueRange/@upperValue	13 88 *(example: 5000)*
0039	04	subindex	04 *(IO_StructCRC)*
003A	00 30	bitOffset	00 30
003C	02	xsi:type=UIntegerT, bitLength=16	02
003D	9A 28	RecordItemInfo/@defaultValue (see A.2.7)	*(Vendor defined)*
003F	05	subindex	05 *(TechParCRC)*
0040	00 10	bitOffset	00 10
0042	03	xsi:type=UIntegerT, bitLength=32	03
0043	00 00 00 00	RecordItemInfo/@defaultValue	00 00 00 00 *(dummy CRC)*
0047	06	subindex	06 *(ProtParCRC)*
0048	00 00	bitOffset	00 00
004A	02	xsi:type=UIntegerT, bitLength=16	02
004B	00 00	RecordItemInfo/@defaultValue	00 00 *(dummy CRC)*
Calculated 32-bit FSP_ParamDescCRC signature value: 1860635738			See E.5.7

The sample serialization in Table E.11 shows 77 octets to be secured via the CRC-32 polynomial listed in Table D.1. FS-Master tool shall check the signature after import of the IODD. Only a few values are variable and "*Vendor defined*" (see offsets: 002D, 002E, 0033 to 0037, and 003D). FS-Master tool can compare the remaining values with preset values as an additional safety measure.

The "*dummy CRC*" are placeholders to be replaced by the FS-Master tool once the user assigned the actual parameter values.

E.5.7 FST and FSP parameters and Data Storage

FST parameters shall be described within the IODD. A "packed" parameter transfer via one ISDU that is not described within the IODD is possible for Data Storage as long as the result in the Device/FS-Device is the same as with discrete ISDUs (see 11.7.6). A manufacturer/vendor is responsible to guarantee this behavior.

FSP parameters (authenticity and protocol) shall be described within the IODD also and are part of Data Storage.

E.5.8 Sample IODD of an FS-Device

The following XML code represents the sample code of an FS-Device IODD. It refers to the Process Data example in Figure A.2. A complete IODD file with name *IO-Link-Safety-01-20220121-IODD1.1.xml* can be downloaded from the website referenced in Annex I.

This sample IODD contains already calculated CRC signature values:

```
<?xml version="1.0" encoding="utf-8"?>
<IODevice xmlns="http://www.io-link.com/IODD/2010/10"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.io-
link.com/IODD/2010/10 IODD1.1.xsd">
 <DocumentInfo version="V1.00" releaseDate="2022-01-21" copyright="Copyright 2022 IO-Link
Community"/>
 <ProfileHeader>
  <ProfileIdentification>IO Device Profile</ProfileIdentification>
  <ProfileRevision>1.1</ProfileRevision>
  <ProfileName>Device Profile for IO Devices</ProfileName>
  <ProfileSource>IO-Link Consortium</ProfileSource>
  <ProfileClassID>Device</ProfileClassID>
  <ISO15745Reference>
   <ISO15745Part>1</ISO15745Part>
   <ISO15745Edition>1</ISO15745Edition>
   <ProfileTechnology>IODD</ProfileTechnology>
  </ISO15745Reference>
 </ProfileHeader>
 <ProfileBody>
  <DeviceIdentity vendorId="65535" vendorName="IO-Link Community" deviceId="4608">
   <VendorText textId="T_VendorText"/>
   <VendorUrl textId="T_VendorUrl"/>
   <DeviceName textId="T_DeviceName"/>
   <DeviceFamily textId="T_DeviceFamily"/>
   <DeviceVariantCollection>
    <DeviceVariant productId="Safety-01" deviceSymbol="IO-Link-Safety-pic.png" deviceIcon="IO-
Link-Safety-icon.png">
     <Name textId="TN_ProductName"/>
     <Description textId="TD_ProductDescr"/>
    </DeviceVariant>
   </DeviceVariantCollection>
  </DeviceIdentity>
  <DeviceFunction>
   <Features blockParameter="true" dataStorage="true" profileCharacteristic="16384 16385">
    <SupportedAccessLocks parameter="false" dataStorage="false" localParameterization="false"
localUserInterface="false"/>
   </Features>
   <DatatypeCollection>
    <Datatype id="D_FSP_Authenticity" xsi:type="RecordT" bitLength="88"
subindexAccessSupported="false">
     <RecordItem subindex="1" bitOffset="56">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
      <Name textId="TN_V_FSP_Authenticity_1"/>
      <Description textId="TD_V_FSP_Authenticity_1"/>
     </RecordItem>
     <RecordItem subindex="2" bitOffset="24">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
```

```
      <Name textId="TN_V_FSP_Authenticity_2"/>
      <Description textId="TD_V_FSP_Authenticity_2"/>
     </RecordItem>
     <RecordItem subindex="3" bitOffset="16">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="8"/>
      <Name textId="TN_V_FSP_Port"/>
      <Description textId="TD_V_FSP_Port"/>
     </RecordItem>
     <RecordItem subindex="4" bitOffset="0">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
      <Name textId="TN_V_FSP_AuthentCRC"/>
      <Description textId="TD_V_FSP_AuthentCRC"/>
     </RecordItem>
    </Datatype>
    <Datatype id="D_FSP_Protocol" xsi:type="RecordT" bitLength="96"
subindexAccessSupported="false">
     <RecordItem subindex="1" bitOffset="88">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="8">
       <SingleValue value="1"/>
      </SimpleDatatype>
      <Name textId="TN_V_FSP_ProtVersion"/>
      <Description textId="TD_V_FSP_ProtVersion"/>
     </RecordItem>
     <RecordItem subindex="2" bitOffset="80">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="8">
       <!-- Which of these SingleValues is supported is device specific. Only one shall be
referenced. -->
       <!--
       <SingleValue value="1" checkElement="minOccurs 0">
        <Name textId="TN_SV_FSP_ProtMode_crc16"/>
       </SingleValue>
       -->
       <SingleValue value="2">
        <Name textId="TN_SV_FSP_ProtMode_crc32"/>
       </SingleValue>
      </SimpleDatatype>
      <Name textId="TN_V_FSP_ProtMode"/>
      <Description textId="TD_V_FSP_ProtMode"/>
     </RecordItem>
     <RecordItem subindex="3" bitOffset="64">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="16">
       <!-- Which ValueRange is supported is device specific (but the lowerValue must be >0) -->
       <ValueRange lowerValue="100" upperValue="5000"/>
      </SimpleDatatype>
      <Name textId="TN_V_FSP_Watchdog"/>
      <Description textId="TD_V_FSP_Watchdog"/>
     </RecordItem>
     <RecordItem subindex="4" bitOffset="48">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
      <Name textId="TN_V_FSP_IO_StructCRC"/>
      <Description textId="TD_V_FSP_IO_StructCRC"/>
     </RecordItem>
     <RecordItem subindex="5" bitOffset="16">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
      <Name textId="TN_V_FSP_TechParCRC"/>
      <Description textId="TD_V_FSP_TechParCRC"/>
     </RecordItem>
     <RecordItem subindex="6" bitOffset="0">
      <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
      <Name textId="TN_V_FSP_ProtParCRC"/>
      <Description textId="TD_V_FSP_ProtParCRC"/>
     </RecordItem>
    </Datatype>
    <Datatype id="D_FST_State" xsi:type="BooleanT">
     <SingleValue value="false">
      <Name textId="TN_SV_FST_State_inactive"/>
     </SingleValue>
     <SingleValue value="true">
      <Name textId="TN_SV_FST_State_active"/>
     </SingleValue>
    </Datatype>
    <Datatype id="D_FSP_PDin" xsi:type="RecordT" bitLength="112"
subindexAccessSupported="false">
     <!-- Technology specific safety process data have subindex 1..126 -->
     <!-- Datatype order for subindices is: boolean, integer16, integer32 -->
     <!-- Boolean array  -->
     <!-- There may be no gaps between the booleans, but the last octet may contain less than
eight booleans. -->
```

```
        <RecordItem subindex="1" bitOffset="104">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_1"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="2" bitOffset="105">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_2"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="3" bitOffset="106">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_3"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="4" bitOffset="107">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_4"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="5" bitOffset="108">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_5"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="6" bitOffset="109">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_6"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="7" bitOffset="110">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_7"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="8" bitOffset="111">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_8"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="9" bitOffset="96">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_9"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="10" bitOffset="97">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_10"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="11" bitOffset="98">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_11"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="12" bitOffset="99">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_12"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <RecordItem subindex="13" bitOffset="100">
         <DatatypeRef datatypeId="D_FST_State"/>
         <Name textId="TN_V_FST_State_13"/>
         <Description textId="TD_V_FST_State"/>
        </RecordItem>
        <!-- Integer16 values -->
        <!-- Integer values always are aligned to byte borders. -->
        <RecordItem subindex="14" bitOffset="80">
         <SimpleDatatype xsi:type="IntegerT" bitLength="16"/>
         <Name textId="TN_V_FST_Integer16Value"/>
         <Description textId="TD_V_FST_Integer16Value"/>
        </RecordItem>
        <!-- Safety Code has fixed subindex 127 -->
        <RecordItem subindex="127" bitOffset="32">
         <SimpleDatatype xsi:type="OctetStringT" fixedLength="6"/>
         <Name textId="TN_V_FSP_SafetyCode"/>
         <Description textId="TD_V_FSP_SafetyCode"/>
        </RecordItem>
```

```xml
        <!-- Non-safety process data has subindex 128..255 -->
        <RecordItem subindex="128" bitOffset="0">
         <SimpleDatatype xsi:type="UIntegerT" bitLength="32"/>
         <Name textId="TN_V_Revolutions"/>
         <Description textId="TD_V_Revolutions"/>
        </RecordItem>
       </Datatype>
       <Datatype id="D_FSP_PDout" xsi:type="RecordT" bitLength="48"
subindexAccessSupported="false">
        <!-- Technology specific safety process data have subindex 1..126 -->
        <!-- Datatype order for subindices is: boolean, integer16, integer32 -->
        <!-- Safety Code has fixed subindex 127 -->
        <RecordItem subindex="127" bitOffset="0">
         <SimpleDatatype xsi:type="OctetStringT" fixedLength="6"/>
         <Name textId="TN_V_FSP_SafetyCode"/>
         <Description textId="TD_V_FSP_SafetyCode"/>
        </RecordItem>
        <!-- Non-safety process data has subindex 128..255 -->
       </Datatype>
      </DatatypeCollection>
      <VariableCollection>
       <StdVariableRef id="V_DirectParameters_1"/>
       <StdVariableRef id="V_SystemCommand">
        <StdSingleValueRef value="129"/>
        <StdSingleValueRef value="131"/>
       </StdVariableRef>
       <StdVariableRef id="V_VendorName" defaultValue="IO-Link Community"/>
       <StdVariableRef id="V_VendorText" defaultValue="http://www.io-link.com"/>
       <StdVariableRef id="V_ProductName" defaultValue="IO-Link Safety Device"/>
       <StdVariableRef id="V_ProductID" defaultValue="Safety-01"/>
       <StdVariableRef id="V_ProductText" defaultValue="Sample IODD for a device with IO-Link
Safety"/>
       <StdVariableRef id="V_SerialNumber"/>
       <StdVariableRef id="V_HardwareRevision"/>
       <StdVariableRef id="V_FirmwareRevision"/>
       <StdVariableRef id="V_ApplicationSpecificTag" excludedFromDataStorage="false"
defaultValue="***"/>
       <StdVariableRef id="V_DeviceStatus" defaultValue="0"/>
       <StdVariableRef id="V_DetailedDeviceStatus" fixedLengthRestriction="4"/>
       <StdVariableRef id="V_ProcessDataInput"/>
       <StdVariableRef id="V_ProcessDataOutput"/>
       <Variable id="V_CP_FunctionTag" index="25" accessRights="rw" excludedFromDataStorage="false"
defaultValue="***">
        <Datatype xsi:type="StringT" fixedLength="32" encoding="UTF-8"/>
        <Name textId="TN_V_CP_FunctionTag"/>
        <Description textId="TD_V_CP_FunctionTag"/>
       </Variable>
       <Variable id="V_CP_LocationTag" index="26" accessRights="rw" excludedFromDataStorage="false"
defaultValue="***">
        <Datatype xsi:type="StringT" fixedLength="32" encoding="UTF-8"/>
        <Name textId="TN_V_CP_LocationTag"/>
        <Description textId="TD_V_CP_LocationTag"/>
       </Variable>
       <Variable index="16896" id="V_FSP_Authenticity" accessRights="rw">
        <DatatypeRef datatypeId="D_FSP_Authenticity"/>
        <RecordItemInfo subindex="1" defaultValue="0"/>
        <RecordItemInfo subindex="2" defaultValue="0"/>
        <RecordItemInfo subindex="3" defaultValue="0"/>
        <RecordItemInfo subindex="4" defaultValue="0"/>
        <Name textId="TN_V_FSP_Authenticity"/>
        <Description textId="TD_V_FSP_Authenticity"/>
       </Variable>
       <Variable index="16897" id="V_FSP_Protocol" accessRights="rw">
        <DatatypeRef datatypeId="D_FSP_Protocol"/>
        <!-- FSP_ProtVersion: 1= valid -->
        <RecordItemInfo subindex="1" defaultValue="1"/>
        <!-- FSP_ProtMode: 1  (16 bit CRC) or 2 (32 bit CRC)= valid -->
        <RecordItemInfo subindex="2" defaultValue="2"/>
        <!-- FSP_Watchdog: 1 .. 65535 = valid -->
        <RecordItemInfo subindex="3" defaultValue="100"/>
        <!-- FSP_IO_StructCRC: = valid -->
        <RecordItemInfo subindex="4" defaultValue="39464"/>
        <!-- FSP_TechParCRC: 0= invalid -->
        <RecordItemInfo subindex="5" defaultValue="0"/>
        <!-- FSP_ProtParCRC: 0= invalid -->
        <RecordItemInfo subindex="6" defaultValue="0"/>
        <Name textId="TN_V_FSP_Protocol"/>
        <Description textId="TD_V_FSP_Protocol"/>
```

```
   </Variable>
   <!-- Note: Variable FSP_VerifyRecord (index="16898") shall not be describedin the IODD. -->
   <Variable id="V_FSP_TimeToReady" index="16912" accessRights="ro" defaultValue="10">
    <!-- resolution 10 ms -->
    <Datatype xsi:type="UIntegerT" bitLength="16"/>
    <Name textId="TN_V_FSP_TimeToReady"/>
    <Description textId="TD_V_FSP_TimeToReady"/>
   </Variable>
   <Variable id="V_FSP_MinShutDownTime" index="16913" accessRights="ro" defaultValue="100">
    <!-- resolution 10 ms -->
    <Datatype xsi:type="UIntegerT" bitLength="16"/>
    <Name textId="TN_V_FSP_MinShutDownTime"/>
    <Description textId="TD_V_FSP_MinShutDownTime"/>
   </Variable>
   <Variable id="V_FSP_ParamDescCRC" index="16914" accessRights="ro" defaultValue="1860635738">
    <Datatype xsi:type="UIntegerT" bitLength="32"/>
    <Name textId="TN_V_FSP_ParamDescCRC"/>
    <Description textId="TD_V_FSP_ParamDescCRC"/>
   </Variable>
   <Variable id="V_FSP_WCDT" index="16915" accessRights="ro" defaultValue="10">
    <!-- in ms -->
    <Datatype xsi:type="UIntegerT" bitLength="16"/>
    <Name textId="TN_V_FSP_WCDT"/>
    <Description textId="TD_V_FSP_WCDT"/>
   </Variable>
   <Variable id="V_FSP_OFDT" index="16916" accessRights="ro" defaultValue="10">
    <!-- in ms -->
    <Datatype xsi:type="UIntegerT" bitLength="16"/>
    <Name textId="TN_V_FSP_OFDT"/>
    <Description textId="TD_V_FSP_OFDT"/>
   </Variable>
   <!-- Non-Safety (Standard) Parameter example -->
   <Variable index="64" id="V_NonSafetyParameter" accessRights="rw" defaultValue="123">
    <Datatype xsi:type="IntegerT" bitLength="16"/>
    <Name textId="TN_V_NonSafetyParameter"/>
    <Description textId="TD_V_NonSafetyParameter"/>
   </Variable>
   <!-- FS Technology Parameter example -->
   <Variable index="65" id="V_FST_DiscrepancyTime" accessRights="rw" defaultValue="0">
    <Datatype xsi:type="UIntegerT" bitLength="16"/>
    <Name textId="TN_V_FST_DiscrepancyTime"/>
    <Description textId="TD_V_FST_DiscrepancyTime"/>
   </Variable>
   <Variable index="66" id="V_FST_Filter" accessRights="rw" defaultValue="4">
    <Datatype xsi:type="UIntegerT" bitLength="16"/>
    <Name textId="TN_V_FST_Filter"/>
    <Description textId="TD_V_FST_Filter"/>
   </Variable>
  </VariableCollection>
  <ProcessDataCollection>
   <ProcessData id="P_ProcessData">
    <ProcessDataIn id="PI_PDin" bitLength="112">
     <DatatypeRef datatypeId="D_FSP_PDin"/>
     <Name textId="TN_PDin"/>
    </ProcessDataIn>
    <ProcessDataOut id="PO_PDout" bitLength="48">
     <DatatypeRef datatypeId="D_FSP_PDout"/>
     <Name textId="TN_PDout"/>
    </ProcessDataOut>
   </ProcessData>
  </ProcessDataCollection>
  <EventCollection>
   <Event code="45056" type="Warning">
    <Name textId="TN_EV_FSP_TransmissionError_CRCSignature"/>
    <Description textId="TD_EV_FSP_TransmissionError_CRCSignature"/>
   </Event>
   <Event code="45057" type="Warning">
    <Name textId="TN_EV_FSP_TransmissionError_Counter"/>
    <Description textId="TD_EV_FSP_TransmissionError_Counter"/>
   </Event>
   <Event code="45058" type="Error">
    <Name textId="TN_EV_FSP_TransmissionError_Timeout"/>
    <Description textId="TD_EV_FSP_TransmissionError_Timeout"/>
   </Event>
   <Event code="45059" type="Error">
    <Name textId="TN_EV_FSP_UnexpectedAuthenticationCode"/>
    <Description textId="TD_EV_FSP_UnexpectedAuthenticationCode"/>
   </Event>
```

```
   <Event code="45060" type="Error">
    <Name textId="TN_EV_FSP_UnexpectedAuthenticationPort"/>
    <Description textId="TD_EV_FSP_UnexpectedAuthenticationPort"/>
   </Event>
   <Event code="45061" type="Error">
    <Name textId="TN_EV_FSP_Incorrect_AuthentCRC"/>
    <Description textId="TD_EV_FSP_Incorrect_AuthentCRC"/>
   </Event>
   <Event code="45062" type="Error">
    <Name textId="TN_EV_FSP_Incorrect_ProtParCRC"/>
    <Description textId="TD_EV_FSP_Incorrect_ProtParCRC"/>
   </Event>
   <Event code="45063" type="Error">
    <Name textId="TN_EV_FSP_Incorrect_TechParCRC"/>
    <Description textId="TD_EV_FSP_Incorrect_TechParCRC"/>
   </Event>
   <Event code="45064" type="Error">
    <Name textId="TN_EV_FSP_Incorrect_IO_StructCRC"/>
    <Description textId="TD_EV_FSP_Incorrect_IO_StructCRC"/>
   </Event>
   <Event code="45065" type="Error">
    <Name textId="TN_EV_FSP_WatchdogTimeOutOfSpec"/>
    <Description textId="TD_EV_FSP_WatchdogTimeOutOfSpec"/>
   </Event>
   <Event code="45066" type="Error">
    <Name textId="TN_EV_FSP_NoFSVerifyRecord"/>
    <Description textId="TD_EV_FSP_NoFSVerifyRecord"/>
   </Event>
  </EventCollection>
  <UserInterface>
   <ProcessDataRefCollection>
    <ProcessDataRef processDataId="PI_PDin">
     <!-- Space for technology specific functional safety PD Input Data-->
     <ProcessDataRecordItemInfo subindex="1"/>
     <ProcessDataRecordItemInfo subindex="2"/>
     <ProcessDataRecordItemInfo subindex="3"/>
     <ProcessDataRecordItemInfo subindex="4"/>
     <ProcessDataRecordItemInfo subindex="5"/>
     <ProcessDataRecordItemInfo subindex="6"/>
     <ProcessDataRecordItemInfo subindex="7"/>
     <ProcessDataRecordItemInfo subindex="8"/>
     <ProcessDataRecordItemInfo subindex="9"/>
     <ProcessDataRecordItemInfo subindex="10"/>
     <ProcessDataRecordItemInfo subindex="11"/>
     <ProcessDataRecordItemInfo subindex="12"/>
     <ProcessDataRecordItemInfo subindex="13"/>
     <ProcessDataRecordItemInfo subindex="14"/>
     <!-- Safety Code -->
     <ProcessDataRecordItemInfo subindex="127" displayFormat="Hex"/>
     <!-- Space for technology specific non-safety PD Input Data-->
     <ProcessDataRecordItemInfo subindex="128"/>
    </ProcessDataRef>
    <ProcessDataRef processDataId="PO_PDout">
     <!-- Space for technology specific functional safety PD Output Data-->
     <!-- Safety Code -->
     <ProcessDataRecordItemInfo subindex="127" displayFormat="Hex"/>
     <!-- Space for technology specific non-safety PD Output Data-->
    </ProcessDataRef>
   </ProcessDataRefCollection>
   <MenuCollection>
    <Menu id="M_OR_Ident">
     <VariableRef variableId="V_VendorName"/>
     <VariableRef variableId="V_VendorText"/>
     <VariableRef variableId="V_ProductName"/>
     <VariableRef variableId="V_ProductText"/>
     <VariableRef variableId="V_ProductID"/>
     <VariableRef variableId="V_SerialNumber"/>
     <VariableRef variableId="V_HardwareRevision"/>
     <VariableRef variableId="V_FirmwareRevision"/>
     <VariableRef variableId="V_ApplicationSpecificTag" accessRightRestriction="ro"/>
     <VariableRef variableId="V_CP_FunctionTag" accessRightRestriction="ro"/>
     <VariableRef variableId="V_CP_LocationTag" accessRightRestriction="ro"/>
    </Menu>
    <Menu id="M_MSR_Ident">
     <VariableRef variableId="V_VendorName"/>
     <VariableRef variableId="V_VendorText"/>
     <VariableRef variableId="V_ProductName"/>
     <VariableRef variableId="V_ProductText"/>
```

```
      <VariableRef variableId="V_ProductID"/>
      <VariableRef variableId="V_SerialNumber"/>
      <VariableRef variableId="V_HardwareRevision"/>
      <VariableRef variableId="V_FirmwareRevision"/>
      <VariableRef variableId="V_ApplicationSpecificTag"/>
      <VariableRef variableId="V_CP_FunctionTag"/>
      <VariableRef variableId="V_CP_LocationTag"/>
     </Menu>
     <Menu id="M_OR_Param">
      <MenuRef menuId="M_OR_Param_NonSafety"/>
      <MenuRef menuId="M_OR_FST_Param"/>
      <MenuRef menuId="M_OR_FSP_Param"/>
      <MenuRef menuId="M_OMSR_FSP_Param_Aux"/>
     </Menu>
     <Menu id="M_MSR_Param">
      <MenuRef menuId="M_MSR_Param_NonSafety"/>
      <MenuRef menuId="M_MSR_FST_Param"/>
      <MenuRef menuId="M_MSR_FSP_Param"/>
      <MenuRef menuId="M_OMSR_FSP_Param_Aux"/>
      <MenuRef menuId="M_MSR_Param_GeneralSettings"/>
     </Menu>
     <Menu id="M_OR_Param_NonSafety">
      <Name textId="TN_M_Param_NonSafety"/>
      <VariableRef variableId="V_NonSafetyParameter" accessRightRestriction="ro"/>
     </Menu>
     <Menu id="M_MSR_Param_NonSafety">
      <Name textId="TN_M_Param_NonSafety"/>
      <VariableRef variableId="V_NonSafetyParameter" unitCode="1056"/>
     </Menu>
     <Menu id="M_OR_FST_Param">
      <Name textId="TN_M_FST_Param"/>
      <VariableRef variableId="V_FST_DiscrepancyTime" accessRightRestriction="ro"
unitCode="1056"/>
      <VariableRef variableId="V_FST_Filter" accessRightRestriction="ro" unitCode="1056"/>
     </Menu>
     <Menu id="M_MSR_FST_Param">
      <Name textId="TN_M_FST_Param"/>
      <VariableRef variableId="V_FST_DiscrepancyTime" unitCode="1056"/>
      <VariableRef variableId="V_FST_Filter" unitCode="1056"/>
     </Menu>
     <Menu id="M_OR_FSP_Param">
      <Name textId="TN_M_FSP_Param"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="1" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="2" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="3" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="4" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="1" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="2" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="3" accessRightRestriction="ro"
displayFormat="Dec.0" offset="0.0" gradient="1.0" unitCode="1056"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="4" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="5" accessRightRestriction="ro"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="6" accessRightRestriction="ro"/>
     </Menu>
     <Menu id="M_MSR_FSP_Param">
      <Name textId="TN_M_FSP_Param"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="1"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="2"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="3"/>
      <RecordItemRef variableId="V_FSP_Authenticity" subindex="4"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="1"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="2"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="3" displayFormat="Dec.0" offset="0.0"
gradient="1.0" unitCode="1056"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="4"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="5"/>
      <RecordItemRef variableId="V_FSP_Protocol" subindex="6"/>
     </Menu>
     <Menu id="M_OMSR_FSP_Param_Aux">
      <Name textId="TN_M_FSP_Param_Aux"/>
      <VariableRef variableId="V_FSP_TimeToReady" displayFormat="Dec.0" offset="0.0"
gradient="10.0" unitCode="1056"/>
             <VariableRef variableId="V_FSP_MinShutDownTime" displayFormat="Dec.0" offset="0.0"
gradient="10.0" unitCode="1056"/>
      <VariableRef variableId="V_FSP_WCDT" displayFormat="Dec.0" offset="0.0" gradient="1.0"
unitCode="1056"/>
      <VariableRef variableId="V_FSP_OFDT" displayFormat="Dec.0" offset="0.0" gradient="1.0"
unitCode="1056"/>
```

```xml
     </Menu>
     <Menu id="M_MSR_Param_GeneralSettings">
      <Name textId="TN_M_CP_Param_GeneralSettings"/>
      <VariableRef variableId="V_SystemCommand">
       <Button buttonValue="129">
        <Description textId="TD_STD_SystemCommand_ApplicationReset"/>
       </Button>
      </VariableRef>
     </Menu>
     <Menu id="M_OMSR_Observe">
      <VariableRef variableId="V_ProcessDataInput"/>
      <VariableRef variableId="V_ProcessDataOutput"/>
     </Menu>
     <Menu id="M_OR_Diag">
      <MenuRef menuId="M_OMSR_CP_Diag_DeviceStatusInfo"/>
     </Menu>
     <Menu id="M_MSR_Diag">
      <MenuRef menuId="M_OMSR_CP_Diag_DeviceStatusInfo"/>
      <MenuRef menuId="M_MSR_CP_Diag_ServiceFunctions"/>
     </Menu>
     <Menu id="M_OMSR_CP_Diag_DeviceStatusInfo">
      <Name textId="TN_M_CP_Diag_DeviceStatusInfo"/>
      <VariableRef variableId="V_DeviceStatus"/>
      <VariableRef variableId="V_DetailedDeviceStatus"/>
     </Menu>
     <Menu id="M_MSR_CP_Diag_ServiceFunctions">
      <Name textId="TN_M_CP_Diag_ServiceFunctions"/>
      <VariableRef variableId="V_SystemCommand">
       <Button buttonValue="131">
        <Description textId="TD_STD_SystemCommand_BackToBox"/>
       </Button>
      </VariableRef>
     </Menu>
    </MenuCollection>
    <ObserverRoleMenuSet>
     <IdentificationMenu menuId="M_OR_Ident"/>
     <ParameterMenu menuId="M_OR_Param"/>
     <ObservationMenu menuId="M_OMSR_Observe"/>
     <DiagnosisMenu menuId="M_OR_Diag"/>
    </ObserverRoleMenuSet>
    <MaintenanceRoleMenuSet>
     <IdentificationMenu menuId="M_MSR_Ident"/>
     <ParameterMenu menuId="M_MSR_Param"/>
     <ObservationMenu menuId="M_OMSR_Observe"/>
     <DiagnosisMenu menuId="M_MSR_Diag"/>
    </MaintenanceRoleMenuSet>
    <SpecialistRoleMenuSet>
     <IdentificationMenu menuId="M_MSR_Ident"/>
     <ParameterMenu menuId="M_MSR_Param"/>
     <ObservationMenu menuId="M_OMSR_Observe"/>
     <DiagnosisMenu menuId="M_MSR_Diag"/>
    </SpecialistRoleMenuSet>
   </UserInterface>
  </DeviceFunction>
 </ProfileBody>
 <CommNetworkProfile xsi:type="IOLinkCommNetworkProfileT" iolinkRevision="V1.1">
  <TransportLayers>
   <PhysicalLayer bitrate="COM3" minCycleTime="2000" sioSupported="true"
mSequenceCapability="43">
    <Connection xsi:type="M12-4ConnectionT" connectionSymbol="IO-Link-Safety-con-pic.png">
     <ProductRef productId="Safety-01"/>
     <Description textId="TD_Connection"/>
     <Wire1 function="L+" color="BN"/>
     <Wire2 function="Other" color="WH"/>
     <Wire3 function="L-" color="BU"/>
     <Wire4 function="C/Q" color="BK"/>
    </Connection>
   </PhysicalLayer>
  </TransportLayers>
  <Test>
   <Config1 index="64" testValue="0x55,0x99"/>
   <Config2 index="256" testValue="0x01,0x02,0x03,0x04"/>
   <Config3 index="24"
testValue="0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20"/>
   <Config7 index="254">
    <EventTrigger appearValue="01" disappearValue="02"/>
    <EventTrigger appearValue="03" disappearValue="04"/>
   </Config7>
```

```xml
  </Test>
 </CommNetworkProfile>
 <ExternalTextCollection>
  <PrimaryLanguage xml:lang="en">
   <Text id="T_VendorText" value="USE - Universal, Smart, Easy"/>
   <Text id="T_VendorUrl" value="http://www.io-link.com"/>
   <Text id="T_DeviceName" value="Safety Device"/>
   <Text id="T_DeviceFamily" value="Safety Device Family"/>
   <Text id="TN_ProductName" value="Safety Device 01"/>
   <Text id="TD_ProductDescr" value="Sample for a device with IO-Link Safety"/>
   <Text id="TD_Connection" value="Port type A with OSSDe. Pin 4 = C/Q/OSSD1e, Pin 2 = OSSD2e"/>
   <Text id="TN_PDin" value="PD Input"/>
   <Text id="TN_PDout" value="PD Output"/>
   <Text id="TN_V_CP_FunctionTag" value="Function Tag"/>
   <Text id="TD_V_CP_FunctionTag" value="Possibility to mark a device with function-specific
information."/>
   <Text id="TN_V_CP_LocationTag" value="Location Tag"/>
   <Text id="TD_V_CP_LocationTag" value="Possibility to mark a device with location-specific
information."/>
   <Text id="TD_STD_SystemCommand_ApplicationReset" value="The parameter of the technology-
specific application are set to default values. Identification parameter remain unchanged. An
upload to the data storage of the master will be executed, if activated in the port
configuration of the master."/>
   <Text id="TD_STD_SystemCommand_BackToBox" value="The parameter of the device are set to
factory default values and communication will be inhibited until the next power cycle. Note:
Directly detach the device from the master port!"/>
   <!-- IO-Link Non-Safety Parameter -->
   <Text id="TN_V_NonSafetyParameter" value="Sample Non-Safety Parameter"/>
   <Text id="TD_V_NonSafetyParameter" value="Configures any functionality of the device which is
not relevant for the safety function."/>
   <Text id="TN_V_Revolutions" value="Revolutions"/>
   <Text id="TD_V_Revolutions" value="Shows a non-safety measurement value."/>
   <!-- IO-Link Safety FS Technology Parameter -->
   <Text id="TN_V_FST_DiscrepancyTime" value="Discrepancy Time"/>
   <Text id="TD_V_FST_DiscrepancyTime" value="Defines the functional safety relevant value for
the technology specific discrepancy time."/>
   <Text id="TN_V_FST_Filter" value="Filter"/>
   <Text id="TD_V_FST_Filter" value="Defines the functional safety relevant value for the
technology specific signal filter setting."/>
   <Text id="TN_V_FST_State_1" value="FS-PD Boolean State 1"/>
   <Text id="TN_V_FST_State_2" value="FS-PD Boolean State 2"/>
   <Text id="TN_V_FST_State_3" value="FS-PD Boolean State 3"/>
   <Text id="TN_V_FST_State_4" value="FS-PD Boolean State 4"/>
   <Text id="TN_V_FST_State_5" value="FS-PD Boolean State 5"/>
   <Text id="TN_V_FST_State_6" value="FS-PD Boolean State 6"/>
   <Text id="TN_V_FST_State_7" value="FS-PD Boolean State 7"/>
   <Text id="TN_V_FST_State_8" value="FS-PD Boolean State 8"/>
   <Text id="TN_V_FST_State_9" value="FS-PD Boolean State 9"/>
   <Text id="TN_V_FST_State_10" value="FS-PD Boolean State 10"/>
   <Text id="TN_V_FST_State_11" value="FS-PD Boolean State 11"/>
   <Text id="TN_V_FST_State_12" value="FS-PD Boolean State 12"/>
   <Text id="TN_V_FST_State_13" value="FS-PD Boolean State 13"/>
   <Text id="TD_V_FST_State" value="Indicates the state of the boolean status data."/>
   <Text id="TN_SV_FST_State_inactive" value="Inactive"/>
   <Text id="TN_SV_FST_State_active" value="Active"/>
   <Text id="TN_V_FST_Integer16Value" value="FS-PD Integer 16 Value"/>
   <Text id="TD_V_FST_Integer16Value" value="Shows the FS-measurement value."/>
   <Text id="TN_V_FSP_SafetyCode" value="FS Safety Code"/>
   <Text id="TD_V_FSP_SafetyCode" value="Control/Status octet and CRC."/>
   <Text id="TN_V_FSP_Authenticity" value="Authenticity"/>
   <Text id="TD_V_FSP_Authenticity" value="Authenticity parameters."/>
   <Text id="TN_V_FSP_Authenticity_1" value="FSP_Authenticity_1"/>
   <Text id="TD_V_FSP_Authenticity_1" value="&quot;A-Code&quot; from the upper level FSCP
system."/>
   <Text id="TN_V_FSP_Authenticity_2" value="FSP_Authenticity_2"/>
   <Text id="TD_V_FSP_Authenticity_2" value="Extended &quot;A-Code&quot; from the upper level
FSCP system."/>
   <Text id="TN_V_FSP_Port" value="FSP_Port"/>
   <Text id="TD_V_FSP_Port" value="PortNumber identifying the particular FS-Device."/>
   <Text id="TN_V_FSP_AuthentCRC" value="FSP_AuthentCRC"/>
   <Text id="TD_V_FSP_AuthentCRC" value="CRC-16 across authenticity parameters."/>
   <Text id="TN_V_FSP_Protocol" value="Protocol"/>
   <Text id="TD_V_FSP_Protocol" value="Protocol parameters."/>
   <Text id="TN_V_FSP_ProtVersion" value="FSP_ProtVersion"/>
   <Text id="TD_V_FSP_ProtVersion" value="Protocol version (1=current version)."/>
   <Text id="TN_V_FSP_ProtMode" value="FSP_ProtMode"/>
   <Text id="TD_V_FSP_ProtMode" value="Protocol mode (1=16 bit CRC, 2=32 bit CRC)"/>
   <!--
```

```
   <Text id="TN_SV_FSP_ProtMode_crc16" value="16 bit CRC"/>
   -->
   <Text id="TN_SV_FSP_ProtMode_crc32" value="32 bit CRC"/>
   <Text id="TN_V_FSP_Watchdog" value="FSP_Watchdog"/>
   <Text id="TD_V_FSP_Watchdog" value="Monitoring of IO update."/>
   <Text id="TN_V_FSP_IO_StructCRC" value="FSP_IO_StructCRC"/>
   <Text id="TD_V_FSP_IO_StructCRC" value="CRC-16 across IO structure description block."/>
   <Text id="TN_V_FSP_TechParCRC" value="FSP_TechParCRC"/>
   <Text id="TD_V_FSP_TechParCRC" value="Securing code across FST (technology specific
parameter)."/>
   <Text id="TN_V_FSP_ProtParCRC" value="FSP_ProtParCRC"/>
   <Text id="TD_V_FSP_ProtParCRC" value="CRC-16 across protocol parameters"/>
   <Text id="TN_V_FSP_ParamDescCRC" value="FSP_ParamDescCRC"/>
   <Text id="TD_V_FSP_ParamDescCRC" value="CRC-32 signature securing authenticity, protocol, and
FS I/O structure description within IODD."/>
   <Text id="TN_V_FSP_TimeToReady" value="Time To Ready"/>
   <Text id="TD_V_FSP_TimeToReady" value="The time that is required by the device after power on
in order to be ready for communication."/>
      <Text id="TN_V_FSP_MinShutDownTime" value="Minimum shut down time"/>
      <Text id="TD_V_FSP_MinShutDownTime" value="Minimum time to wait after shut down of FS-
Device."/>
   <Text id="TN_V_FSP_WCDT" value="Worst-case Delay Time"/>
   <Text id="TD_V_FSP_WCDT" value="Time from triggering an FS-Device (sensor) until the output
shows a corresponding signal change or Process Data change. For an FS-Device (actuator) it is
the time from signal change or Process Data change to the actuator's safe state."/>
   <Text id="TN_V_FSP_OFDT" value="One Fault Delay Time"/>
   <Text id="TD_V_FSP_OFDT" value="Time from occurrence of a fault within the FS-Device to a
corresponding signal change or Process Data change"/>
   <Text id="TN_EV_FSP_TransmissionError_CRCSignature" value="Transmission error (CRC
signature)"/>
   <Text id="TD_EV_FSP_TransmissionError_CRCSignature" value=""/>
   <Text id="TN_EV_FSP_TransmissionError_Counter" value="Transmission error (Counter)"/>
   <Text id="TD_EV_FSP_TransmissionError_Counter" value=""/>
   <Text id="TN_EV_FSP_TransmissionError_Timeout" value="Transmission error (Timeout)"/>
   <Text id="TD_EV_FSP_TransmissionError_Timeout" value=""/>
   <Text id="TN_EV_FSP_UnexpectedAuthenticationCode" value="Unexpected authentication code"/>
   <Text id="TD_EV_FSP_UnexpectedAuthenticationCode" value=""/>
   <Text id="TN_EV_FSP_UnexpectedAuthenticationPort" value="Unexpected authentication port"/>
   <Text id="TD_EV_FSP_UnexpectedAuthenticationPort" value=""/>
   <Text id="TN_EV_FSP_Incorrect_AuthentCRC" value="Incorrect FSP_AuthentCRC"/>
   <Text id="TD_EV_FSP_Incorrect_AuthentCRC" value=""/>
   <Text id="TN_EV_FSP_Incorrect_ProtParCRC" value="Incorrect FSP_ProtParCRC"/>
   <Text id="TD_EV_FSP_Incorrect_ProtParCRC" value=""/>
   <Text id="TN_EV_FSP_Incorrect_TechParCRC" value="Incorrect FSP_TechParCRC"/>
   <Text id="TD_EV_FSP_Incorrect_TechParCRC" value=""/>
   <Text id="TN_EV_FSP_Incorrect_IO_StructCRC" value="Incorrect FSP_IO_StructCRC"/>
   <Text id="TD_EV_FSP_Incorrect_IO_StructCRC" value=""/>
   <Text id="TN_EV_FSP_WatchdogTimeOutOfSpec" value="Watchdog time out of specification"/>
   <Text id="TD_EV_FSP_WatchdogTimeOutOfSpec" value=""/>
   <Text id="TN_EV_FSP_NoFSVerifyRecord" value="No FS_VerifyRecord received"/>
   <Text id="TD_EV_FSP_NoFSVerifyRecord" value=""/>
   <Text id="TN_M_Param_NonSafety" value="Non-Safety (Standard) Parameter"/>
   <Text id="TN_M_FST_Param" value="FS Technology Parameter"/>
   <Text id="TN_M_FSP_Param" value="FS Protocol Parameter"/>
   <Text id="TN_M_FSP_Param_Aux" value="FS Auxiliary Parameter"/>
   <Text id="TN_M_CP_Diag_DeviceStatusInfo" value="Device Status Information"/>
   <Text id="TN_M_CP_Diag_ServiceFunctions" value="Service Functions"/>
   <Text id="TN_M_CP_Param_GeneralSettings" value="General Settings"/>
  </PrimaryLanguage>
 </ExternalTextCollection>
 <Stamp crc="2515840685"><Checker name="IODD-Checker V1.1.7" version="V1.1.7.0"/></Stamp>
</IODevice>
```

Annex F
(normative)

Device tool Interface (DTI) for SDCI

F.1 Purpose of DTI

For integration of SDCI Devices in a Master tool, IODD files shall be used provided by the Device manufacturer. Syntax and semantics of these files are standardized (see [16]) such that the Devices can be integrated independently from the vendor/manufacturer.

However, some applications/standards such as functional safety require a so-called Dedicated Tool for e.g. parameter setting and validation, at least as a complement to the IODD method. This Dedicated Tool shall communicate with its Device and is responsible for the data integrity according to IEC 62061 and ISO 13849-1. In the following, the term "Device tool" is used within this document. Without any additional standardized technology, such an SDCI system would force the user

- to know which Device tool is required for a particular Device,
- to enter the communication parameters of the Device both in the Master tool and in the Device tool and to keep the parameters consistent,
- to store consistent configuration and parameterization data from both the Master tool and the Device tool at one single place to archive project data.

In addition, it would face the Device manufacturer

- with the necessity to implement the communication functionality for each supported field bus system, and
- with the problem of nested communication whenever the target Device is located in a different network and only a proprietary gateway interconnects the networks.

A solution is the Device Tool Interface (DTI) technology specified herein after. It can be used for safety (FS-Master/FS-Device) as well as for non-safety (Master/Device) SDCI devices.

F.2 Base model

The Device Tool Interface (DTI) comprises three main parts according to Figure 61:

- An invocation interface between Master tool and Device tool;
- A backward interface between Master tool and Device tool ("Backchannel");
- A communication interface between Device tool and a Communication Server.

The combination of these three parts leads to the following user interaction.

A Master tool is supposed to be already installed on a PC running Microsoft Windows operating system. A Device is configured with the help of the corresponding IODD file of the Device manufacturer. This step includes assignment of Port addresses and adjustment of the Device parameters defined in the IODD.

Now, DTI allows for associating Device tool identification with SDCI Device identification. The Master tool uses DTI specific mechanisms to find the Device tool for a given Device. It provides for example in the context menu of a selected Device an entry that can be used to invoke the Device tool. As soon as the Device tool is active, it identifies the selected Device. The user can instantly establish a communication with the Device without entering address information and alike and assign parameter values. Assigned values can be returned to the Master tool using the Backchannel.

For the communication server part, DTI relies on technology specified in [18]. DTI comprises mechanisms to store and maintain Device data objects (project data).

F.3 Invocation interface

F.3.1 Overview

The invocation interface is used to transfer information from the representation of the Device in the Master tool to the Device tool. In order to achieve a high flexibility and to be able to identify different versions of the interface, both the description of the Device tool capabilities and the invocation parameters are stored in XML based documents. For the assignment from Master tool to Device tool the system registry of the Microsoft Windows operating system is used.

Figure F.1 shows the principle of the DTI invocation interface part.

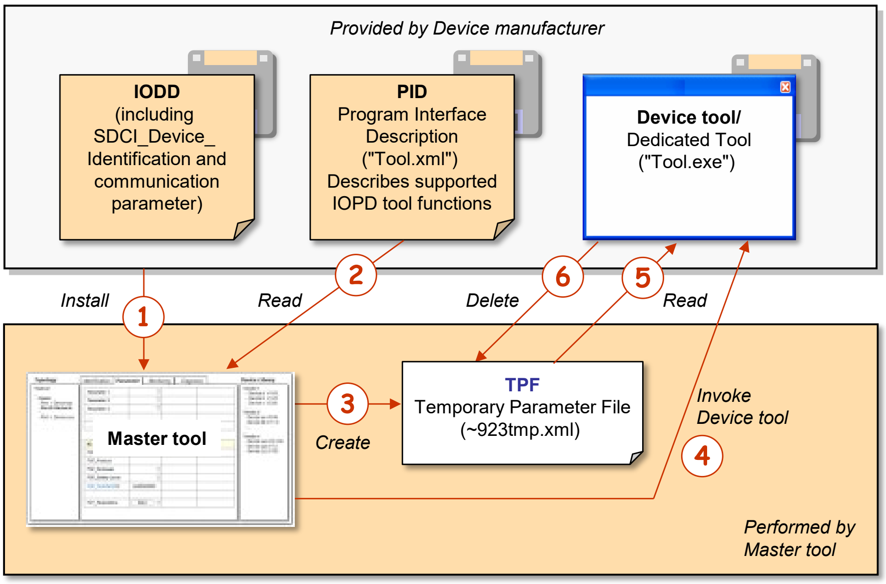


Figure F.1 – Principle of DTI invocation interface

Precondition for the mechanism is the availability of the Master tool and all used Device tools on one and the same PC.

For the tool invocation the following steps are required:

(1) As usual, the IODD file is imported into the Master tool. The Device is configured, and communication settings are made. With the help of (SDCI) Device Identification data the Master tool can find the installed Device tool and the directory path to the "Program Interface Description" (PID) file. Annex F.3.2 describes this procedure in detail.

(2) The Master tool reads the content of the PID file. This file contains information about the interface version and the supported tool functions. The structure of the PID file is described in Annex F.3.3.

(3) Before launching the Device tool, the Master tool creates a new "Temporary Parameter File" (TPF) that contains all invocation parameters. See F.3.4 for details.

(4) The Master tool launches the Device tool and passes the name of the TPF. See F.3.4.

(5) The Device tool reads and interprets the content of the TPF file.

(6) The Device tool deletes the TPF file after processing. See F.3.4.

F.3.2 Detection of Device tool

F.3.2.1 Registry structure

For DTI to identify the type of an SDCI Device, a specific, unique, and unambiguous SDCI "Device_Identifier" is used in the PC system registry and within the Temporary Parameter File (TPF).

Figure F.2 shows the structure of the DTI part of the registry. Each class in the diagram represents a registry key. Each attribute in the diagram represents a string value of the registry key. The semantics of the attributes is defined in Table F.1 and Table F.2.

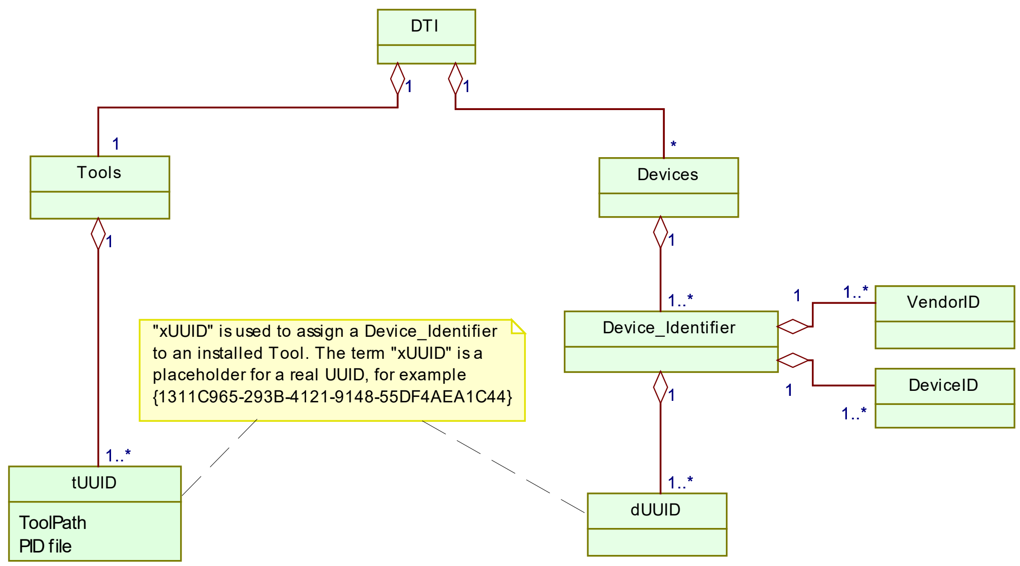


Figure F.2 – Structure of the registry

Since for an SDCI Device_Identifier an unlimited number of "UUID" elements can be inserted, the Master tool shall handle all tools of these "UUID" elements.

F.3.2.2 Device tool specific registry entries

Each version of a Device tool is represented by one UUID in the system registry.

The installation program of a Device tool (32 bit or 64 bit) shall insert this UUID as key under its appropriate registry path:

HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link Community\DTI\Tools or

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IO-Link Community\DTI\Tools

A Master tool shall check both registry paths.

Within this key, two attributes with string values shall be used:

- "PID-file", containing the absolute path and name of the installed PID file, and
- "AppPath", containing the absolute path and name of the executable Device tool file including its file extension (.exe).

Figure F.3 illustrates registry entries for Devices and tools.

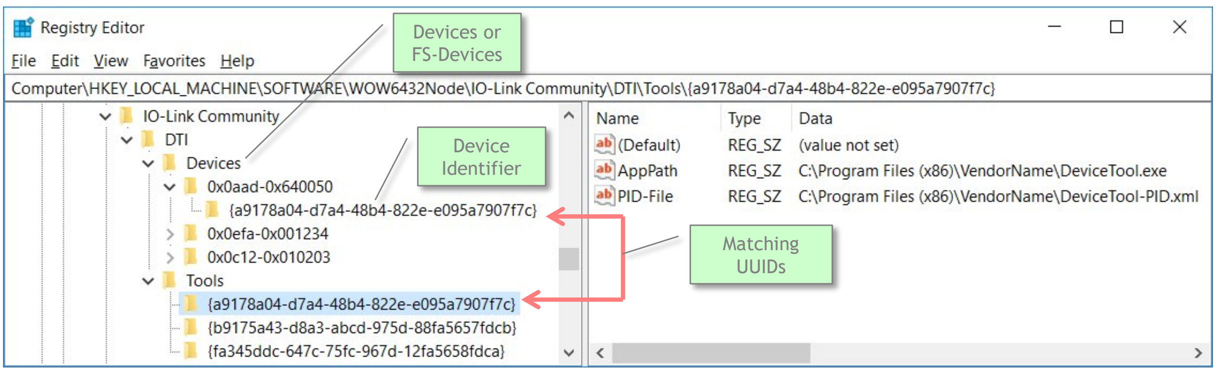


Figure F.3 – Example of a DTI registry

If different versions of a Device tool for the same Device type exist (same Device_Identifier), each version requires a separate UUID in the registry. In the PID files of the Device tools, different version information shall be provided in the attribute "ToolDescription" of the element

"ToolDescription" (see Table F.1). This leads to multiple items in the context menu of the Master tool, differing in the description text.

NOTE The advantage of a separate entry of the “ToolPath” keyword is a simpler installation procedure for the Device tool. It can install the PID file without a need to modify this file.

The installation program of a Device tool shall also insert each UUID as key under the registry path

HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link Community\DTI\SDCI Devices\<Device Identifier>

Devices are identified unambiguously via the following items:

- VendorID (assigned by SDCI support organization, see Annex I);
- DeviceID (assigned by Device/FS-Device manufacturer).

This information is part of the IO Device Description (IODD), which allows the Master tool to work with the Device (data, parameter) without establishing an online connection to the Device. The IDs can be found at the following locations within an IODD:

a) //ISO15745Profile/ProfileBody/DeviceIdentity/@vendorId;
b) //ISO15745Profile/ProfileBody/DeviceIdentity/@deviceId.

With the help of the registry, the Master tool can read the required information about the Device tool (in case of safety: Dedicated Tool). Location and structure for the entries shall be commonly agreed upon.

All entries shall be provided by the Device tool under the following registry path:

HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link Community\DTI\Devices

Within this path one or more keys can be inserted with the following field structure:

0xvvvv-0xdddddd

The meaning of the fields is:

vvvv: Four-character VendorID in hexadecimal coding

dddddd: Six-character DeviceID in hexadecimal coding.

The question mark character "?" can be used in the DeviceID as wildcard to replace one single character. The number of question marks is only limited by the size of the field. If wildcards are used, the Device tool is responsible for the check whether it supports the selected object.

The assignment to the tool is made by a string value within this key. The UUID shall be used as name for the string value. The number of string values is not limited, which in turn means an unlimited number of tools that can be assigned to the same Device.

Examples for valid keys (see Figure F.3):

0x0A99-0x00880D	The tool can be launched in the context of a Device with a DeviceID 0x00880D from the vendor with the VendorID 0x0A99.
0x0F3B-0x002B??	The tool can be started in the context of Devices with a DeviceID in the range of 0x002B00 to 0x002BFF from the vendor with the VendorID 0x0F3B.

F.3.2.3 Processing of the Registry Data

The installation program of the Device tool is responsible to insert the keys in the system registry as defined in Annex F.3.2.2.

Figure F.4 shows an activity diagram illustrating the detection of a Device tool in the registry via SDCI "Device_Identifier".

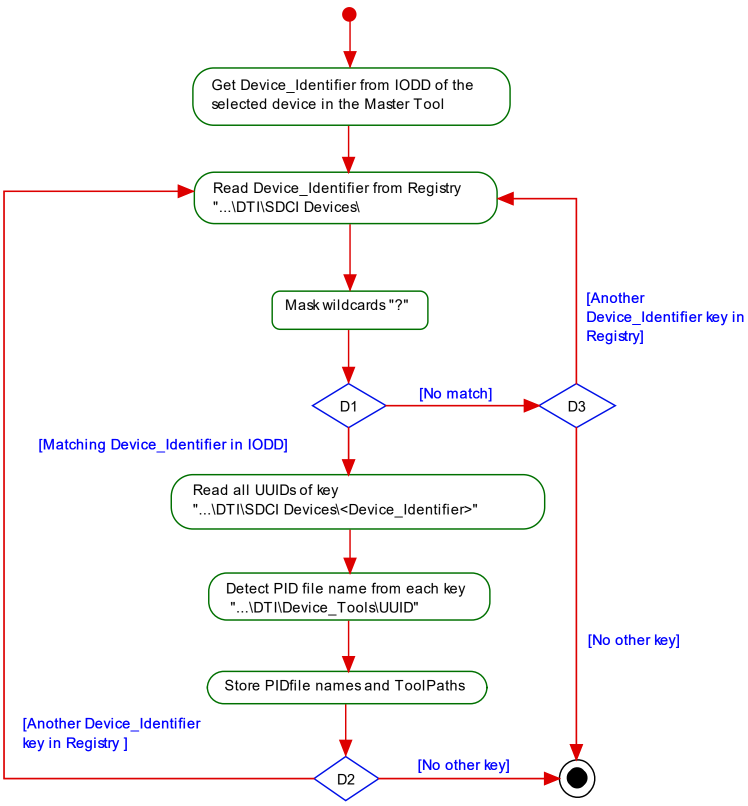


Figure F.4 – Detection of a Device tool in registry

NOTE All registry keys in Figure F.4 are relative to the path HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link Community.

In a first step, the Master tool gets the Device_Identifier from the IODD of the selected object in the Master tool. Then all sub keys in the system registry path ...DTI\SDCI Devices shall be compared with this Device_Identifier. If a sub key matches (excepting wildcards), the UUID sub key of this key is used to find the PID file name in the registry path DTI\Device Tools\<UUID>. Since the same PID file name can be found in different locations in the registry, the context menu of the Master tool shall only show the Device tools with different PID file names. As a last step, the information in the PID file is used to build the menu items of the Master tool (Figure F.5).

F.3.3 Program Interface Description – PID

F.3.3.1 General

The Program Interface Description (PID) file describes the properties of the Device tool and contains data which are required by the Master tool to build menu items in its graphical user interface (GUI). The PID file is an XML document. The corresponding XML schema is defined in F.9.2. UTF-8 shall be used for character encoding.

This PID file shall be provided by the manufacturer of a Device/Device tool and installed by the installation program associated with the Device tool. This installation program shall also insert the name and installation path in the system registry (see F.3.2).

The PID file allows the Master tool to extend its GUI menu structure by the name of the Device tool such that the user is able to launch the Device tool for example from the context menu of a selected Device as illustrated exemplary in Figure F.5.

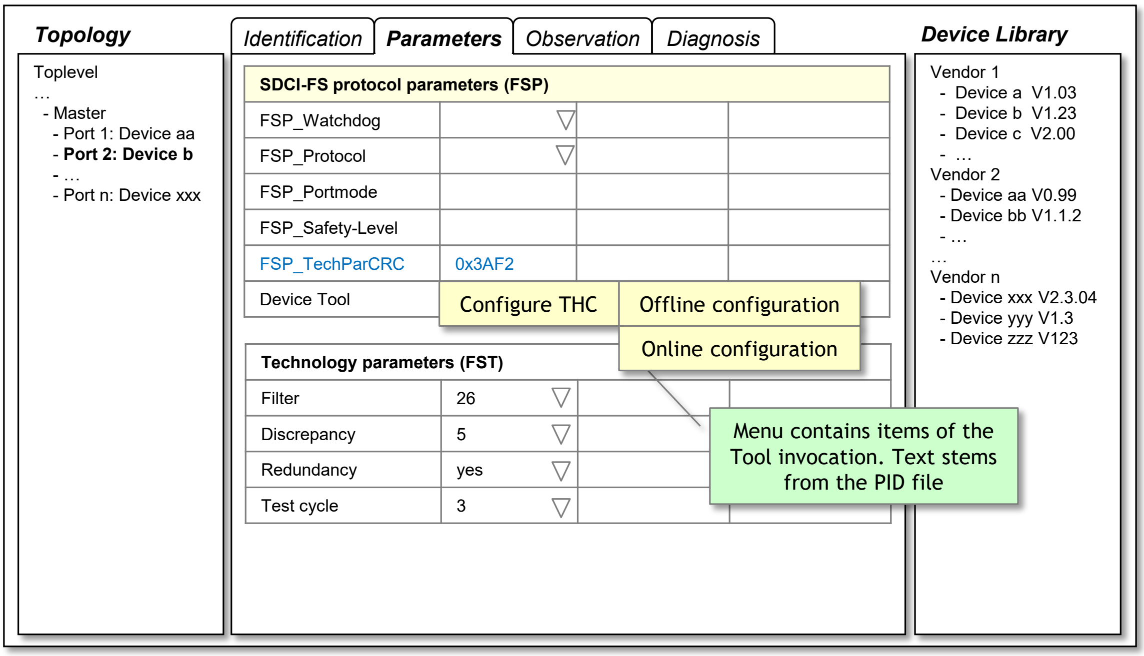


Figure F.5 – Menu for Device tool invocation

F.3.3.2 Structure of the PID file

The PID file is an XML based document. Its XML schema is shown in Figure F.13. Namespace URI for this file is http://www.io-link.com/DTI/2017/02/PID (see F.9.2). Hint: The schema file for "primitives" shall be within the same file folder than the main schema file (see F.9.6).

The elements of Figure F.13 are specified in Table F.1. The column "SV" indicates the schema version a particular attribute has been introduced.

Table F.1 – Description of PID file elements

Element	Attribute	Type	M/O	SV	Description
ProgramInterface	–	–	M	1.0	Root element
DocumentInfo	–	–	M	1.0	Meta information about the PID file
	Version	xsd:string	M	1.0	Contains the schema version of PID interface definition. Also determines the newest TPF version supported by this tool. The value shall comply with the following regular expression: \d+(\.\d+)* In this version, the string "1.1" shall be used.
General	–	–	M	1.0	General information about the Device tool
	VendorName	xsd:string	M	1.0	Contains the name of the Device vendor
ToolDescription	–	–	1..n	1.0	Is used to define language dependent text information for description of the Device tool. A ToolDescription element in English language is mandatory.
	xml:lang	xsd:language	M	1.0	Defines the language of the text. The "2-letter coding" or the "3-letter coding" as defined in ISO 639 shall be used.
	ToolName	xsd:string	M	1.0	Describes the function of the Device tool. This text shall be used to

Element	Attribute	Type	M/O	SV	Description
					extend the GUI menu items of the Master tool. Default element in English language shall always be present.
	ToolDescription	xsd:string	O	1.0	Contains a short description of the Device tool.
Invocation Prefix	–	–	–	1.0	With this element, the command line arguments of the called Device tool can be modified. If a Device tool is able to interpret different command line arguments, usually a prefix is used to define the semantic of an argument. If an InvocationPrefix is present in the PID file, the Master tool shall insert a blank character as delimiter between the InvocationPrefix string and the file name of the TPF. To interpret the command line argument as a file name for a DTI call, a Device tool shall be launched as follows: *DeviceTool.exe -i "c:\tmp\TPF01.xml"* In this case, the prefix "-i" shall be entered in the PID file.
	Name	xsd:string	O	1.0	Defines which command line prefix is used when the tool is launched. If this attribute is not present, only the file name of the TPF is used as command line argument. NOTE Since the datatype "string" is used, blank characters (ASCII 32 dec) are allowed. XML Entities are allowed and shall be converted by the Master tool.
ConformanceClass	–	–	M	1.0	Describes the conformance class of the engineering system.
	Name	xsd:string	M	1.0	Contains the name of the conformance class (F.8.1). One of the following values is allowed: "C1", "C2", or "C3"
EntryPoints	–	–	M	1.0	This optional element shall be used, if a Device tool has more than one entry point.
EntryPoint	–	–	1 to n	1.0	This element represents an entry point of the Device tool. Entry points are used to generate additional sub menu items in the "ToolDescription" context menu of the Master tool. Using entry points a Device tool can provide direct access to tool specific views or functions.
	ID	xsd:string	M	1.0	The attribute "ID" identifies an EntryPoint. It shall be unique within a PID file.
InfoText	–	–	1 to n	1.0	The element "InfoText" is used to define language dependent text information for description of the entry point. This information can be used to extend the GUI menu items of the Master tool. An InfoText element in English language shall always be present here.
	xml:lang	xsd:string	M	1.0	Defines the language of the text. The "2-letter coding" or the "3-letter

Element	Attribute	Type	M/O	SV	Description
					coding" as defined in ISO 639 shall be used.
	EntryPointName	xsd:string	M	1.0	Describes the function of the entry point. This text shall be used to extend the GUI menu items of the Master tool.
	EntryPointDescription	xsd:string	O	1.0	Contains a short description of the entry point.

F.3.3.3 Example PID file

The following XML code shows an example content of a PID file with EntryPoints.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProgramInterface xmlns="http://www.io-link.com/DTI/2017/02/PID"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
link.com/DTI/2017/02/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2017/02/PID
iosafe_pid_schema_20170225.xsd">
 <DocumentInfo version="V1.0"/>
 <General vendorName="IO-Link Community">
  <ToolDescription name="Configure THC" description="IO-Link Safety Device" lang="en"/>
  <ToolDescription name="Konfiguriere THC" description="IO-Link Safety Device" lang="de"/>
  <InvocationPrefix name="/"/>
 </General>
 <EntryPoints>
  <EntryPoint id="1">
   <InfoText name="Offline Configuration" description="Offline Configuration" lang="en"/>
   <InfoText name="Offline Konfiguration" description="Offline Konfiguration" lang="de"/>
  </EntryPoint>
  <EntryPoint id="2">
   <InfoText name="Online Configuration" description="Online Configuration" lang="en"/>
   <InfoText name="Online Konfiguration" description="Online Konfiguration" lang="de"/>
  </EntryPoint>
 </EntryPoints>
 <ConformanceClass name="C3"/>
</ProgramInterface>
```

F.3.4 Temporary Parameter File – TPF

F.3.4.1 General

Due to the large number of parameters to be transferred from the Master tool to the Device tool, a parameter transfer by command line arguments is not a good solution. The necessary syntax would become too complex to cover all aspects.

Instead, all required parameters are included into an XML file, called Temporary Parameter File (TPF) by the Master tool and thus, the name of the XML file is passed as the only command line argument. If the Device tool requires a command line switch, this information can be extracted from the PID file. See "InvocationPrefix" in Table F.1 for details.

The XML schema for the TPF is defined in F.9.3. For character encoding, UTF-8 shall be used. The Master tool shall use the newest TPF schema version supported by both the Master tool and the Device tool.

After the TPF is interpreted, the Device tool shall delete the TPF file.

F.3.4.2 Structure of a TPF

The structure of the TPF is defined by the XML schema shown in Figure F.14. This schema is built in a generic manner, which means, a new parameter does not require the schema itself to be updated. Thus, new parameters can be introduced without a new definition of the TPF structure.

Namespace URI for this file is http://www.io-link.com/DTI/2017/02/TPF (see F.9.3). The elements of the XML schema in Figure F.14 are specified in Table F.2. The column "SV" indicates the schema version a particular attribute has been introduced.

Table F.2 – Elements of a TPF

Element	Attribute	Type	M/O	SV	Description
InvocationInterface	–	–	M	1.0	Root element
General	–	–	M	1.0	General information about the TPF file
	schemaPath	xsd:string	M	1.0	This attribute defines the path where the schema files for FDT communication schemas and TPF/PID file are stored. • This schema files shall be installed on this path by the Master tool • The path does not change during runtime of the Master tool • The path can be used from a Device tool to initialize the XML parser. NOTE Even if no schema validation is used, some XML parsers need the location of the schema files for initialization. In this case, a Device tool does not need to install an own set of schema files – it should use the schema files in the path defined by this attribute.
	projectRelatedPath	xsd:string	M	1.0	The attribute "ProjectRelatedPath" contains information about a directory which is assigned to the project context of the Master tool. A Device tool should use this path for storage of its Device data. The format and structure of this data is defined by the Device tool itself. Within this directory, additional subdirectories can be created. The Master tool is responsible to keep all data in the directory tree in its project context. That means, if the project is copied or archived, also this data shall be copied or archived. The attribute "ProjectRelatedPath" contains a unique path (directory) for each combination of Master project and DTI Device tool. For example, different directories are used for the same tool, if two Master tool projects are used. The file name in "ProjectRelatedPath" shall consist of the drive letter and an absolute path expression. Alternatively, the UNC notation can be used instead of the drive letter.
	portName	xsd:string	M	1.0	Name of used FS-Master Port
	portId	xsd:string	M	1.0	ID of used FS-Master Port 1 to n
	masterName	xsd:string	M	1.0	User defined name of FS-Master
	displayNameES	xsd:string	M	1.0	Display name of the Master tool in the language specified in attribute "currentLanguage". The Device tool can use this name in error messages or user dialogs to provide more understandable texts.

Element	Attribute	Type	M/O	SV	Description
	currentLanguage	xsd:string	M	1.0	Defines which language shall be used by the Device tool for TPF. The "2-letter coding" or the "3-letter coding" as defined in ISO 639 can be used. If a Device tool does not support the selected language, the tool shall use its default language.
	commServerProgID	xsd:string	O	1.0	This attribute contains the ProgID of the Communication Server provided by the Master tool manufacturer. It allows the Device tool to use the Communication Server functionality. See F.5.6 for details. If this attribute is not provided, the Master tool does not support a Communication Server.
	busCategory	xsd:string	M	1.1	This attribute is used to specify the used communication protocol. It also can be used to find a corresponding Communication Server. Default value is "2C4CD8B8-D509-4ECB-94A7-019F12569C8B"
	selectedEntryPoint	xsd:string	O	1.0	Defines, which entry point of the Device tool was selected in the Master tool when the Device tool was launched. This attribute shall contain only values defined in the attribute "ID" of any element "EntryPoint" of the corresponding PID file. This attribute allows the Device tool to show an entry point specific GUI when it was launched. If the PID file does not contain any EntryPoint elements, this attribute shall not be used in the TPF.
	conformanceClass	xsd:string	M	1.0	Contains the name of the conformance class of the Master tool. One of the following values is allowed: "C2" or "C3". See Table F.8.
DeviceItem	–	–	M	1.0	Used to describe the Device selected in the ES.
	vendorId	xsd:string	M	1.0	See IEC 61131-9
	productId	xsd:string	M	1.0	See IEC 61131-9
	deviceId	xsd:string	M	1.0	See IEC 61131-9
	usedConfigFileCRC	xsd:string	M	1.0	IODD stamp
	usedConfigFile	xsd:string	M	1.0	The keyword usedConfigFile contains the file name of the used description file (e.g. IODD). The file name shall consist of the drive letter, an absolute path expression and the file extension. Alternatively, the UNC notation can be used instead of the drive letter. The Device tool It is not allowed to modify the content of the description file.
	reference	xsd:string	M	1.0	Used to identify FS-Device within engineering project

Element	Attribute	Type	M/O	SV	Description
	commReference	xsd:string	M	1.0	This attribute is used with the Communication Server (CS) to address a Device instance unambiguously within the PC. The unique nature of this attribute shall be ensured by the Master tool. The structure of the attribute is only defined by the Master tool. It is not allowed to interpret the syntax of this keyword in the Device tool. LineFeed characters (ASCII 10 dec) are not allowed in the string. This attribute shall be provided for all Device instances of a TPF, if the Device tool wants to use the CS interface (Conformance Class 3 (C3)) and the commReference is different from the DeviceReference.
VariableInstanceData	–	–	M	1.0	Element "VariableInstanceData" is a container for "Variable" elements (= parameter).
Variable	–	–	1 to n	1.0	Contains information about a variable.
	variableId	xsd:string	M	1.0	Contains the parameter ID
Item	–	–	1 to n	1.0	Contains information about a Subindex of a variable.
	subindex	xsd:string	M	1.0	See IEC 61131-9
	value	xsd:string	M	1.0	Contains the parameter value. In absence of a parameter-specific rule for the representation of the value: Numerical values shall use the decimal coding without left-hand zeros. Negative values shall have a hyphen (ASCII 45 dec) prefix. Separator for floating point values is a dot (ASCII 46 dec). Other separators are not permitted.
	state	xsd:string	M	1.0	Contains parameter status
	error	xsd:string	M	1.0	Contains parameter error

F.3.4.3 Example of a TPF

The following XML code shows the content of an exemplary TPF file.

```
<?xml version="1.0" encoding="utf-8"?>
<InvocationInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.io-
link.com/DTI/2017/02/TPF" xsischemaLocation="http://www.io-link.com/DTI/2017/02/TPF
IOsafe_TPF_Schema_20170225.xsd" xmlns:prim="http://www.io-link.com/DTI/2017/02/Primitives">
    <General currentLanguage="en" commServerProgID="" />
    <VariableInstanceData>
        <Variable variableID="V_VendorName">
            <Item subindex="0" state="initial" error="0" value="IO-Link Community" />
        </Variable>
        <Variable variableID="V_VendorText">
            <Item subindex="0" state="initial" error="0" value="http://www.io-link.com" />
        </Variable>
        <Variable variableID="V_ProductName">
            <Item subindex="0" state="initial" error="0" value="IO-Link Safety Device" />
        </Variable>
        <Variable variableID="V_ProductText">
            <Item subindex="0" state="initial" error="0" value="Sample IODD for a device with
IO-Link Safety" />
        </Variable>
        <Variable variableID="V_ProductID">
```

```xml
        <Item subindex="0" state="initial" error="0" value="Safety-01" />
    </Variable>
    <Variable variableID="V_SerialNumber">
        <Item subindex="0" state="empty" error="0" value="" />
    </Variable>
    <Variable variableID="V_HardwareRevision">
        <Item subindex="0" state="empty" error="0" value="" />
    </Variable>
    <Variable variableID="V_FirmwareRevision">
        <Item subindex="0" state="empty" error="0" value="" />
    </Variable>
    <Variable variableID="V_ApplicationSpecificTag">
        <Item subindex="0" state="database" error="0" value="***" />
    </Variable>
    <Variable variableID="V_CP_FunctionTag">
        <Item subindex="0" state="database" error="0" value="***" />
    </Variable>
    <Variable variableID="V_CP_LocationTag">
        <Item subindex="0" state="database" error="0" value="***" />
    </Variable>
    <Variable variableID="V_ProcessDataInput">
        <Item subindex="1" state="empty" error="0" value="" />
        <Item subindex="2" state="empty" error="0" value="" />
        <Item subindex="3" state="empty" error="0" value="" />
        <Item subindex="4" state="empty" error="0" value="" />
        <Item subindex="5" state="empty" error="0" value="" />
        <Item subindex="6" state="empty" error="0" value="" />
        <Item subindex="7" state="empty" error="0" value="" />
        <Item subindex="8" state="empty" error="0" value="" />
        <Item subindex="9" state="empty" error="0" value="" />
        <Item subindex="10" state="empty" error="0" value="" />
        <Item subindex="11" state="empty" error="0" value="" />
        <Item subindex="12" state="empty" error="0" value="" />
        <Item subindex="13" state="empty" error="0" value="" />
        <Item subindex="14" state="empty" error="0" value="" />
        <Item subindex="127" state="empty" error="0" value="" />
        <Item subindex="128" state="empty" error="0" value="" />
    </Variable>
    <Variable variableID="V_ProcessDataOutput">
        <Item subindex="127" state="empty" error="0" value="" />
    </Variable>
    <Variable variableID="V_NonSafetyParameter">
        <Item subindex="0" state="initial" error="0" value="123" />
    </Variable>
    <Variable variableID="V_FST_DiscrepancyTime">
        <Item subindex="0" state="database" error="0" value="5" />
    </Variable>
    <Variable variableID="V_FST_Filter">
        <Item subindex="0" state="database" error="0" value="5" />
    </Variable>
    <Variable variableID="V_FSP_Authenticity">
        <Item subindex="1" state="database" error="0" value="123" />
        <Item subindex="2" state="database" error="0" value="0" />
        <Item subindex="3" state="database" error="0" value="1" />
        <Item subindex="4" state="database" error="0" value="0" />
    </Variable>
    <Variable variableID="V_FSP_Protocol">
        <Item subindex="1" state="database" error="0" value="1" />
        <Item subindex="2" state="database" error="0" value="2" />
        <Item subindex="3" state="database" error="0" value="100" />
        <Item subindex="4" state="database" error="0" value="39464" />
        <Item subindex="5" state="database" error="0" value="23746" />
        <Item subindex="6" state="database" error="0" value="0" />
    </Variable>
    <Variable variableID="V_FSP_TimeToReady">
        <Item subindex="0" state="initial" error="0" value="10" />
    </Variable>
    <Variable variableID="V_FSP_MinShutDownTime">
        <Item subindex="0" state="initial" error="0" value="100" />
    </Variable>
    <Variable variableID="V_FSP_WCDT">
        <Item subindex="0" state="initial" error="0" value="10" />
    </Variable>
    <Variable variableID="V_FSP_OFDT">
        <Item subindex="0" state="initial" error="0" value="10" />
    </Variable>
    <Variable variableID="V_DeviceStatus">
        <Item subindex="0" state="initial" error="0" value="0" />
```

```xml
        </Variable>
        <Variable variableID="V_DetailedDeviceStatus">
            <Item subindex="1" state="empty" error="0" value="" />
            <Item subindex="2" state="empty" error="0" value="" />
            <Item subindex="3" state="empty" error="0" value="" />
            <Item subindex="4" state="empty" error="0" value="" />
        </Variable>
    </VariableInstanceData>
</InvocationInterface>
```

F.3.5 Temporary Backchannel File – TBF

F.3.5.1 General

The TBF should be transfered by a new transaction of the communication server. This transaction is initiated by the Device tool and can be performed automatically or upon user request. Transaction acknowledgements (TAF) should be implemented indicating reception of the instance values by the Master tool or indicating a transaction fault (see F.3.6).

F.3.5.2 Structure of the TBF

The structure of the TBF is defined by the XML schema Figure F.15. This schema is built in a generic manner, which means, a new parameter does not require the schema itself to be updated. Thus, new parameters can be introduced without a new definition of the TBF structure.

Namespace URI for this file is http://www.io-link.com/DTI/2017/02/TBF (see F.9.4). The elements of Figure F.15 are specified in Table F.3. The column "SV" indicates the schema version a particular attribute has been introduced.

Table F.3 – Elements of the TBF

Element	Attribute	Type	M/O	SV	Description
ReturnInterfaceRequest	–	–	M	1.0	Root element
VariableInstanceData	–	–	M	1.0	The element "VariableInstanceData" is a container for "Variable" elements (= parameter).
Variable	–	–	1 to n	1.0	Contains information about a variable.
	variableId	xsd:string	M	1.0	Contains the parameter ID
Item	–	–	1 to n	1.0	Contains information about a Subindex of a variable.
	subindex	xsd:string	M	1.0	See IEC 61131-9
	value	xsd:string	M	1.0	Contains the parameter value. In absence of a parameter-specific rule for the representation of the value: Numerical values shall use the decimal coding without left-hand zeros. Negative values shall have a hyphen (ASCII 45 dec) prefix. Separator for floating point values is a dot (ASCII 46 dec). Other separators are not permitted.
	state	xsd:string	M	1.0	Contains parameter status
	error	xsd:string	M	1.0	Contains parameter error

F.3.5.3 Example of a TBF

The following XML code shows the content of an exemplary TBF file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ReturnInterfaceRequest xmlns="http://www.io-link.com/DTI/2017/02/TBF"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
link.com/DTI/2017/02/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2017/02/TBF
IOsafe_TBF_Schema_20170225.xsd">
    <VariableInstanceData>
```

```
        <Variable variableID="V_ApplicationSpecificTag">
            <Item subindex="0" state="database" error="0" value="***" />
        </Variable>
        <Variable variableID="V_CP_FunctionTag">
            <Item subindex="0" state="database" error="0" value="***" />
        </Variable>
        <Variable variableID="V_CP_LocationTag">
            <Item subindex="0" state="database" error="0" value="***" />
        </Variable>
        <Variable variableID="V_NonSafetyParameter">
            <Item subindex="0" state="initial" error="0" value="123" />
        </Variable>
        <Variable variableID="V_FST_DiscrepancyTime">
            <Item subindex="0" state="database" error="0" value="6" />
        </Variable>
        <Variable variableID="V_FST_Filter">
            <Item subindex="0" state="database" error="0" value="6" />
        </Variable>
        <Variable variableID="V_FSP_Authenticity">
            <Item subindex="1" state="database" error="0" value="123" />
            <Item subindex="2" state="database" error="0" value="0" />
            <Item subindex="3" state="database" error="0" value="1" />
            <Item subindex="4" state="database" error="0" value="0" />
        </Variable>
        <Variable variableID="V_FSP_Protocol">
            <Item subindex="1" state="database" error="0" value="1" />
            <Item subindex="2" state="database" error="0" value="2" />
            <Item subindex="3" state="database" error="0" value="100" />
            <Item subindex="4" state="database" error="0" value="39464" />
            <Item subindex="5" state="database" error="0" value="592319" />
            <Item subindex="6" state="database" error="0" value="0" />
        </Variable>
    </VariableInstanceData>
</ReturnInterfaceRequest>
```

F.3.6 Temporary Acknowledgment File – TAF

F.3.6.1 General

Transaction acknowledgements should be implemented indicating reception of the instance values by the Master tool or indicating a transaction fault. The same mechanism is used as with TBF (see F.3.5).

F.3.6.2 Structure of the TAF

The structure of the TAF is defined by the XML schema shown in Figure F.15. However, the root name has changed to "ReturnInterfaceResponse".

Namespace URI for this file is http://www.io-link.com/DTI/2017/02/TBF (see F.9.4). The elements of Figure F.15 are specified in Table F.4. The column "SV" indicates the schema version a particular attribute has been introduced.

Table F.4 – Elements of the TAF

Element	Attribute	Type	M/O	SV	Description
ReturnInterfaceResponse	–	–	M	1.0	Root element
Response	–	–	M	1.0	Contains the response of the ES.
	value	xsd:boolean	M	1.0	Contains the indication of success

F.3.6.3 Example of a TAF

The following XML code shows the content of an exemplary TAF file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReturnInterfaceResponse xmlns="http://www.io-link.com/DTI/2017/02/TBF"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
link.com/DTI/2017/02/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2017/02/TBF
IOsafe_TBF_Schema_20170225.xsd">
  <Response value="true"/>
</ReturnInterfaceResponse>
```

F.3.7 Invocation behavior

F.3.7.1 Conventions on Device tool invocation

Since the directory path of the TPF can contain "blank" characters, the Device tool shall use the double quote character (") at the beginning and the end of the string when the ".exe" file is invoked.

It is not required for the invoking Master tool to monitor the status of the launched Device tools. Even in case an instance of a Device tool is already running, the Master tool will generate a new Device tool invocation whenever the user launches the same tool again.

Therefore, it is the task of the Device tool to handle multiple invocations. Table F.5 lists invocation cases and possible behaviors.

Table F.5 – Invocation cases and behaviors

Case	Behavior
Device tool is launched once	No conflicts
Device tool is already running and works on the same Device instance as in a prior session.	– The tool should be brought to the foreground of the GUI desktop – Invocation of another instance of the Device tool shall be avoided
Device tool is already running and works on another Device instance as provided by the DTI call. The provided DeviceReference is *known* in the Device tool.	The behavior depends on the design of the Device tool: – Another tool instance is launched and opens its Device data – The active GUI is brought to the foreground of the desktop in order to show the Device data of the selected Device
Device tool is already running and works on another Device instance as provided by the DTI call. The provided DeviceReference is *not known* in the Device tool.	The behavior depends on the design of the Device tool: – Another tool instance is launched and creates a new Device instance – The active GUI is brought to the foreground of the desktop in order to create a new Device instance of the selected Device

If a Device tool is invoked via DTI, this tool should not call another Device tool because the Communication Server cannot interconnect (no nested communication defined for a DTI Communication Server).

F.3.7.2 Handling of the TPF

The name of the TPF will be provided to the Device tool as a command line parameter. This name shall consist of a drive letter, an absolute path expression and the file extension. Alternatively, the UNC notation can be used instead of the drive letter. The Master tool is responsible to create the file and unlock it before the Device tool is invoked in such a manner that the Device tool has full access to the file. The file name itself is only temporary and a new file name is generated with each tool invocation.

After interpretation of the content of the TPF file, the Device tool shall delete this file. Since the Master tool can also delete this file when it is restarted, it is recommended for the Device tool to make a "private" copy of the file when the Device tool is launched.

F.4 Device data objects (DDO)

F.4.1 General

There are scenarios, where parameter instance data for a certain project are created off-site, for example via a "USB Master" tool enhanced for SDCI-FS, which can be mailed to a facility where they are imported into the engineering tool of the overall automation project. These data are called Device Data Objects (DDO). Figure F.6 shows such a scenario. In essence, DDO are used to export/import parameter instance data of Devices from one Master tool to the other.

A TPF usually contains already those instance data. Thus, in principle, a DDO consists of TPF data and the associated IODD for consistency reasons.

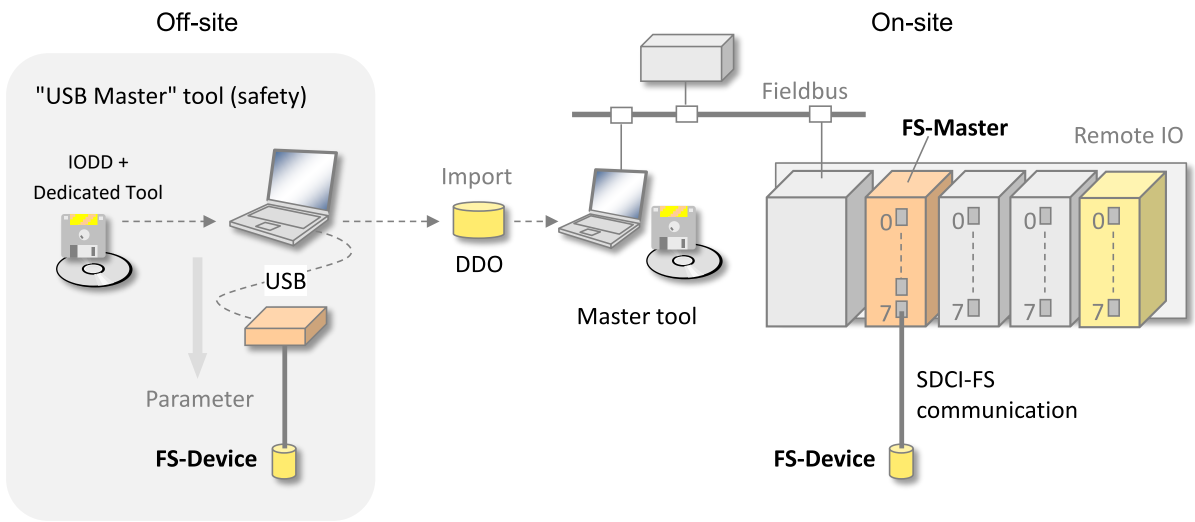


Figure F.6 – Purpose of Device data objects (DDO)

F.4.2 Structure of DDO package

A Master tool creates a DDO package as zip archive consisting of the TPF as defined in F.3.4 and the corresponding IODD together with its associated files, such as icons, pictures, and alike (see [16]). The archive file name extension shall be "iolddo". The name of the archive file can be chosen freely.

F.5 Communication Interface

F.5.1 General

As already explained in Clause F.1, there is no seamless communication solution for stand-alone Device tools such as "Dedicated Tools" for functional safety in SDCI so far. The only possibility in the past has been a separate point-to-point communication connection, for example RS232, USB, or alike, between a Device and a PC running the Device tool software. Each of these connections requires appropriate driver software with different programming API for the Device and for the different PC communication interfaces.

This leads to the problem that a Device tool either can work only with one particular communication interface or that the Device tool has to implement different APIs for Device driver integration.

Another problem in a plant is that the network structure often requires communication across network boundaries (Routing). Due to the many fieldbuses and different communication protocols, it is very cumbersome to achieve an integrated network with routing functions for Device tools down to the associated Device (see Figure F.7).

The second major part of DTI solves two problems:

- All Devices/FS-Devices and their Device tools/Dedicated Tools can rely on one particular communication interface.
- The chosen communication technology is standardized in [18] and solves the routing problem across network boundaries.

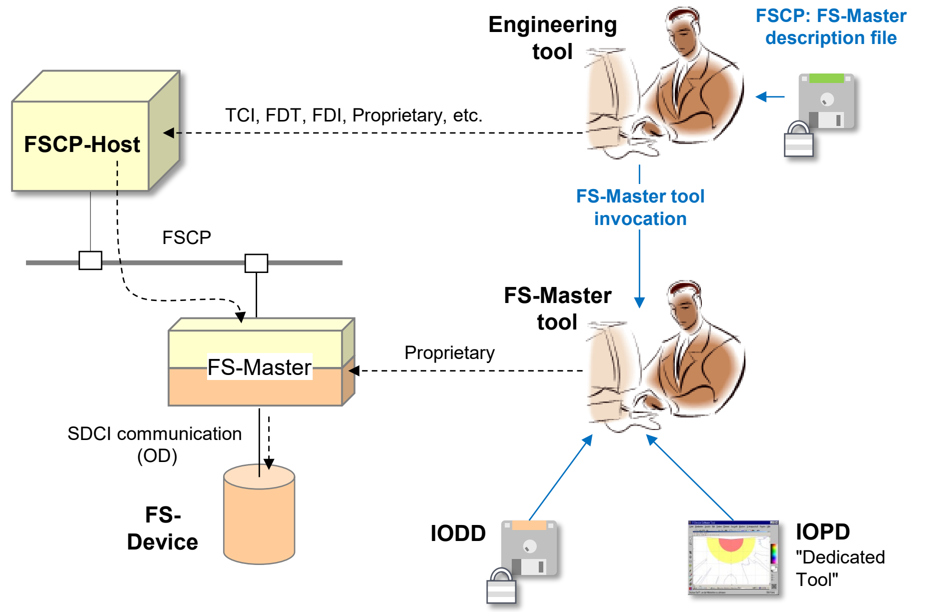


Figure F.7 – Communication routes between Device tool and Device

F.5.2 Principle of DTI communications

The communication interface consists of a component which provides a unique interface (API) to the Device tool. This component can provide communication functionality for different field busses and proprietary network protocols. The communication parameters which are necessary to establish a connection are entered in the Master tool and passed to the Device tool when it is launched.

Figure F.7 shows fieldbus or proprietary networks between the PC and the Device. Figure F.8 shows the mapping to software and Communication Servers. In this case, the Communication Server (Fieldbus) requires information about the network protocol. This routing information is generated by the Engineering System and transferred to the Communication Server (Fieldbus). Since manufacturer specific data is exchanged, the Communication Server and the Engineering system shall be provided by the same manufacturer.

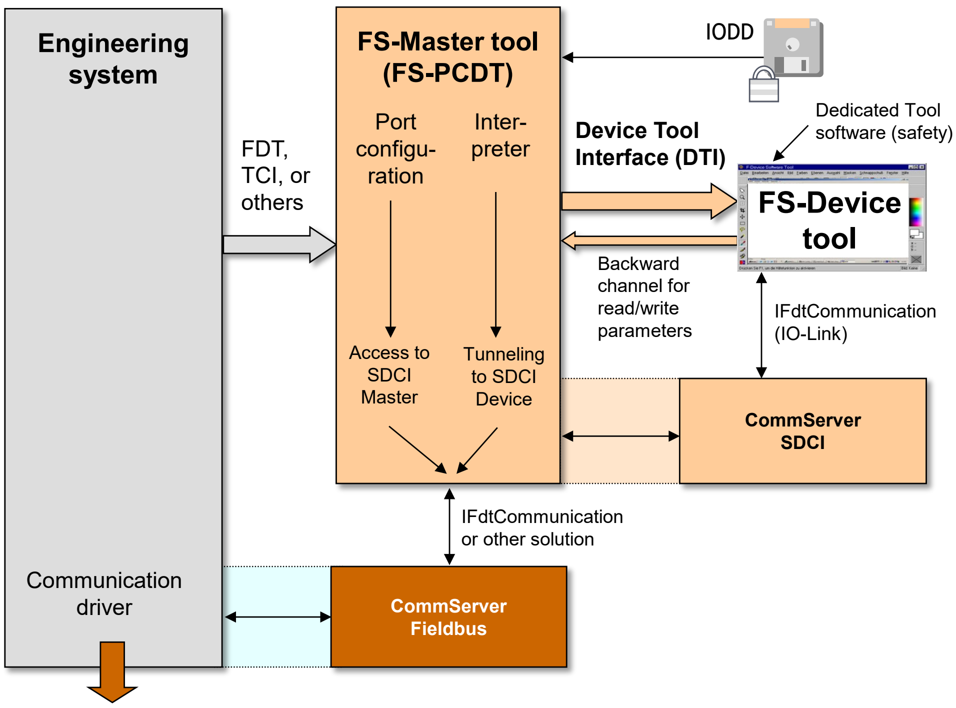


Figure F.8 – Routing across networks and SDCI

The routing information for the second Communication Server (SDCI) is generated by the Master tool and transferred to this CS. When the Device tool is started, only a communication reference to the Device is passed. This reference is forwarded from the Device tool to the Communication Server. With the help of the routing information from the Engineering system, the Communication Server can create physical network addresses and establish a connection to the Device.

Figure F.9 shows the relationships between the components involved.

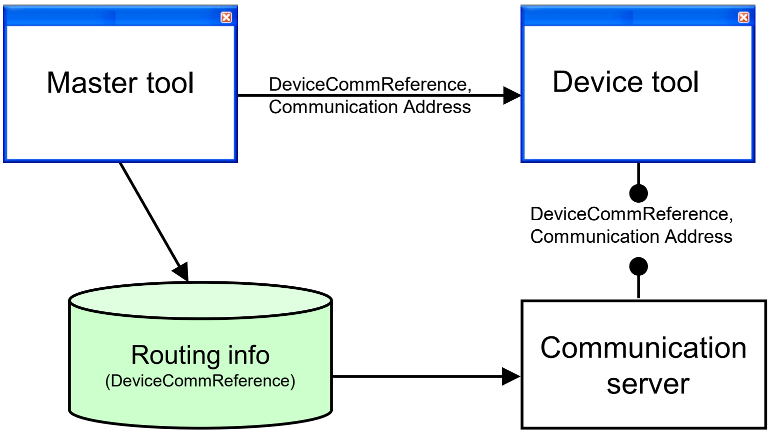


Figure F.9 – Communication Server

It is always possible for a Device tool to use its native communication interfaces (for example serial RS232) as an alternative besides the Communication Server.

F.5.3 Gateways

A Communication Server allows a communication connection across network boundaries (see Figure F.8).

The Engineering System, all Device tools and the Communication Server are located on the same PC which is connected e.g. via an Ethernet adapter to a network. The target Devices can be found behind a gateway which can work in different ways. From the Device tool point of view, it is irrelevant where the Device is located because the network structure is handled by the Communication Server.

The Communication Server is potentially able to manage all gateway types which are supported by the Engineering System itself. The gateway functionalities itself are encapsulated by the Communication Server. Only gateway types known by the Communication Server can be supported (no nested communication).

If a Device can be reached through multiple paths in the network, it is up to the Engineering System to decide, which network path is used for communication.

F.5.4 Configuration of the Communication Server

In order to build the network communication addresses from the Device communication reference, the Communication Server requires configuration data from the Engineering System/Master tool. The structure of configuration data itself and the way how the data is sent to the Communication Server is manufacturer specific and will not be standardized.

F.5.5 Definition of the Communication Interface

The Communication Server implements the interface "IFdtCommunication" and uses the "IFdtCommunication-Events" and "IFdtCommunicationEvents2" as described in [18]. All other DTM interfaces which are described in [18] are not relevant for the Communication Server. Due to this constraint, a Communication Server cannot be used in an FDT environment as communication DTM.

F.5.6 Sequence for establishing a communication relation

An interaction of Engineering System/Master tool, Device tool and Communication Server (CS) is required to establish a communication relation. The sequence is as follows:

At first, a Device is integrated into the Master tool with the corresponding configuration file (IODD). Within the Engineering System, communication addresses and bus parameter are adjusted. Together with other network data, topology data for the network is the result.

Furthermore, the Master tool shall build a unique Device communication reference. This reference is passed to the Device tool when it is launched with the help of the TPF (keyword “DeviceCommReference”). The Device tool is now able to establish a connection to the Device using the Communication Server and Device communication reference.

The Communication Server itself interprets the Device communication reference and converts it to network addresses. Therefore, it uses the configuration data from the Master tool. Because it is up to the CS to decide if the Device communication reference or the communication address itself is used, the Device tool shall always pass both attributes in the ConnectRequest XML document. If no routing functionality is required, the CS does not require the proprietary configuration. In order to connect, the CS can use the communication address itself from the Master tool. Figure F.10 shows how a communication connection is established.

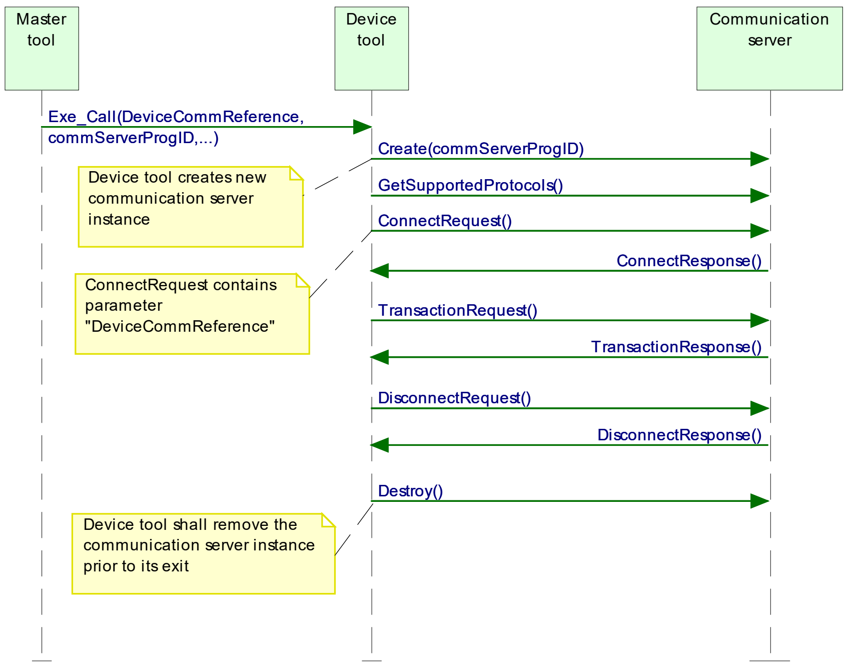


Figure F.10 – Sequence chart for establishing communication

The passed ProgID (Keyword commServer-ProgID) can be used to create a new instance of the Communication Server by the Device tool. There is a 1:1 relationship between Device tool and Communication Server instance. The Communication Server instance can connect to one or more Devices.

Figure F.11 shows a code fragment in C++ as an example on how to create a new instance.

commServer-ProgID of TPF

```
r1 = CLSIDFromProgID(L"TCI.MyCommunicationServer", &clsid);
r1 = CoCreateInstance (clsid, NULL, CLSCTX_INPROC_SERVER,
     __uuidof (IFdtCommunication), (void**)&m_pITciCommunicationServer);
```

Figure F.11 – Create Communication Server instance

It is recommended to create the Communication Server instance as "in process server" (CLSCTX_INPROC_SERVER) due to performance issues.

After a new instance of the Communication Server is created, all methods of the interface "IFdtCommunication" can be called. At first a Device tool shall call the "GetSupportedProtocols" method to find out if the required protocol is supported by the CS. If not, the Device tool shall inform the user. A new connection is established with help of the function "ConnectRequest". Among others, as invocation parameter a pointer to the callback interface (Interface IFdtCommunicationEvents) is passed. This means that a Device tool shall implement this interface.

The Device tool is responsible to release the Communication Server instance when the tool exits. If the Communication Server instance was created in the process of the Device tool, as recommended before, this is done automatically since the instance is terminated with the process of the Device tool.

F.5.7 Usage of the Communication Server in stand-alone mode

If a Device tool is not called from a Master tool with DTI, it shall find out the ProgID of the Communication Server by itself. In this case the "Component Categories" of the system registry can be used (HKEY_CLASSES_ROOT\Component Categories).

The following values are defined for the DTI Communication Server:

Symbolic Name of CatID: CATID_DTI_CS

UUID of CatID: {7DDC60A6-1FD4-45a2-917F-0F8FC371BC57}

A Device tool can find out the ProgID of the Communication Server with the help of the Standard Component Categories Manager. If more than one component is assigned to this category, the user of the Device tool shall select one of the Communication Servers.

If a Communication Server does not support the "Stand-Alone" mode (i.e. a Communication Server instance cannot be created by a Device tool), a system registry entry should not be made.

A Device tool that supports Conformance Class 3 and is intended for "Stand-Alone" mode shall store the DeviceCommReferences together with its DDOs. Whenever the DeviceCommReference is changed by the Master tool while copying the entire project or while retrieving the project, the Device tool shall check and – if changed – update the DeviceCommReference when called from the Master tool with DTI. There are two general possibilities:

1) The Device tool checks and updates the DeviceCommReference of a particular Device immediately before connection. Hint: After copy/retrieval of a Master tool project, the user should call the Device tool via DTI and connect to the particular Device(s) prior to the connection to this/these Device(s) later on in "Stand-Alone" mode.
2) The Device tool checks and updates the DeviceCommReferences of all Devices immediately after being called by the Master tool via DTI. Hint: After copy/retrieval of a Master tool project, the user should call the Device tool via DTI. Then, all Devices can be connected later in “Stand-Alone” mode.

F.5.8 SDCI specifics

The SDCI schema defined in [19] shall be used as communication schema.

Table F.6 shows the mapping between the TPF keywords and the attributes in the communication schema.

The communication parameters passed during the Device tool invocation shall be used as input for the Connect Request XML document to be used in the connect method. Additionally, the device communication reference (Keyword "DeviceCommReference" in Table F.6) shall be entered in the Connect Request XML document as attribute “systemTag”.

Table F.6 – Communication Schema mapping

Attribute of ConnectRequest element (FDTIOLinkCommunicationSchema.xml)	Parameter Keyword in TPF file	Remarks
fdt:nodeId	–	Unused
systemTag	"DeviceCommReference" attribute of element "Device".	

Figure F.12 shows an example.

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTIOLinkCommunicationSchema.xml"
        xmlns:fdt="x-schema:FDTDataTypesSchema.xml">
        <ConnectRequest systemTag="Controller3/Gateway2/Unit1"/>
</FDT>
```

DeviceCommReference of TPF

Figure F.12 – Example of a Connect Request XML document for SDCI

F.5.9 Changing communication settings

If it is necessary to change the communication address (Master, Port) in the Master tool, the Device tool needs information about the new communication address. This shall be done via relaunching the Device tool by the user of the Master tool. During relaunch, the new communication parameters are passed to the Device tool. With these communication parameters a new communication relation can be established to the Device.

If the Device communication reference is used instead of the communication address between Device tool and Communication Server, no relaunch of the tool is required, because the Device communication reference does not change whenever the communications address changes. In this case, the Communication Server itself can reconnect to the Device with the new communication address (Master, Port).

For an existing connection, changed communication parameters in the Master tool project shall not have any impact. Changed communication parameters shall be used when a connection is (re)established.

F.6 Reaction on incorrect tool behavior

Table F.7 describes the system reaction if a Master tool or Device tool works incorrectly.

Table F.7 – Reaction on incorrect tool behavior

Fault	Description	System reaction
XML structure of PID file not valid	The PID file of a Device tool does not validate with the XML Schema in F.9.1	The Master tool should only show an error message if required schema elements or attributes are missing. All unknown elements or attributes shall be ignored.
XML structure of TPF file not valid	The TPF file generated by the Master tool does not validate with the XML Schema in F.9.3	The Device tool should only show an error message if required schema elements or attributes are missing. All unknown elements or attributes shall be ignored.
Device tool cannot be invoked	When the operation system is instructed to create a new process (tool invocation) the function returns an error code. Reason could be that the path of the exe file in the system registry is incorrect.	Master tool shall show an error message (tool cannot be invoked) with the name and path of the exe file.
CommunicationServer object cannot be created. See F.5.6	The "CoCreateInstance" function returns an error code when an object with the ProgID of the TPF should be instantiated.	The Device tool should show an error message.

Fault	Description	System reaction
TPF file not deleted by the Device tool	The TPF file was not removed by the Device tool as described in F.3.1	Master tool should delete the TPF file when it is launched (garbage collection). If the file cannot be deleted, the Master tool should not show an error message.
DeviceCommReference not valid (Communication channel cannot be established). See Clause F.5.	Device tool is using a not existing DeviceCommReference in the Master tool.	The Device tool should show an error message.

F.7 Compatibility

F.7.1 Schema validation

XML documents can easily be validated with the help of standard parsers and schema files. If the structure of an XML document does not follow the rules defined in the corresponding schema, the XML parser rejects the document. This is not very practical if tools with different versions of DTI files shall work together since a newer XML document cannot be processed by previous software.

To implement a robust model, the Master tool and the Device tools shall ignore any XML attributes or elements not recognizable in a valid XML document. This means that XML schema validation shall not be used. The schema files in Annex F.9 are for information purposes only.

The installation program of the Device tool can always install the newest PID file version. The Master tool shall ignore any unknown XML attributes or elements.

F.7.2 Version policy

If it is necessary to modify the structure definition of a TPF with the result that a new version of the invocation interface is defined, the Master tool shall ensure that the right version of the TPF is created. That means it shall use an earlier version of the structure if the Device tool is only able to support the earlier version.

The PID file version of the Device tool determines the newest supported version of the corresponding Device tool. See Annex F.3.3 for details.

If a Device tool supports a newer version than the Master tool, the Master tool uses its newest TPF version. In this case the Device tool shall work with the old schema version.

F.8 Scalability

F.8.1 Scalability of a Device tool

The manufacturer of a Device tool can choose to support different function levels of DTI as shown in Table F.8.

Table F.8 – DTI conformance classes

Conformance Class	Description
C1 (Navigation)	Setup program creates system registry entries as described in Annex F.3.2. This allows the user to invoke the Device tool from the context of a selected Device in the Master tool without any impact on an existing Device tool itself.
C2 (Parameter transfer)	The Device tool uses the information of the TPF. In this case, for example, the tool can read FST parameter instances or use a communication address for its proprietary communication channel. This way, the user can be relieved from multiple entries. The implementation effort is limited to evaluation of the TPF file for internal initialization of the Device tool.
C3 (DTI communication with optional backchannel)	The full functionality is available if the Device tool uses the DTI Communication Server. This component enables the tool to manage all network boundaries implemented by the Master tool. In this case the Device tool shall support the IFdtCommunication/IFdtCommunicationEvents/IFdtCommunicationEvents2 interface.

Conformance Class	Description
	In case of the backchannel option, the Master tool uses the information of the TBF. In this case, for example, the tool can read FST parameter instances or use the I/O Process Data description. This way, the user can be relieved from multiple entries. The implementation effort is limited to evaluation of the TBF file for internal processing of the Master tool.

Table F.9 shows the DTI relevant features of a Device tool.

Table F.9 – DTI feature levels of Device tools

Function	Annex	Conformance Class	Feature Name for PID file
Make system registry entries	F.3.2	C1	–
Provide PID file during installation procedure	F.3.3	C1	–
Avoid multiple program instances		C2	–
Interpret TPF	F.3.4	C2	–
Delete TPF	F.3.7.2	C2	–
Use the Communication Server interface		C3	–

F.8.2 Scalability of a Master tool

A Master tool shall support all DTI feature levels/conformance classes.

F.8.3 Interactions at conformance class combinations

Table F.10 defines how a Master tool and a Device tool shall interact depending on their conformance class.

Table F.10 – Interactions at conformance class combinations

Master tool	Device tool	Interaction
C2 or C3	C1	Device tool is launched, no parameters are passed. The Master shall not generate a TPF because it would not be deleted by the Device tool.
C2 or C3	C2	Device tool is launched, Parameters are passed through TPF.
C2	C3	Device tool is launched, Parameters are passed through TPF.
C3	C3	Device tool is launched, Parameters are passed through TPF. Communication via Communication Server is possible.

F.9 Schema definitions

F.9.1 General

The schema definitions in this Annex F.9 are for information only (see Annex F.7.1).

F.9.2 Schema of the PID

Figure F.13 shows the XML schema of the Program Interface Description file.

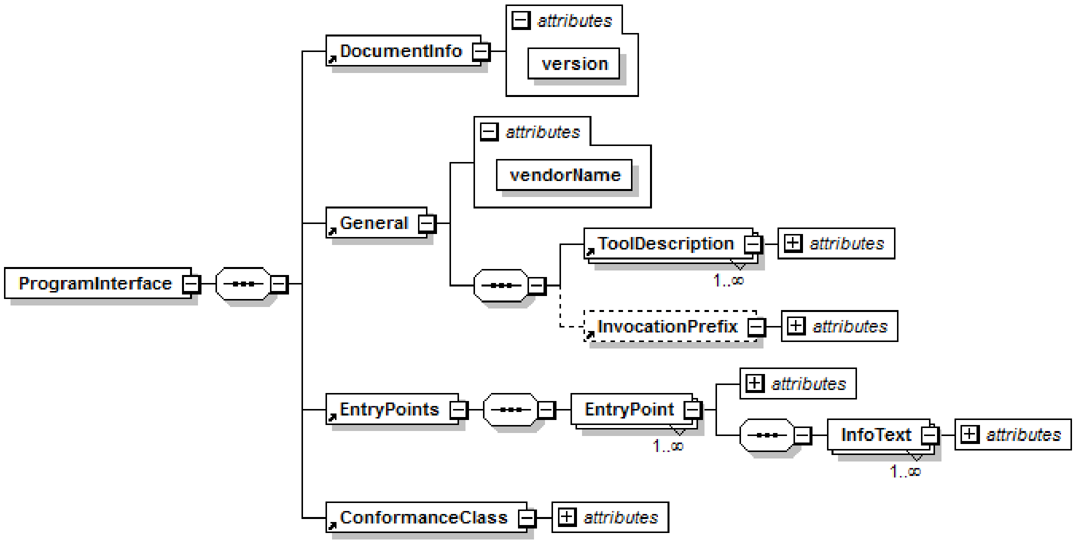


Figure F.13 – XML schema of the PID file

Figure F.13 is based on the XML code as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2017/02/PID" xmlns:prim="http://www.io-
link.com/DTI/2017/02/Primitives" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.io-link.com/DTI/2017/02/PID" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xsd:import namespace="http://www.io-link.com/DTI/2017/02/Primitives" schemaLocation="DTI-
Primitives1.0.xsd"/>
  <xsd:element name="DocumentInfo">
    <xsd:complexType>
      <xsd:attribute name="version" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="V\d+(\.\d+){1,7}"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ToolDescription">
    <xsd:complexType>
      <xsd:attribute name="lang" type="xsd:string" use="required"/>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="InvocationPrefix">
    <xsd:complexType>
      <xsd:attribute name="name" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="General">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ToolDescription" maxOccurs="unbounded"/>
        <xsd:element ref="InvocationPrefix" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="vendorName" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="EntryPoints">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="EntryPoint" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:complexContent>
```

```xml
              <xsd:extension base="prim:ObjectT">
                <xsd:sequence>
                  <xsd:element name="InfoText" maxOccurs="unbounded">
                    <xsd:complexType>
                      <xsd:attribute name="lang" type="xsd:string" use="required"/>
                      <xsd:attribute name="name" type="xsd:string" use="required"/>
                      <xsd:attribute name="description" type="xsd:string" use="required"/>
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
                <xsd:attribute name="id" type="prim:IdT" use="required"/>
              </xsd:extension>
            </xsd:complexContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ConformanceClass">
    <xsd:complexType>
      <xsd:attribute name="name" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="C1"/>
            <xsd:enumeration value="C2"/>
            <xsd:enumeration value="C3"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ProgramInterface">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="DocumentInfo"/>
        <xsd:element ref="General"/>
        <xsd:element ref="EntryPoints"/>
        <xsd:element ref="ConformanceClass"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

F.9.3 Schema of the TPF

Figure F.14 shows the XML schema of the Temporary Parameter File.

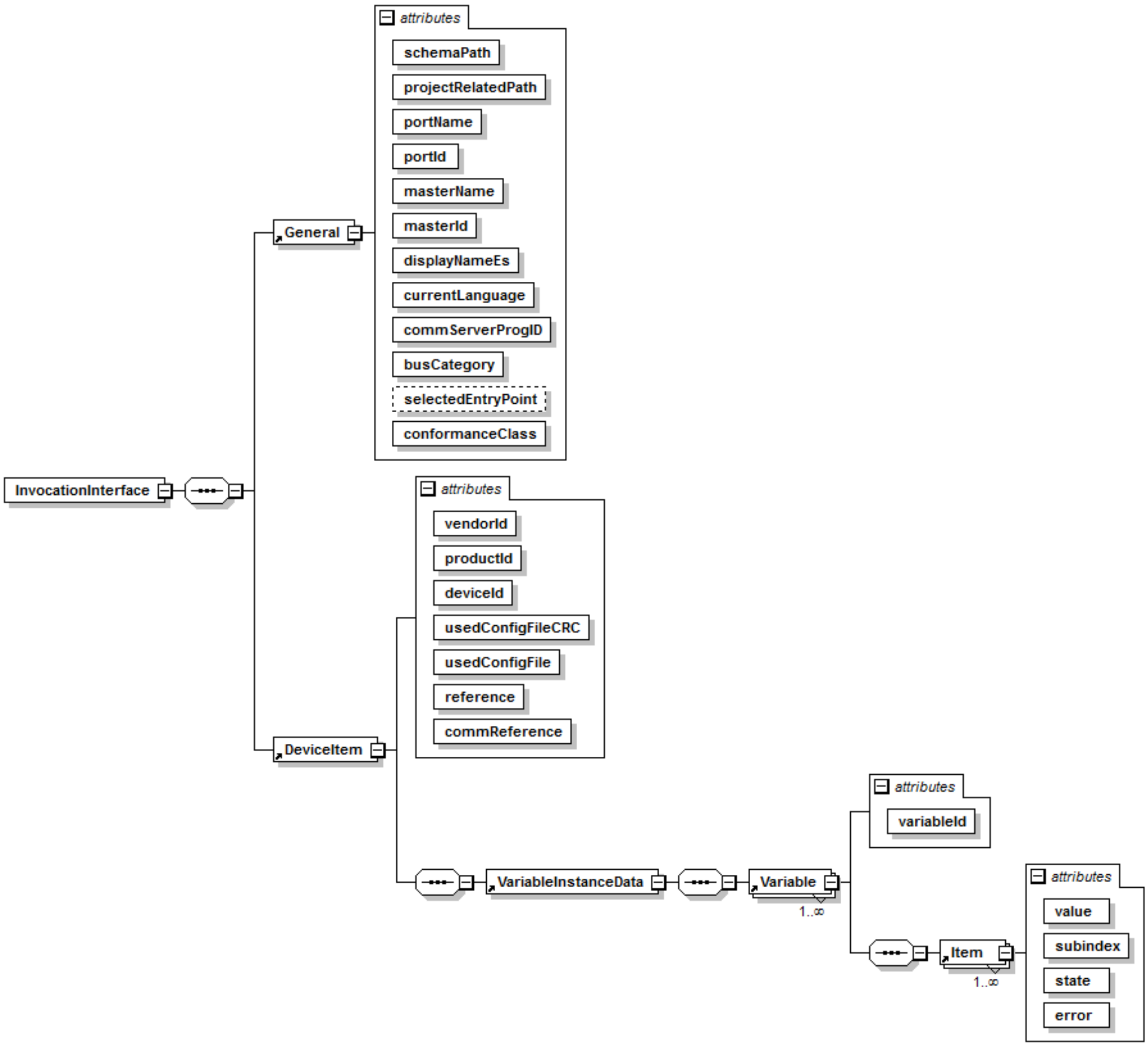


Figure F.14 – XML schema of the TPF

Figure F.14 is based on the XML code as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2017/02/TPF" xmlns:prim="http://www.io-
link.com/DTI/2017/02/Primitives" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.io-link.com/DTI/2017/02/TPF" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xsd:import namespace="http://www.io-link.com/DTI/2017/02/Primitives" schemaLocation="DTI-
Primitives1.0.xsd"/>
  <xsd:element name="VariableInstanceData">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Variable" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Variable">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Item" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="variableId" type="prim:IdT" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Item">
    <xsd:complexType>
      <xsd:attribute name="value" type="xsd:string" use="required"/>
      <xsd:attribute name="subindex" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:unsignedShort">
            <xsd:maxInclusive value="255"/>
```

```
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="state" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="empty"/>
            <xsd:enumeration value="initial"/>
            <xsd:enumeration value="device"/>
            <xsd:enumeration value="read error"/>
            <xsd:enumeration value="write error"/>
            <xsd:enumeration value="valid"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="error" type="xsd:integer" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="InvocationInterface">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="General"/>
        <xsd:element ref="DeviceItem"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="General">
    <xsd:complexType>
      <xsd:attribute name="schemaPath" type="xsd:string" use="required"/>
      <xsd:attribute name="projectRelatedPath" type="xsd:string" use="required"/>
      <xsd:attribute name="portName" type="xsd:string" use="required"/>
      <xsd:attribute name="portId" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:unsignedShort">
            <xsd:maxInclusive value="255"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="masterName" type="xsd:string" use="required"/>
      <xsd:attribute name="masterId" type="xsd:int" use="required"/>
      <xsd:attribute name="displayNameEs" type="xsd:string" use="required"/>
      <xsd:attribute name="currentLanguage" type="xsd:string" use="required"/>
      <xsd:attribute name="commServerProgID" type="xsd:string" use="required"/>
      <xsd:attribute name="busCategory" type="xsd:string" use="required"/>
      <xsd:attribute name="selectedEntryPoint" type="prim:IdT" use="optional"/>
      <xsd:attribute name="conformanceClass" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="C1"/>
            <xsd:enumeration value="C2"/>
            <xsd:enumeration value="C3"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <!-- IOLink -->
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="DeviceItem">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="VariableInstanceData"/>
      </xsd:sequence>
      <xsd:attribute name="vendorId" type="xsd:unsignedShort" use="required"/>
      <xsd:attribute name="productId" type="xsd:string" use="required"/>
      <xsd:attribute name="deviceId" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:unsignedInt">
            <xsd:maxInclusive value="16777215"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="usedConfigFileCRC" type="xsd:int" use="required"/>
      <xsd:attribute name="usedConfigFile" type="xsd:string" use="required"/>
      <xsd:attribute name="reference" type="xsd:string" use="required"/>
      <xsd:attribute name="commReference" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

F.9.4 Schema of the TBF

Figure F.15 shows the XML schema of the Temporary Backchannel File.

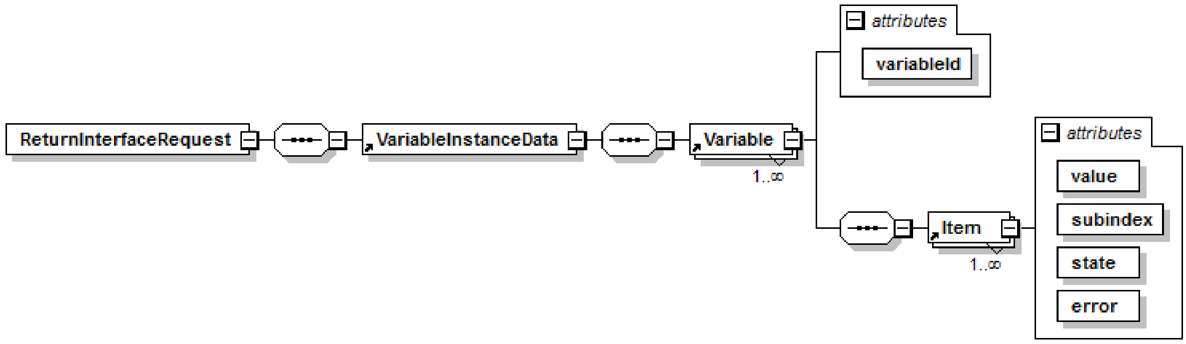


Figure F.15 – XML schema of a TBF

Figure F.15 is based on the XML code as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2017/02/TBF" xmlns:prim="http://www.io-
link.com/DTI/2017/02/Primitives" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.io-link.com/DTI/2017/02/TBF">
  <xsd:import namespace="http://www.io-link.com/DTI/2017/02/Primitives" schemaLocation="DTI-
Primitives1.0.xsd"/>
  <xsd:element name="VariableInstanceData">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Variable" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Variable">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Item" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="variableId" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Item">
    <xsd:complexType>
      <xsd:attribute name="value" type="xsd:string" use="required"/>
      <xsd:attribute name="subindex" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:unsignedShort">
            <xsd:maxInclusive value="255"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="state" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="empty"/>
            <xsd:enumeration value="initial"/>
            <xsd:enumeration value="device"/>
            <xsd:enumeration value="read error"/>
            <xsd:enumeration value="write error"/>
            <xsd:enumeration value="valid"/>
            <!--xsd:enumeration value="changed"/-->
            <!-- should be transferred to device or stored in database before DTI invocation -->
            <!-- could be changed to empty before DTI invocation -->
            <!-- could be changed to empty or valid before DTI invocation -->
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="error" type="xsd:integer" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Response">
    <xsd:complexType>
      <xsd:attribute name="value" type="xsd:boolean" use="required"/>
    </xsd:complexType>
```

```
  </xsd:element>
  <xsd:element name="ReturnInterfaceRequest">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="VariableInstanceData"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ReturnInterfaceResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Response"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:group name="ReturnInterface">
    <xsd:choice>
      <xsd:element ref="ReturnInterfaceRequest"/>
      <xsd:element ref="ReturnInterfaceResponse"/>
    </xsd:choice>
  </xsd:group>
</xsd:schema>
```

F.9.5 Schema of the TAF

The schema of the TAF corresponds to the schema of the TBF in F.9.4.

F.9.6 Schema of DTI primitives

The DTI primitives are defined in the XML code as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2017/02/Primitives"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.io-
link.com/DTI/2017/02/Primitives">
  <xsd:annotation>
    <xsd:documentation>In this schema, only the necessary types and attributes for DTI are used
from the Common Primitives Schema.</xsd:documentation>
    <xsd:appinfo>
      <schemainfo versiondate="20170225"/>
    </xsd:appinfo>
  </xsd:annotation>
  <!-- SIMPLE TYPES -->
  <xsd:simpleType name="IdT">
    <xsd:annotation>
      <xsd:documentation>Base Type for Object identifiers</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="GuidT">
    <xsd:annotation>
      <xsd:documentation>GUID</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="\{[0-9A-Fa-f]{8}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{4}\-[0-
9A-Fa-f]{12}\}"/>
      <xsd:pattern value="[0-9A-Fa-f]{8}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{4}\-[0-9A-Fa-f]{4}\-[0-9A-
Fa-f]{12}"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- _____________________________________________________ -->
  <!-- COMPLEX TYPES -->
  <!-- Main Types -->
  <xsd:complexType name="DocumentT">
    <xsd:annotation>
      <xsd:documentation>Type for all top level elements</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="DocumentInfo" type="DocumentInfoT"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="DocumentInfoT">
    <xsd:attribute name="Version" type="xsd:string" use="required" fixed="1.1"/>
  </xsd:complexType>
  <!-- ELEMENT DECLARATIONS -->
  <!-- _____________________________________________________ -->
  <!-- Text Definition Elements-->
```

```
  <xsd:complexType name="ObjectT">
    <xsd:annotation>
      <xsd:documentation>Base type</xsd:documentation>
    </xsd:annotation>
  </xsd:complexType>
  <xsd:complexType name="FeatureT">
    <xsd:annotation>
      <xsd:documentation>Base type</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="Name" type="xsd:string" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="ParameterT" mixed="true">
    <xsd:attribute name="Name" type="xsd:string" use="required"/>
  </xsd:complexType>
  <!-- _____________________________________________________ -->
  <!--Specialized Parameters-->
  <xsd:complexType name="StringParameterT">
    <xsd:complexContent>
      <xsd:extension base="ParameterT">
        <xsd:attribute name="Value" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ELEMENT DECLARATIONS -->
  <xsd:element name="Document" type="DocumentT">
    <xsd:unique name="OBJ-ID">
      <xsd:selector xpath=".//*"/>
      <xsd:field xpath="@ID"/>
    </xsd:unique>
  </xsd:element>
  <xsd:element name="Object" type="ObjectT"/>
  <xsd:element name="Parameter" type="ParameterT"/>
  <xsd:element name="StringParameter" type="StringParameterT" substitutionGroup="Parameter"/>
  <xsd:element name="Feature" type="FeatureT"/>
  <xsd:simpleType name="ConformanceClassEnumT">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="C1"/>
      <xsd:enumeration value="C2"/>
      <xsd:enumeration value="C3"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

Annex G
(normative)

Main scenarios of SDCI-FS

G.1 Overview

Table G.1 shows main scenarios, the initial key parameters, and the associated system activities. Its purpose is to provide a brief overview and it contains references to clauses with detailed descriptions.

Table G.1 – Main scenarios of SDCI-FS

Scenario	Initial parameters	System activities
OSSD operation (on FS-DI or FS-Master)	Authenticity = 0 Port = 0 FSP_TechParCRC ≠ 0 (factory settings)	1. Modify FST parameter via "USB Master" tool (option; see off-site commissioning in IEC 61131-9) and IODD 2. Adapt FSP_TechParCRC (see 11.7.8) using "Dedicated Tool" 3. FS-Device evaluates validity of technology parameters (FST) via FSP_TechParCRC at STARTUP 4. Plug, validate & play (default)
Commissioning (Test = monitored operation) See Figure G.1	Authenticity = 0 Port = 0 FSP_TechParCRC ≠ 0 (factory settings)	1. Set FSP_TechParCRC = 0 temporarily (FS-Device and FSP_Verify-Record) via FS-Master tool 2. Assign Authenticity and Port to FS-Device via FS-Master tool and via Authenticity record 3. Assign FSP parameter via FS-Master tool and assign FSP_VerifyRecord to FS-Master (see 9.4.2 "Local modifications") 4. PowerOFF/ON FS-Device (reset) for a duration of at least the time defined by auxiliary parameter FSP_MinShutDownTime (see A.2.12) 5. FS-Master transfers FSP_VerifyRecord to FS-Device (see A.2.10 and 10.4.3.1) 6. Run in test mode (No verification except FSP_TechParCRC = "0"; Data Storage disabled) 7. FST parameter can now be modified 8. FS-Master tool and FS-Master are responsible to indicate test mode and/or to prevent from running in test mode w/o tool connection. For this purpose, the FS-Master shall store the Port configuration with FSP_TechParCRC = "0" in a volatile manner. Thus, after restart of the FS-Master ("power cycle") w/o tool, the FS-Device will not receive a valid Port configuration and will not perform safety communication.
Commissioning (Arm and validate) See Figure G.1	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC = 0	1. Assign actual FSP_TechParCRC (FS-Device and FSP_VerifyRecord) via FS-Master tool 2. Transfer FSP Parameter records to FS-Master, secured by FSP_ProtParCRC via FS-Master tool (SMI service) 3. Port restart after Port configuration (see IEC 61131-9) 4. Upload parameters to Data Storage (FSP and FST) in PREOPERATE, see "Backup/Restore" in IEC 61131-9 5. FS-Master transfers FSP_VerifyRecord to FS-Device (see A.2.10) 6. Run in armed mode (Verification: Authenticity + FSP_TechParCRC compared), see 11.7.6 7. Validation according to safety manual of FS-Device.
Armed to Commissioning	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC ≠ 0	1. Set FSP_TechParCRC = 0 temporarily (FS-Device and FSP_Verify-Record) via FS-Master tool 2. Follow activity 4. to activity 8. in scenario "Commissioning"
Replacement by FS-Device with factory settings w/o tools	Authenticity = 0 Port = 0 FSP_TechParCRC ≠ 0	1. Download and adopt parameters from Data Storage (FSP and FST) if Authenticity and Port = 0 (see 12.5.1 and 12.5.2 in IEC 61131-9) 2. Run in armed mode (Verification: Authenticity + FSP_TechParCRC compared), see 11.7.6 and A.2.10

Scenario	Initial parameters	System activities
		3. Validation according to safety manual of FS-Device.
Misconnection of configured FS-Devices	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC ≠ 0	1. No adoption of downloaded parameters from Data Storage (FSP and FST) since Authenticity and Port ≠ "0" in FS-Device 2. SCL not started (Verification: Authenticity + FSP_TechParCRC compared), see 11.7.6 and A.2.10 3. Error message: "Misconnection" (0xB003 or 0xB004, see Annex B).
NOTE "A-Code" refers to IEC 61784-3:2021		

G.2 Sequence chart of commissioning

Sequence chart in Figure G.1 illustrates major activities during commissioning of an FS-Device with factory settings. First phase is the test phase of FS-Device and safety functions while in monitored operation by personnel. Second phase comprises arming of Port and corresponding FS-Device as well as validation of the safety function according to safety manuals.

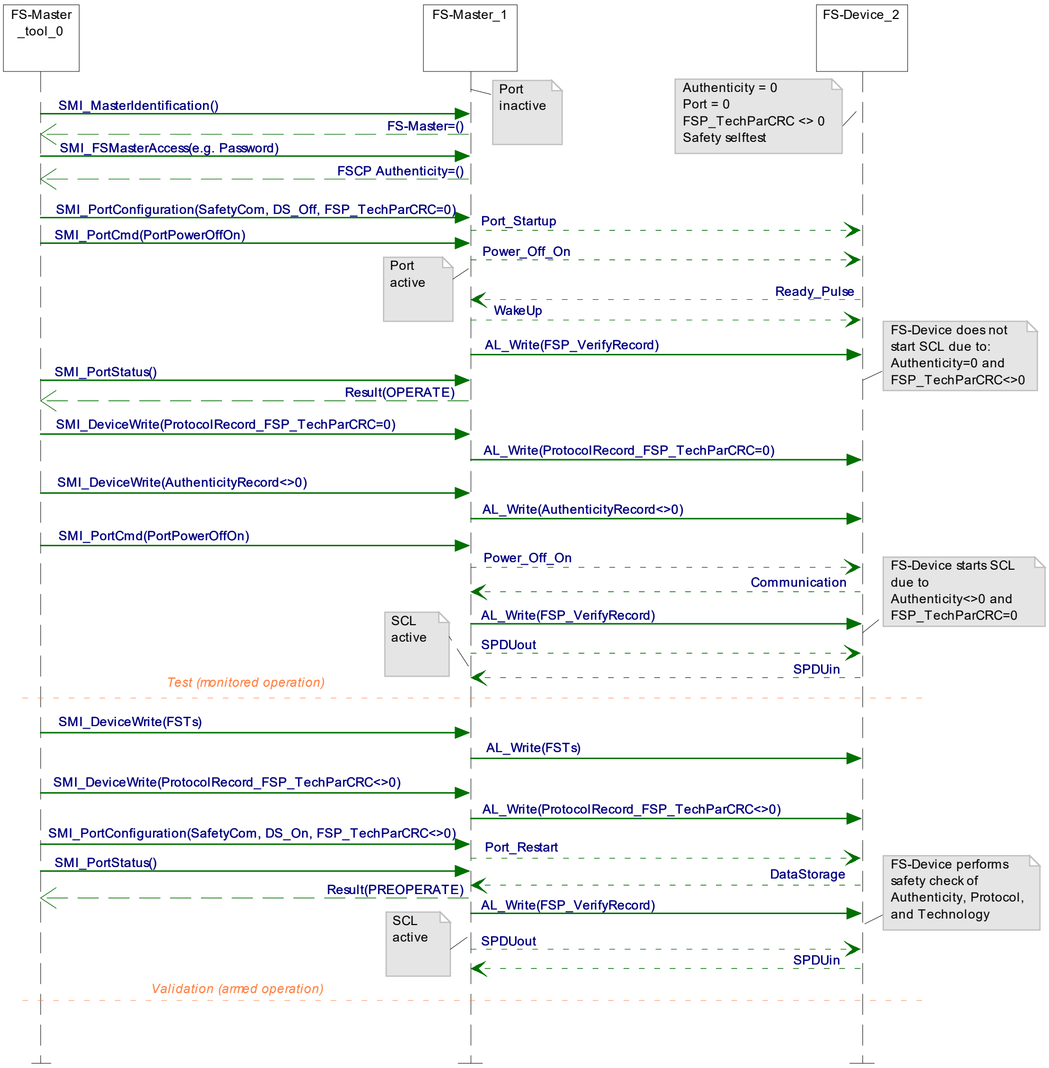


Figure G.1 – Commissioning with test and armed operation

G.3 Sequence chart of replacement

Sequence chart in Figure G.2 illustrates major activities after an FS-Device replacement by one with factory settings.

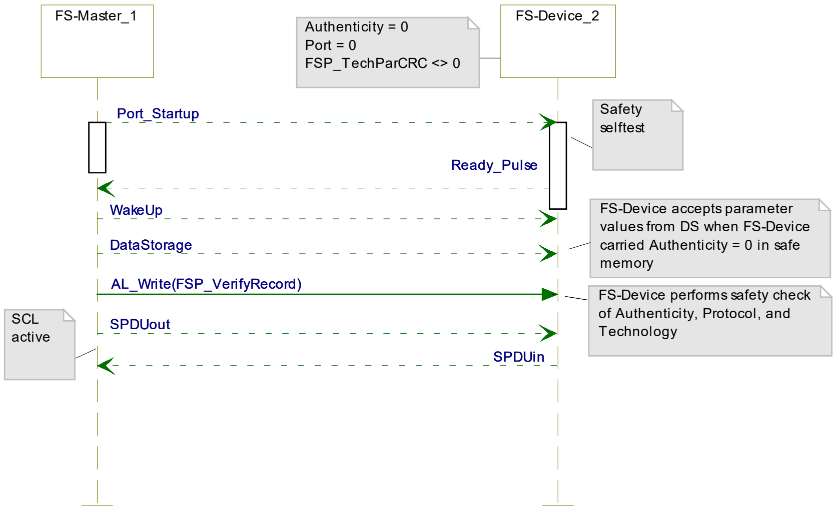


Figure G.2 – FS-Device replacement

G.4 Sequence chart of misconnection

Sequence chart in Figure G.3 illustrates major activities after an FS-Device replacement by one with other parameters than factory settings.

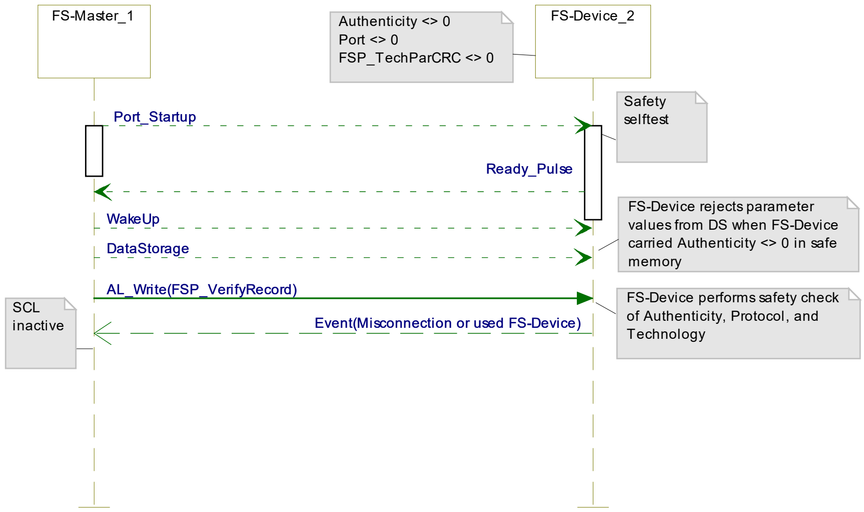


Figure G.3 – FS-Device misconnection

Annex H
(normative)

System requirements

H.1 Indicators

H.1.1 General

Indicators for FS-Devices are not mandatory since for example proximity sensors may be too small for LEDs (light emitting diode).

FS-Masters and FS-Devices may be used in a mix of different technologies such as

- Fieldbus safety modules for inputs (e.g. F-DI module) or outputs (e.g. F-DO module),
- Safety devices such as light curtains connected to fieldbuses via FSCPs,
- SDCI Masters and Devices.

Thus, it is the designer's responsibility to layout the indication of the signal status, modes, or operations for FS-Masters and FS-Devices.

H.1.2 OSSDe

In case an FS-Master Port is running in OSSDe mode it behaves similar to an F-DI module port. One possibility of indication is using the same indication as with the SIO mode.

H.1.3 Safety communication

In case an FS-Master Port is running in SCL mode, the normal non-safety operation indication can also be used.

H.1.4 Acknowledgment request

A machine is not allowed to restart automatically after a stop. Usually, after repair or clearance, the signal/service "ChFAckReq" is switched ON as specified in 11.11.4 and 11.11.5. It is highly recommended to indicate this signal on an FS-Master Port and optionally on FS-Devices where it is likely to cause a trip due to high frequency or duration of exposure to a safety function.

H.2 Installation guidelines, electrical safety, and security

SDCI installation guidelines shall be considered (see [20]).

Only FS-Masters and FS-Devices providing a short form functional safety assessment report according to IEC 61508 or ISO 13849-1 together with a certificate of the assessment body are permitted. The short form report shall indicate all considered clauses and paragraphs of the used relevant standards and the corresponding assessment results.

Wireless connection between FS-Master and FS-Device is only permitted if interdependency with other wireless connections can be precluded, for example via inductive couplers.

No components in the link between FS-Master and FS-Device are permitted that are storing, inserting, or delaying messages.

Manufacturer/vendor of FS-Masters and/or FS-Devices shall define installation constraints for the operation of OSSD devices or FS-Devices in OSSDe mode within their safety manuals.

Requirements of IEC 61010-2-201 and IEC 60204-1 with respect to electrical safety (SELV/PELV) shall be observed.

The zones and conduit concept of IEC 62443 applies for security and/or the rules of the applicable FSCP system.

H.3 Safety function response time

Safety manuals of FS-Master shall provide information on how to determine the safety function response time for OSSDe and for communication modes (see Clause H.6).

H.4 Duration of demands

Short demands of FS-Devices may not trip a safety function due to its chain of independent communication cycles across the network. Therefore, a demand shall last for at least two SCL (SPDU) cycles.

H.5 Maintenance and repair

FS-Devices can be replaced at runtime. Restart of the corresponding safety function is only permitted if there is no hazardous process state, after validation of the safety function(s), and after an operator acknowledgment.

H.6 Safety manual

FS-Masters and FS-Devices shall provide safety manuals according to the relevant national and international standards, for example IEC 61784-3:2021.

Manufacturer/vendor of FS-Masters and/or FS-Devices shall specify appropriate mitigation means in the safety manual for the deployment of SDCI-FS components in harsh industrial environment such as in EMC zones B and C according to IEC 61131-2.

Manufacturer/vendor of FS-Masters and/or FS-Devices shall define all constraints for the operation of OSSD devices or FS-Devices in OSSDe mode within their safety manuals.

Manufacturer/vendor of FS-Masters and/or FS-Devices shall define all constraints for the operation of FS-Devices in communication mode within their safety manuals such as limitations with respect to storing elements, inductive or optical couplers, and alike.

Manufacturer/vendor of FS-Masters and/or FS-Devices shall define the maintenance rules with respect to the PFH-Monitor (see Table 41).

Manufacturer/vendor of FS-Devices shall provide the "worst case delay time" (WCDT) value. WCDT is defined as the time from triggering an FS-Device (sensor) until its output shows a corresponding signal change or Process Data change under worst case conditions. For an FS-Device (actuator) it is the time from signal change or Process Data change to the actuator's safe state.

Manufacturer/vendor of FS-Devices shall provide the "one fault delay time" (OFDT) value. The definition of OFDT is similar to WCDT, however in case of a fault within the FS-Device at the time of triggering. Therefore, the value of OFDT is greater than the value of WCDT.

Manufacturer/vendor of FS-Masters shall provide information on how to determine the safety function response time as specified in IEC 61784-3:2021 using WCDTs and considering OFDTs.

Annex I
(informative)

Information for test and assessment of SDCI-FS components

Information about test laboratories, which test and validate the conformance of SDCI-FS products such as FS-Masters and FS-Devices with IEC 61139-2 can be obtained from the National Committees of the IEC or from the following organization:

IO-Link Community
Haid-und-Neu-Str. 7
76131 Karlsruhe
GERMANY

Phone: +49 721 9658 590
Fax: +49 721 9658 589
E-Mail: info@io-link.com
URL: www.io-link.com

Bibliography

[1] ISO/IEC 19505-2:2012, *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 2: Superstructure*

[2] Bruce P. Douglass, *Real Time UML – Advances in the UML for Real-Time Systems*, 3rd Edition, Addison-Wesley, ISBN 0-321-16076-2

[3] Chris Rupp, Stefan Queins, die SOPHISTen, *UML 2 glasklar – Praxiswissen für die UML-Modellierung*. Hanser-Verlag, 2012, ISBN 978-3-446-43057-0

[4] IEC/TR 62390, *Common Automation Device – Profile Guideline*

[5] IEC 60947-5-5, *Low-voltage switchgear and controlgear – Part 5-5: Control circuit devices and switching elements – Electrical emergency stop device with mechanical latching function*

[6] Position Paper CB24I, *Classification of Binary 24 V Interfaces – Functional Safety aspects covered by dynamic testing*, Edition 2.0.1: https://www.zvei.org/verband/fachverbaende/fachverband-automation/schaltgeraete-schaltanlagen-industriesteuerungen/

[7] Klaus Grimmer, *AIDA_IP-67-Safety_Positionspapier*, June 27th, 2013

[8] IEC 60947-5-2, *Low-voltage switchgear and controlgear – Part 5-2: Control circuit devices and switching elements – Proximity switches*

[9] IO-Link Community, *IO-Link Safety (Single Platform) – Requirements, Use Cases, and Concept Baseline,* V1.0, November 2014, Order No. 10.062

[10] IEC 62769 series: *Field Device Integration (FDI)*

[11] FDT Joint Interest Group, *FDT 2.0 – Specification*, V1.0, Order No. 0001-0008-000

[12] IEC 60947-5-6, *Low-voltage switchgear and controlgear – Part 5-6: Control circuit devices and switching elements – DC interface for proximity sensors and switching amplifiers (NAMUR)*

[13] Philip Koopman, *Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks*, The International Conference on Dependable Systems and Networks, DSN-2004

[14] Philip Koopman, Best CRC Polynomials, https://users.ece.cmu.edu/~koopman/crc/ (Date: 29-Jan-2017)

[15] CRC signature calculator for a seed value of "0": http://www.ghsi.de/pages/subpages/Online%20CRC%20Calculation/index.php?Polynom=10100111010101011&Message=1%0D%0A (Date: 05-Apr-2018)

[16] IO-Link Community, *IO Device Description (IODD)*, V1.1.3, January 2021, Order No. 10.012

[17] IO-Link Community, *IO-Link Common Profile*, V1.1, December 2021, Order No. 10.072

[18] IEC 62453 series: *Field device tool (FDT) interface specification*

[19] FDT Joint Interest Group, *FDT for IO-Link – Annex to FDT Specification – Based on FDT Specification Version 1.2.1,* V1.0, Order No. 0002-0013-000

[20] IO-Link Community, *IO-Link Design Guideline*, V1.0, November 2016, Order No. 10.912

[21] IEC 60947-5-3, *Low-voltage switchgear and controlgear – Part 5-3: Control circuit devices and switching elements – Requirements for proximity devices with defined behaviour under fault conditions*

[22] IEC 61076-2-113, *Connectors for electronic equipment – Product requirements – Part 2-113: Circular connectors – Detail specification for connectors with M12 screw locking with power and signal contacts for data transmission with frequency up to 100 MHz*

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

[24] ISO 14119:2013, *Safety of machinery – Interlocking devices associated with guards – Principles for design and selection*

____

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

IO-Link