

IO-Link Safety System Extensions

Specification

V1.1.5

October 2025

Order No: 10.092

File name: **IO-Link_Safety_System-Extensions_10.092_V1.1.5_Oct2025.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 227.

All specification from the IO-Link Interface specification and test specification is valid unless the deviations are specified in this specification.

Any comments, proposals, requests on this document are appreciated. Please use www.io-link-projects.com for your entries and provide name and email address.

Login: **IOL-Safety11**

Password: **Report**

NOTICE:


The information contained in this document is subject to change without notice. The material in this document details a PNO specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of a PNO specification in any company's products.

The attention of adopters is directed to the possibility that compliance with or adoption of PNO specifications may require use of an invention covered by patent rights. PNO shall not be responsible for identifying patents for which a license may be required by any PNO specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. PNO specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

WHILE THE INFORMATION IN THIS DOCUMENT IS BELIEVED TO BE ACCURATE, PNO MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT INCLUDING, BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR USE.

In no event shall PNO be liable for errors contained herein or for indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party. Compliance with PNO specifications does not absolve manufacturers, from the requirements of safety and regulatory agencies (TÜV, BIA, UL, CSA, etc.).

USE OF TRADEMARKS:

 **IO-Link**® is registered trade mark. The use is restricted for members of the IO-Link Community. More detailed terms for the use can be found in the IO-Link Community Rules on www.io-link.com.

PNO is the owner of several registered trademarks, such as PROFIBUS®, PROFINET®, omlox®, IO-Link®, MTP® and others. More detailed terms for the use can be found on the web page www.profibus.com. Please select buttons "Downloads / Presentations & logos". In some cases, PNO is the licensee of registered trademarks owned by third parties and which may be relevant in regard with products compliant to this document.

PNO shall always be the sole entity that may authorize developers, suppliers and sellers of hardware and software to use certification marks, trademarks or other special designations to indicate compliance with a PNO specification. Products developed using a PNO specification may claim compliance or conformance with a PNO specification only if the hardware and/or software satisfactorily meets the certification requirements set by PNO. Products that do not meet these requirements may claim only that the product was based on a PNO specification and must not claim compliance or conformance with a PNO specification.

COPYRIGHT

Copyright © 2025 PROFIBUS Nutzerorganisation e.V.

Any unauthorized use of this publication may violate Copyright Law, Trademark Law and other legal regulations. This document contains information which is protected by Copyright. No part of this work covered by Copyright herein may be reproduced or used in any form or by any means -graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the publisher.

Publisher:

IO-Link Community

c/o PROFIBUS Nutzerorganisation e.V.

Ohiostrasse 8

76149 Karlsruhe

Germany

Phone: +49 721 / 98 61 97 0

Fax: +49 721 / 98 61 97 11

E-mail: info@io-link.com

Web site: www.io-link.com

LICENSE AGREEMENT

1. License

1.1 Subject of this license agreement is this document issued by the Licensor, in electronic form. If applicable, also software may be provided.

1.2 The user of this document (Licensee) acquires the license solely from PROFIBUS Nutzerorganisation e.V., having its principal place of business in Karlsruhe, Germany (hereinafter referred to as "Licensor").

1.3 This document is not an industrial standard acknowledged by any standardization body or otherwise and may be further enhanced.

2. Rights and Duties of Licensee

2.1 Licensor hereby grants to Licensee the right to use this document exclusively for developing and supporting products compliant with this document. Licensee may copy this document for this purpose and for data backup purposes.

2.2 Licensee shall not be entitled to modify, decompile, reverse engineer or extract any individual parts of this document, unless this is permitted by mandatory Copyright Law. Furthermore, Licensee shall not be entitled to remove any alphanumeric identifiers, trademarks or copyright notices from this document and, insofar as Licensee is entitled to make copies of this document, Licensee shall copy them without alteration.

2.3 Licensee is not entitled to copy and redistribute this document to any third party, except for "Have Made" purposes. All copies must be obtained on an individual basis, directly from the website www.profibus.com or upon request from the Licensor.

2.4. Licensee agrees to comply with all applicable export control, trade, and sanctions regulations as well as embargo regulations of the European Union, the United States of America, and other relevant jurisdictions ("Export Control Regulations"). This applies in particular to the applicable Export Control Regulations relating to Russia and Belarus and with natural or legal persons associated with Russia and Belarus. Should Licensor become aware of any violation of Export Control Regulations with respect to the use of this document, PNO reserves the right to terminate this Agreement without notice and claim damages.

3. Liability of Licensor

3.1 Licensor shall have no obligation to enhance the document and shall assume no liability in case the document or future versions thereof shall not be approved as an industrial standard.

3.2 Licensor's liability for defects as to quality or title of this document, especially in relation to the correctness or absence of defects or the absence of claims or third-party rights or in relation to completeness, usability and/or fitness for purpose are excluded, except for cases involving gross negligence, willful misconduct or fraudulent concealment of a defect.

3.3 Any further liability is excluded unless required by law, e.g. in cases of personal injury or death, willful misconduct, gross negligence, or in case of breach of fundamental contractual obligations. The damages in case of breach of fundamental contractual obligations is limited to the contract-typical, foreseeable damage if there is no willful misconduct or gross negligence.

4. Place of Jurisdiction and Applicable Law

4.1 The sole place of jurisdiction shall be the principal place of business of Licensor.

4.2 All relations arising out of the contract shall be governed by the substantive law of Germany, to the exclusion of the United Nations Convention on Contracts for the International Sale of Goods (CISG).

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.
highly recommended:	indicates that a feature shall be implemented except for well-founded cases. Vendor shall document the deviation within the user manual and within the manufacturer declaration.

CONTENTS

1. License	3
2. Rights and Duties of Licensee	3
3. Liability of Licensor	3
4. Place of Jurisdiction and Applicable Law	3
Conventions:	3
INTRODUCTION	13
1 Scope	15
2 Normative references	15
3 Terms, definitions, symbols, abbreviated terms, and conventions	16
3.1 Terms and definitions	16
3.2 Common terms and definitions	16
3.3 Terms and definitions related to SDCI-FS	19
3.4 Symbols and abbreviated terms	20
3.5 Conventions	22
3.5.1 Behavioral descriptions	22
3.5.2 Memory and transmission octet order	22
4 Overview of SDCI-FS	22
4.1 Purpose of the technology and feature levels	22
4.1.1 Base SDCI-FS technology	22
4.1.2 From "analog" and "switching" to communication	23
4.1.3 Minimized paradigm shift from FS-DI to FS-Master	24
4.1.4 Following the SDCI paradigm (SIO vs. OSSDe/FS-DI)	25
4.1.5 Port class B	27
4.1.6 "USB-Master" with safety parameterization	27
4.1.7 Interoperability matrix of safety devices	28
4.2 Positioning within the automation hierarchy	28
4.3 Wiring, connectors, and power supply	29
4.4 Relationship to SDCI	29
4.5 Communication features and interfaces	30
4.6 Parameterization	30
4.7 Role of FS-Master and FS-Gateway	30
4.8 Mapping to upper-level systems	31
4.9 Structure of the document	31
5 Extensions to the Physical Layer (PL)	31
5.1 Overview	31
5.2 Extensions to PL services	32
5.2.1 PL_SetMode	32
5.2.2 PL_Ready	32
5.3 Transmitter/receiver	32
5.3.1 Assumptions for the expansion to OSSDe	32
5.3.2 OSSDe specifics	33
5.3.3 Start-up of an FS-Device (Ready pulse)	36
5.3.4 Extensions to electric characteristics of a receiver in FS-Device and FS-Master	37
5.4 Extensions to electric and dynamic characteristics of an FS-Device	37

48	5.5	Extensions to electric and dynamic characteristics of an FS-Master Port (FS-DI)	37
49			
50	5.6	FS-Master Port FS-DI interface	38
51	5.7	Wake-up coordination	39
52	5.8	Fast start-up	39
53	5.9	Power supply	39
54	5.10	Medium	40
55	5.10.1	Constraints	40
56	5.10.2	Connectors	40
57	5.10.3	Cable characteristics	40
58	6	Extensions to SIO	40
59	7	Extensions to the data link layer (DL)	40
60	7.1	Overview	40
61	7.2	State machine of the FS-Master DL-mode handler	40
62	7.3	State machine of the FS-Device DL-mode handler	42
63	8	Extensions to the Master Configuration Manager (CM)	44
64	9	Extensions of the FS-Device	45
65	9.1	Principle architecture and models	45
66	9.1.1	FS-Device architecture	45
67	9.1.2	FS-Device model	46
68	9.2	Parameter Manager (PM)	46
69	9.3	Process Data Exchange (PDE)	47
70	9.4	Data Storage (DS)	47
71	9.4.1	General considerations and extensions including safety	47
72	9.4.2	Backup levels	47
73	10	Extensions of the FS-Master	48
74	10.1	Principle architecture	48
75	10.2	SMI service extensions	49
76	10.2.1	Overview	49
77	10.2.2	SMI_FSMasterAccess	50
78	10.2.3	SMI_SPDUIn	51
79	10.2.4	SMI_SPDUOut	52
80	10.2.5	SMI_FSPDInOut	52
81	10.2.6	SMI_PDIn	53
82	10.2.7	SMI_PDOut	53
83	10.3	ArgBlock extensions	53
84	10.3.1	Overview	53
85	10.3.2	FSMasterAccess	54
86	10.3.3	FSCPAAuthenticity	54
87	10.3.4	FSPortConfigList	54
88	10.3.5	FSPortStatusList	56
89	10.3.6	SPDUIn	58
90	10.3.7	SPDUOut	58
91	10.3.8	FSPDInOut	59
92	10.4	Safety Layer Manager (SLM)	60
93	10.4.1	Purpose	60
94	10.4.2	FS_PortModes	60
95	10.4.3	FSP parameter	60

96	10.5	Process Data Exchange (PDE)	63
97	10.6	Data Storage (DS)	65
98	11	Safety communication layer (SCL)	65
99	11.1	Functional requirements	65
100	11.2	Communication errors and safety measures	65
101	11.3	SCL services	66
102	11.3.1	Positioning of safety communication layers (SCL)	66
103	11.3.2	FS-Master SCL services	66
104	11.3.3	FS-Device SCL services	68
105	11.4	SCL protocol	69
106	11.4.1	Protocol phases to consider	69
107	11.4.2	FS-Device faults	70
108	11.4.3	Safety PDU (SPDU)	70
109	11.4.4	FS-Input and FS-Output data	71
110	11.4.5	Port number	71
111	11.4.6	Status and control	71
112	11.4.7	CRC signature	72
113	11.4.8	TADI safety considerations (normative)	73
114	11.4.9	Data types for SDCI-FS	74
115	11.5	SCL behavior	75
116	11.5.1	General	75
117	11.5.2	SCL state machine of the FS-Master	75
118	11.5.3	SCL state machine of the FS-Device	77
119	11.5.4	Sequence charts for several use cases	81
120	11.5.5	Monitoring of safety times	87
121	11.5.6	Reaction in the event of a malfunction	88
122	11.5.7	Start-up (communication)	90
123	11.6	SCL management	90
124	11.6.1	Parameter overview (FSP and FST)	90
125	11.6.2	Parameterization approaches	91
126	11.7	Safety function response time	92
127	11.7.1	General concepts and accuracies	92
128	11.7.2	Integration Aspects	94
129	11.8	Integrity measures	95
130	11.8.1	IODD integrity	95
131	11.8.2	Tool integrity	95
132	11.8.3	Transmission integrity	95
133	11.8.4	Verification record	95
134	11.8.5	Authentication	96
135	11.8.6	Storage integrity	96
136	11.8.7	FS I/O data structure integrity	98
137	11.8.8	Technology parameter (FST) based on IODD	98
138	11.8.9	Technology parameter (FST) based on existing Dedicated Tool (IOPD)	99
139	11.9	Creation of FSP and FST parameters	99
140	11.10	Integration of Dedicated Tools (IOPD)	100
141	11.10.1	IOPD interface	100
142	11.10.2	Standard interfaces	100
143	11.10.3	Backward channel	101

144	11.11	Validation.....	102
145	11.12	Passivation	102
146	11.12.1	Motivation and means	102
147	11.12.2	Port selective (FS-Master)	102
148	11.12.3	Signal selective (FS-Terminal).....	102
149	11.12.4	Qualifier settings in case of communication	102
150	11.12.5	Qualifier handling in case of FS-DI	103
151	11.13	SCL diagnosis.....	104
152	12	Functional safe processing (FS-P).....	105
153	12.1	Recommendations for efficient I/O mappings	105
154	12.2	Embedded FS controller	105
155	Annex A	(normative) Extensions to parameters	106
156	A.1	Indices and parameters for SDCI-FS.....	106
157	A.2	Parameters in detail.....	107
158	A.2.1	FSP_Authenticity	107
159	A.2.2	FSP_Port.....	107
160	A.2.3	FSP_AuthentCRC	108
161	A.2.4	FSP_ProtVersion	108
162	A.2.5	FSP_ProtMode	108
163	A.2.6	FSP_Watchdog.....	108
164	A.2.7	FSP_IO_StructCRC	108
165	A.2.8	FSP_TechParCRC.....	110
166	A.2.9	FSP_ProtParCRC	110
167	A.2.10	FSP_VerifyRecord	110
168	A.2.11	FSP_TimeToReady.....	110
169	A.2.12	FSP_MinShutDownTime	110
170	A.2.13	FSP_WCDT	110
171	A.2.14	FSP_OFDT	111
172	A.2.15	FSP_ParamDescCRC	111
173	Annex B	(normative) Extensions to EventCodes	112
174	B.1	Additional FS-Device EventCodes.....	112
175	B.2	Additional Port EventCodes	112
176	Annex C	(normative) Extensions to Data Types	114
177	C.1	Data types for SDCI-FS	114
178	C.2	BooleanT (bit)	114
179	C.3	IntegerT (16).....	115
180	C.4	IntegerT (32).....	115
181	C.5	Safety Code	116
182	Annex D	(normative) CRC generator polynomials	117
183	D.1	Overview of CRC generator polynomials	117
184	D.2	Residual error probabilities	117
185	D.3	Implementation considerations.....	119
186	D.3.1	Overview	119
187	D.3.2	Bit shift algorithm (16 bit).....	119
188	D.3.3	Lookup table (16 bit).....	119
189	D.3.4	Bit shift algorithm (32 bit).....	120
190	D.3.5	Lookup table (32 bit).....	121
191	D.3.6	Seed values.....	122

192	D.3.7	Octet order for CRC calculation	123
193	Annex E (normative)	IODD extensions	124
194	E.1	General.....	124
195	E.2	Schema	124
196	E.3	IODD constraints	124
197	E.3.1	General rules.....	124
198	E.3.2	Description of the IODD structure	124
199	E.3.3	Behavior of "Reset" SystemCommands in SDCI-FS	125
200	E.3.4	Profile Characteristic	125
201	E.3.5	ProcessDataInput and ProcessDataOutput	125
202	E.4	IODD conventions.....	125
203	E.4.1	Naming.....	125
204	E.4.2	Process Data (PD).....	125
205	E.4.3	IODD conventions for user interface	126
206	E.4.4	Master Tool features.....	126
207	E.5	Securing	126
208	E.5.1	General	126
209	E.5.2	DefaultValues for FSP	127
210	E.5.3	FSP_Authenticity	127
211	E.5.4	FSP_Protocol	127
212	E.5.5	FSP_IO_Description	128
213	E.5.6	Sample serialization for FSP_ParamDescCRC	128
214	E.5.7	FST and FSP parameters and Data Storage	129
215	E.5.8	Sample IODD of an FS-Device.....	129
216	Annex F (normative)	Device tool Interface (DTI) for SDCI	130
217	F.1	Purpose of DTI.....	130
218	F.2	DTI Overview	130
219	F.2.1	Functions and Elements of the Invocation Interface	130
220	F.2.2	Functions and elements of the Communication Interface	131
221	F.2.3	Security	131
222	F.3	Dedicated Tool Invocation interface	131
223	F.3.1	Overview	131
224	F.3.2	Detection of Dedicated Tool.....	132
225	F.3.3	Program Interface Description – PID.....	135
226	F.3.4	Temporary Parameter File – TPF.....	137
227	F.3.5	Temporary Backchannel File – TBF	142
228	F.3.6	Invocation behavior	144
229	F.4	Communication Interface	144
230	F.4.1	General	144
231	F.4.2	Principle of DTI communications.....	145
232	F.4.3	Definition of the Communication Interface.....	146
233	F.4.1	Description of the Communication Interface	146
234	F.5	Reaction on incorrect tool behavior.....	146
235	F.6	Compatibility	147
236	F.6.1	Schema validation	147
237	F.6.2	Version policy	147
238	F.7	Schema definitions for PID, TPF and TBF	147
239	F.7.1	General	147

240	F.7.2	Schema of the PID.....	147
241	F.7.3	Schema of the TPF.....	149
242	F.7.4	Schema of the TBF.....	151
243	F.7.5	Schema of DTI primitives.....	152
244	F.8	Schema Definitions of the Communication Interface.....	152
245	F.8.1	Schema for DTI_DTQ.....	155
246	F.8.2	Schema for DTI_MTR.....	156
247	F.8.3	Schema for DTI Online Channel Base Types.....	157
248	F.9	Yaml file IO-Link DTI OPENAPI.....	159
249	Annex G (normative)	Main scenarios of SDCI-FS.....	161
250	G.1	Overview.....	161
251	G.2	Sequence chart of commissioning.....	163
252	G.3	Sequence chart of replacement.....	164
253	G.4	Sequence chart of misconnection.....	164
254	Annex H (normative)	System requirements.....	165
255	H.1	Indicators.....	165
256	H.1.1	General.....	165
257	H.1.2	FS-DI.....	165
258	H.1.3	Safety communication.....	165
259	H.1.4	FS-Master Tool.....	165
260	H.1.5	Acknowledgment request.....	165
261	H.2	Installation guidelines, electrical safety, and security.....	165
262	H.3	Safety function response time.....	166
263	H.4	Duration of demands.....	166
264	H.5	Maintenance and repair.....	166
265	H.6	Safety manual.....	166
266	Annex I (informative)	Information for test and assessment of SDCI-FS components.....	167
267	Bibliography.....		168
268			
269	Figure 1 – Positioning of SDCI-FS in functional safety automation.....		13
270	Figure 2 – Relationship of this document to standards.....		14
271	Figure 3 – Memory and transmission octet order.....		22
272	Figure 4 – SDCI-FS communication layer model.....		23
273	Figure 5 – Port interface extensions for SDCI-FS.....		23
274	Figure 6 – Migration to SDCI-FS.....		24
275	Figure 7 – Minimized paradigm shift from FS-DI to FS-Master.....		25
276	Figure 8 – FS-Master types and feature levels.....		25
277	Figure 9 – Original pin layout of SDCI including IO-Link Safety extensions (Port class		
278	A) 26		
279	Figure 10 – Level "d" of an FS-Master (Class B).....		27
280	Figure 11 – Off-site configuration and parameterization.....		28
281	Figure 12 – SDCI-FS within the automation hierarchy.....		29
282	Figure 13 – The SDCI physical layer of an FS-Master (class A).....		31
283	Figure 14 – The physical layer of an FS-Device (class A).....		32
284	Figure 15 – Cross compatibility OSSD and OSSDe.....		33
285	Figure 16 – Principle OSSDe function.....		34

286	Figure 17 – Test pulses to detect cross connection faults	35
287	Figure 18 – OSSD timings	36
288	Figure 19 – Typical start-up of an OSSD sensor	36
289	Figure 20 – Start-up of an FS-Device	36
290	Figure 21 – Charge capability at power-up	38
291	Figure 22 – FS-DI input filter conflict resolution	38
292	Figure 23 – Start-up of an FS-Device	39
293	Figure 24 – Required fast start-up timings	39
294	Figure 25 – State machine of the FS-Master DL-mode handler	41
295	Figure 26 – State machine of the FS-Device DL-mode handler	43
296	Figure 27 – Extension to the Configuration Manager (VerifyRecord)	44
297	Figure 28 – Principle architecture of the FS-Device	45
298	Figure 29 – The FS-Device model	46
299	Figure 30 – Principle architecture of the FS-Master	48
300	Figure 31 – SMI service extensions	50
301	Figure 32 – FSP parameter use cases	61
302	Figure 33 – PDE Splitter (only in IO-Link PortMode = “SAFETYCOM”)	64
303	Figure 34 – PDE Composer (only in IO-Link PortMode = “SAFETYCOM”)	64
304	Figure 35 – Positioning of the SDCI-FS Safety Communication Layer (SCL)	66
305	Figure 36 – FS-Master Safety Communication Layer services	67
306	Figure 37 – FS-Device Safety Communication Layer services	68
307	Figure 38 – Protocol phases to consider	70
308	Figure 39 – Safety PDUs of FS-Master and FS-Device	71
309	Figure 40 – The 1 % share rule of IEC 61784-3:2021	73
310	Figure 41 – SCL state machine of the FS-Master	75
311	Figure 42 – SCL state machine of the FS-Device	78
312	Figure 43 – FS-Master and FS-Device both with power ON	81
313	Figure 44 – FS-Master power OFF → ON	82
314	Figure 45 – FS-Device with delayed SCL start	83
315	Figure 46 – FS-Device with power OFF and ON	84
316	Figure 47 – FS-Master detects CRC signature error	85
317	Figure 48 – FS-Device detects CRC signature error	86
318	Figure 49 – Monitoring of the SCL cycle time	87
319	Figure 50 – Parameter types and assignments	91
320	Figure 51 – FSCP-Host-centric system	92
321	Figure 52 – SFRT of a stand-alone FS-Master with processing	93
322	Figure 53 – SFRT including IOL-S and FSCP	94
323	Figure 54 – Structure of the FSP_VerifyRecord	96
324	Figure 55 – Start-up of SDCI-FS	97
325	Figure 56 – Securing of FST parameters via Dedicated Tool	98
326	Figure 57 – Modification of FST parameters via Dedicated Tool	99
327	Figure 58 – Creation of FSP and FST parameters	100
328	Figure 59 – Example of a communication hierarchy	101

329	Figure 60 – Motivation for Port selective passivation	102
330	Figure 61 – Qualifier handler (communication)	103
331	Figure 62 – Qualifier handler (FS-DI)	103
332	Figure 63 – Qualifier behavior per FS-Master Port	104
333	Figure 64 – Mapping efficiency issues	105
334	Figure A.1 – Instance of an FS I/O data description	109
335	Figure A.2 – Example FS I/O data structure with non-safety data	109
336	Figure A.3 – Securing of safety parameters	111
337	Figure C.1 – Example of a BooleanT data structure	114
338	Figure C.2 – Safety Code of an output message	116
339	Figure C.3 – Safety Code of an input message	116
340	Figure D.1 – CRC-16 generator polynomial	118
341	Figure D.2 – CRC-32 generator polynomial	118
342	Figure D.3 – Bit shift algorithm in "C" language (16 bit)	119
343	Figure D.4 – CRC-16 signature calculation using a lookup table	119
344	Figure D.5 – Bit shift algorithm in "C" language (32 bit)	121
345	Figure D.6 – CRC-32 signature calculation using a lookup table	121
346	Figure E.1 – Algorithm to build the FSP parameter CRC signatures	127
347	Figure F.1 – Principle of DTI invocation interface	132
348	Figure F.2 – Structure of the registry	133
349	Figure F.3 – Example of a DTI registry	133
350	Figure F.4 – Detection of a Dedicated Tool in registry	135
351	Figure F.5 – Menu for Dedicated Tool invocation	136
352	Figure F.6 – Communication routes between Dedicated Tool and Device	145
353	Figure F.7 – REST API Communication	146
354	Figure F.8 – XML schema of the PID file	148
355	Figure F.9 – XML schema TPF	149
356	Figure F.10 – XML schema of a TBF	151
357	Figure F.11 – XML Schema DTI_DTQ	153
358	Figure F.12 – DTI_MTR response	154
359	Figure F.13 – DTI_MTR negative response	155
360	Figure G.1 – Commissioning with test and armed operation	163
361	Figure G.2 – FS-Device replacement	164
362	Figure G.3 – FS-Device misconnection	164
363		
364	Table 1 – Operational modes of feature level "a" to "c" (Port class A)	27
365	Table 2 – Interoperability matrix of safety devices	28
366	Table 3 – PL_Ready	32
367	Table 4 – OSSDe states and conditions	34
368	Table 5 – Cross connection faults	35
369	Table 6 – Extension to electric and dynamic characteristics of the FS-Device (OSSDe)	37
370	Table 7 – Extensions to electric and dynamic characteristics of the Port interface	38
371	Table 8 – Cable characteristics	40

372	Table 9 – State transition tables of the FS-Master DL-mode handler	42
373	Table 10 – State transition tables of the FS-Device DL-mode handler	43
374	Table 11 – State transition tables of the Configuration Manager	44
375	Table 12 – Extension to Data Storage (DS) state machine	47
376	Table 13 – Data Storage Backup Levels	47
377	Table 14 – SMI services used for FS-Master.....	49
378	Table 15 – SMI_FSMasterAccess	50
379	Table 16 – SMI_FSPDInOut.....	52
380	Table 17 – ArgBlock types and ArgBlockIDs	53
381	Table 18 – FSMasterAccess	54
382	Table 19 – FSCPAuthenticity	54
383	Table 20 – FSPortConfigList	54
384	Table 21 – FSPortStatusList	56
385	Table 22 – SPDUIIn	58
386	Table 23 – SPDUIOut	58
387	Table 24 – FSPDInOut.....	59
388	Table 25 – Use case reference table.....	61
389	Table 26 – Communication errors and safety measures	65
390	Table 27 – SCL services of FS-Master	67
391	Table 28 – SCL services of FS-Device	69
392	Table 29 – Protocol phases to consider	70
393	Table 30 – Control and counting (Control&MCnt)	71
394	Table 31 – Status and counting mirror (Status&DCnt)	72
395	Table 32 – MCount and DCount_i values	72
396	Table 33 – FS process I/O data types	74
397	Table 34 – Rules for the layout of values and qualifiers	74
398	Table 35 – Order of values and qualifier	75
399	Table 36 – Definition of terms used in SCL state machine of the FS-Master.....	76
400	Table 37 – FS-Master SCL states and transitions	76
401	Table 38 – Definition of terms used in SCL state machine of the FS-Device.....	79
402	Table 39 – FS-Device SCL states and transitions	79
403	Table 40 – Timing constraints	87
404	Table 41 – Accuracies and tolerances for timings	93
405	Table 42 – restore from Data Storage	97
406	Table 43 – Qualifier bits "GOOD/BAD"	103
407	Table 44 – State transition table for the qualifier behavior.....	104
408	Table A.1 – Indices for SDCI-FS	106
409	Table A.2 – Coding of protocol version	108
410	Table A.3 – Coding of protocol mode	108
411	Table A.4 – Generic FS I/O data structure description	109
412	Table B.1 – FS-Device SCL specific EventCodes	112
413	Table B.2 – FS-Master SCL specific EventCodes	112
414	Table C.1 – Data types for SDCI-FS	114

415	Table C.2 – BooleanT for SDCI-FS	114
416	Table C.3 – Example of BooleanT within a RecordT	114
417	Table C.4 – IntegerT(16)	115
418	Table C.5 – IntegerT(16) coding	115
419	Table C.6 – IntegerT(32)	115
420	Table C.7 – IntegerT(32) coding	115
421	Table D.1 – CRC generator polynomials for SDCI-FS	117
422	Table D.2 – Definition of variables used in Figure D.3	119
423	Table D.3 – Definition of variables used in Figure D.4	119
424	Table D.4 – Lookup table for CRC-16 signature calculation	120
425	Table D.5 – Definition of variables used in Figure D.5	121
426	Table D.6 – Definition of variables used in Figure D.4	121
427	Table D.7 – Lookup table for CRC-32 signature calculation	121
428	Table E.1 – Specific behavior of FS-Device "Reset" SystemCommands	125
429	Table E.2 – User actions to replace DefaultValues	127
430	Table E.3 – RecordItems of FSP_Protocol where allowed values shall be serialized	127
431	Table E.4 – Sample serialization for FSP_ParamDescCRC	128
432	Table F.1 – Description of PID file elements	136
433	Table F.2 – Elements of a TPF	137
434	Table F.3 – Elements of the TBF	142
435	Table F.4 – Invocation cases and behaviors	144
436	Table F.5 – Reaction on incorrect tool behavior	146
437	Table G.1 – Main scenarios of SDCI-FS	161
438		

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.

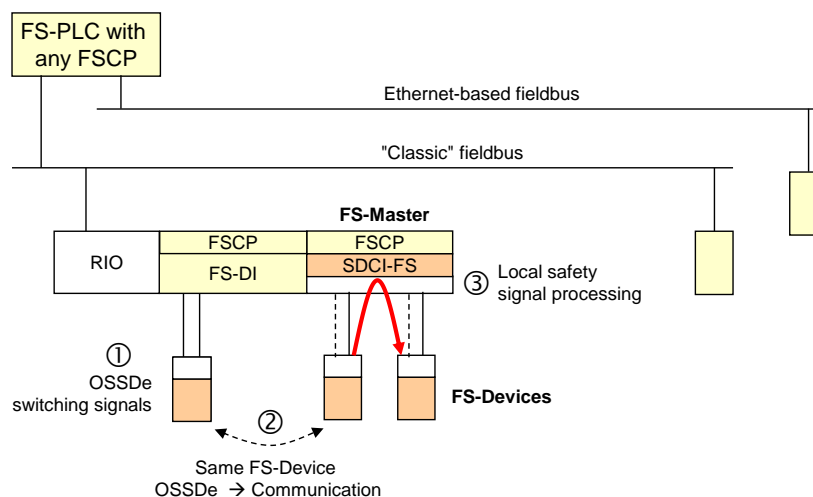


Figure 1 – Positioning of SDCI-FS in functional safety automation

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

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".

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 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 FS-DI functionality. 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 definition for the use 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.

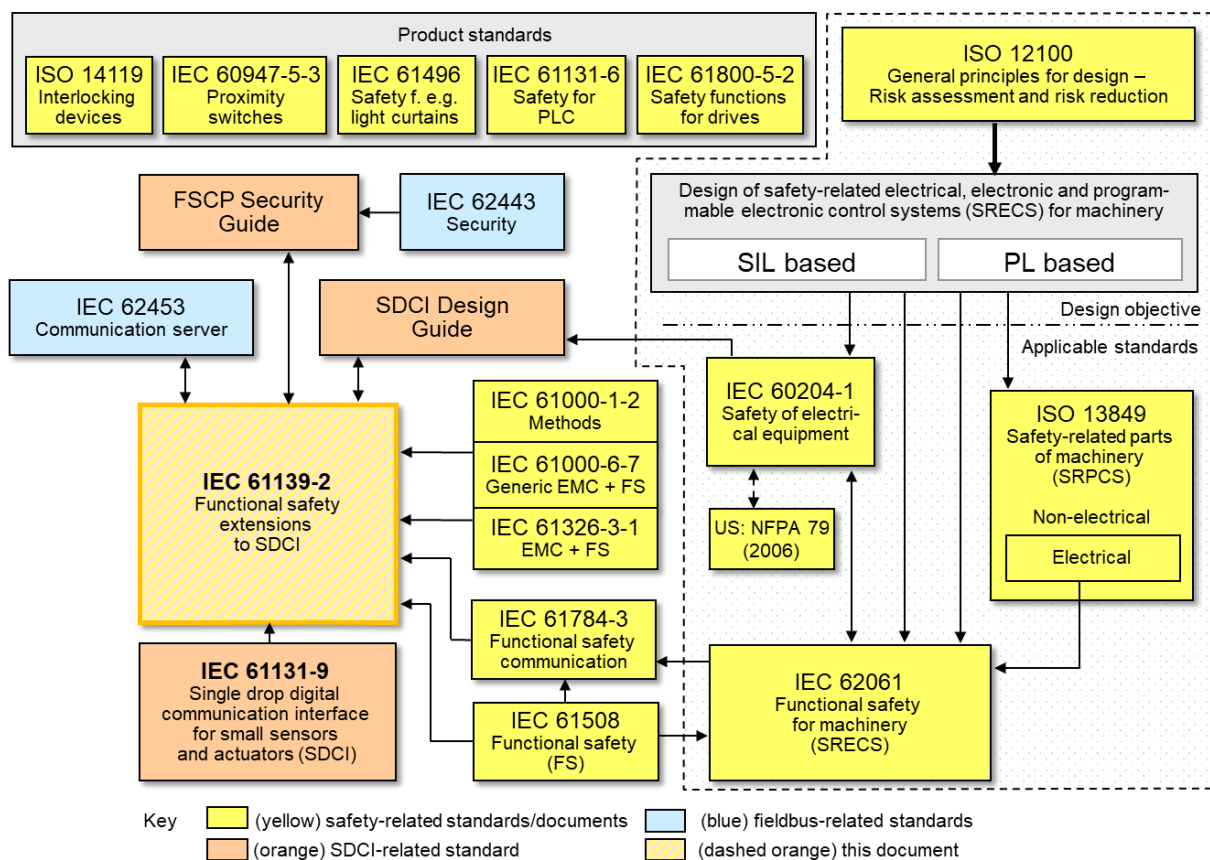


Figure 2 – Relationship of this document to standards

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 PL_e.

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 defined 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 61131-9:2022², *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*

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

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

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

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

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

543 **3 Terms, definitions, symbols, abbreviated terms, and conventions**

544 **3.1 Terms and definitions**

545 For the purposes of this document, the terms and definitions given in IEC 61131-1 and
546 IEC 61131-2 apply. ISO and IEC maintain terminological databases for use in standardization
547 at the following addresses:

- 548 • IEC Electropedia: available at <https://www.electropedia.org/>
- 549 • ISO Online browsing platform: available at <https://www.iso.org/obp/ui>

550 **3.2 Common terms and definitions**

551 **3.2.1** 552 **address**

553 part of the M-sequence control to reference data within data categories of a communication
554 channel

555 **3.2.2** 556 **application layer** 557 AL

558 <SDCI>³ part of the protocol responsible for the transmission of Process Data objects and On-
559 request Data objects

560 **3.2.3** 561 **block parameter** 562 consistent parameter access via multiple Indices or Subindices

563 **3.2.4** 564 **checksum** 565 <SDCI> complementary part of the overall data integrity measures in the data link layer in 566 addition to the UART parity bit

567 **3.2.5** 568 **coded switching** 569 SDCI communication, based on the standard binary signal levels of IEC 61131-2

570 **3.2.6** 571 **COM1** 572 SDCI communication mode with transmission rate of 4,8 kbit/s

573 **3.2.7** 574 **COM2** 575 SDCI communication mode with transmission rate of 38,4 kbit/s

576 **3.2.8** 577 **COM3** 578 SDCI communication mode with transmission rate of 230,4 kbit/s

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

3.2.9**COMx**

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

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: IEC 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 initiative of the German automotive manufacturers)	
AL	application layer	
BEP	bit error probability	
C/Q	connection for communication (C) or switching (Q) signal (SIO, OSSD1e)	
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 series)	
FS-AI/AO	functional safety analog input/analog output	
FS-DI/DO	functional safety digital input/digital output	
FSP	functional safety protocol (parameter)	
FST	functional safety technology (parameter)	
FS-Input	functional safety input data (useable netto safe input data)	
FS-Output	functional safety output data (useable netto safe output data)	
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	

NC	not connected	
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 assignment 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	[https://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].

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.

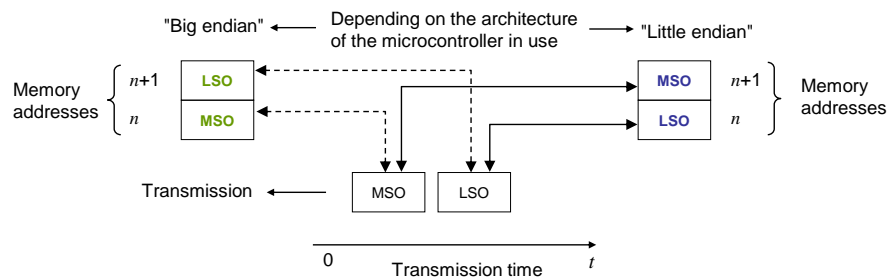


Figure 3 – Memory and transmission octet order

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. Deviations from the original SDCI design are explicitly mentioned. If not mentioned explicitly, all rules of the original standard design apply.

⁴ Numbers in square brackets refer to the bibliography.

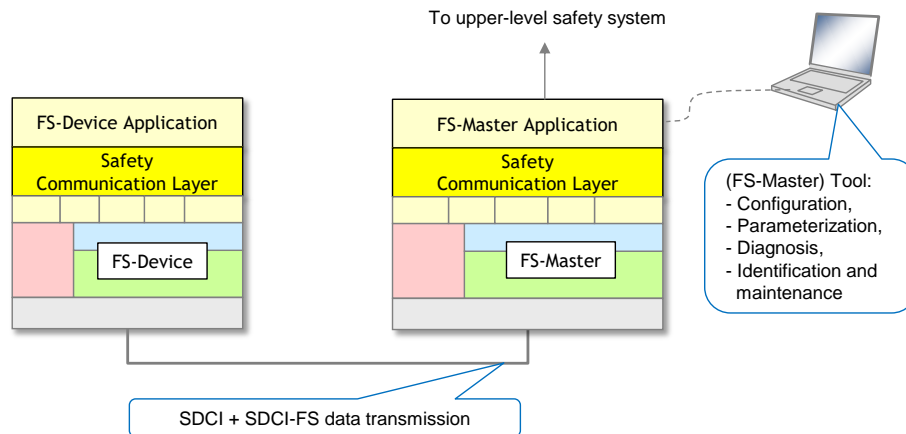


Figure 4 – SDCI-FS communication layer model

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

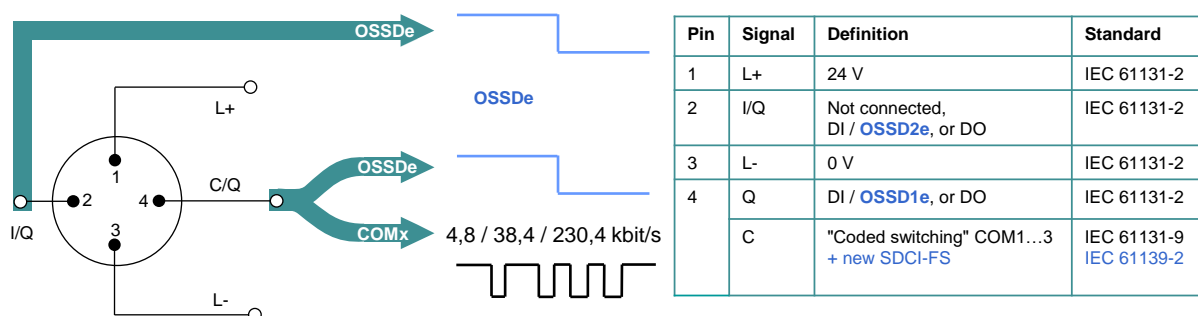
- wireless connections between FS-Master and FS-Device (see Clause H.2),
- cascaded FS-Master/FS-Device systems.

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 concepts described in 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).



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).

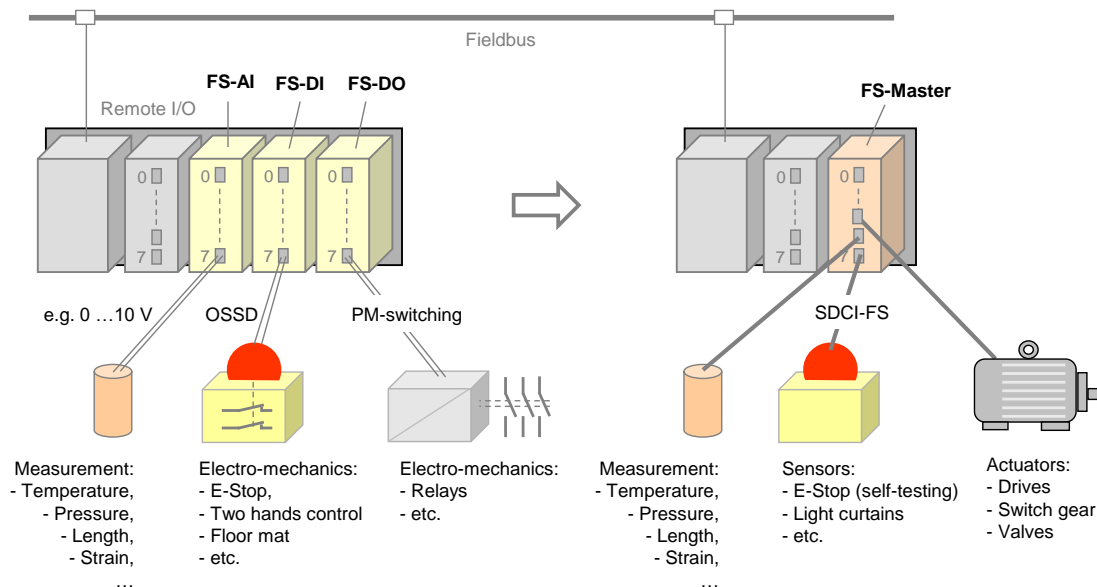


Figure 6 – Migration to SDCI-FS

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

- setup an *authenticity code switch* or *adequate software solution*,
- assign a *watchdog time*,
- 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.

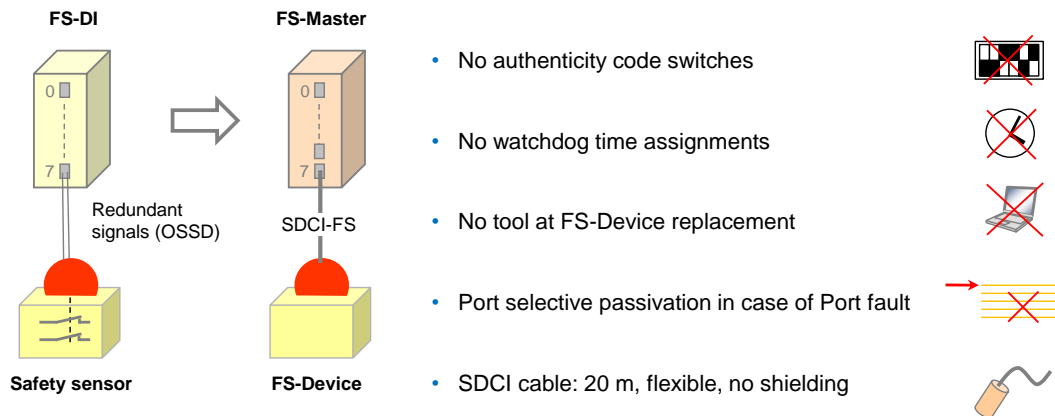


Figure 7 – Minimized paradigm shift from FS-DI to FS-Master

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 8 and IEC 61131-9).

4.1.4 Following the SDCI paradigm (SIO vs. OSSDe/FS-DI)

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.

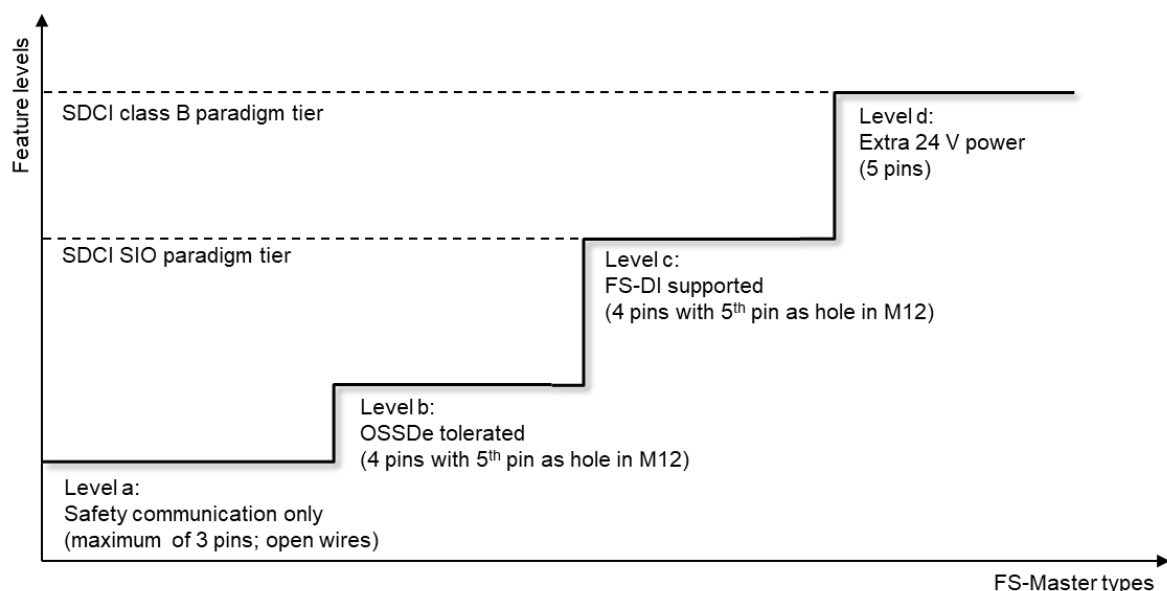


Figure 8 – FS-Master types and feature levels

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.

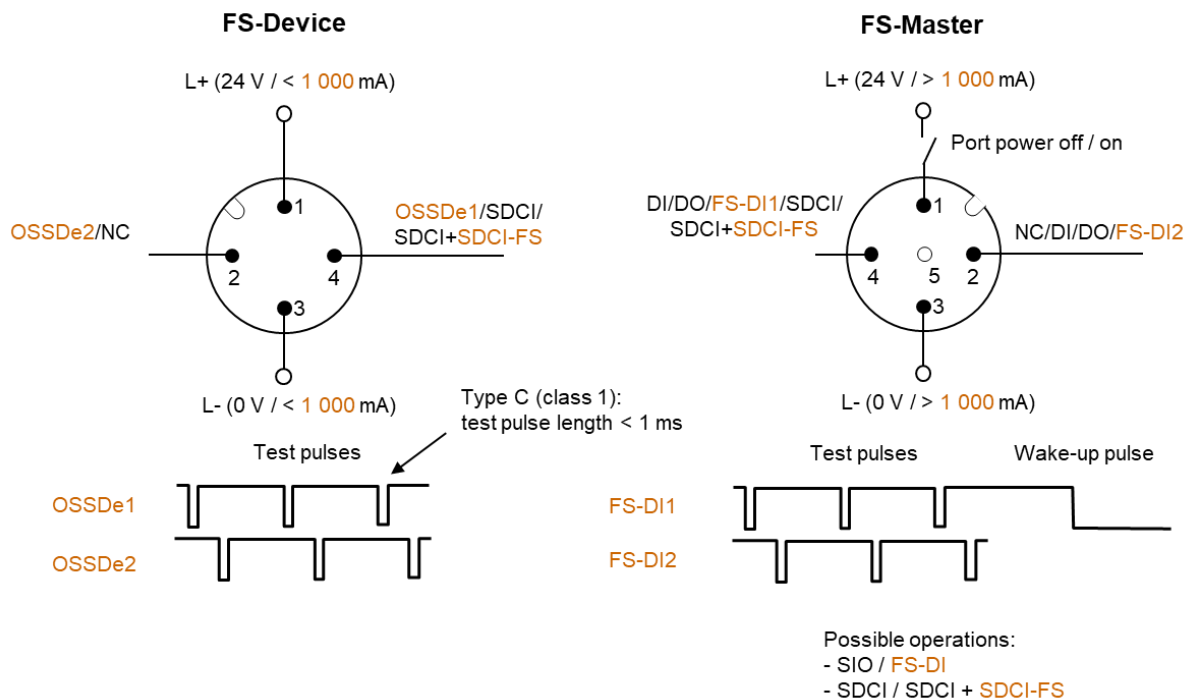


Figure 9 – Original pin layout of SDCI including IO-Link Safety extensions (Port class A)

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. OSSDe2e) besides pin 4 for the primary signal path (e.g. OSSDe1e)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.

Level "c" corresponds to the SIO level of standard SDCI Master. In this case, the FS-Master supports an FS-DI 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.

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

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

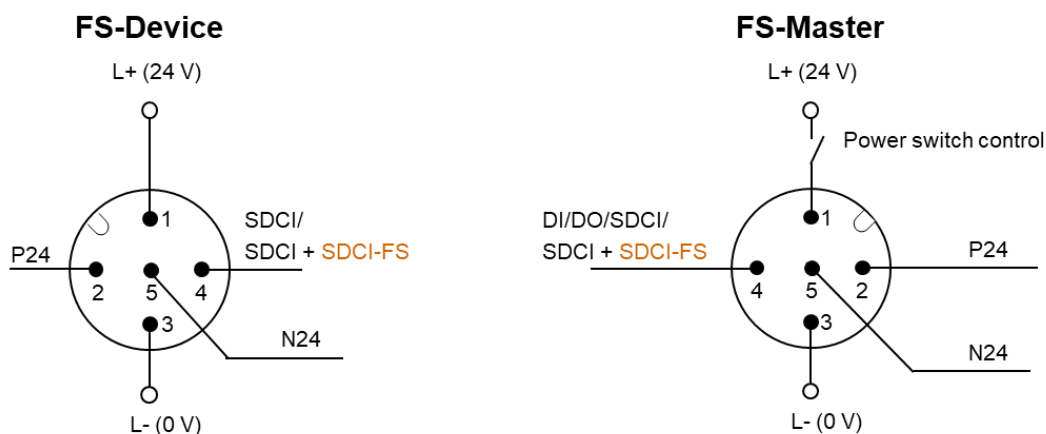
Feature level	FS-Device		FS-Master	
	Pin 2	Pin 4	Pin 2	Pin 4
"a"	- NC (M)	- SDCI (M) - SDCI-FS (M)	- NC (M)	- DI (M) - DO (M) - SDCI (M) - SDCI-FS (M)
"b"			- NC (O) - DI (O) - DO (O)	- DI (M) - DO (M) - SDCI (M) - SDCI-FS (M)
"c"	- OSSD2e (M)	- OSSD1e (M) - SDCI (M) - SDCI-FS (M)	- DI (O) - DO (O) - FS-DI2 (M)	- DI (M) - DO (M) - FS-DI1 (M) - SDCI (M) - SDCI-FS (M)
NC =not connected, physically disconnected Key M = mandatory; O = optional				

FS-Devices of feature level "a" could be connected to every FS-Master level ("a" to "c").

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

**Figure 10 – 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 11.

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).

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

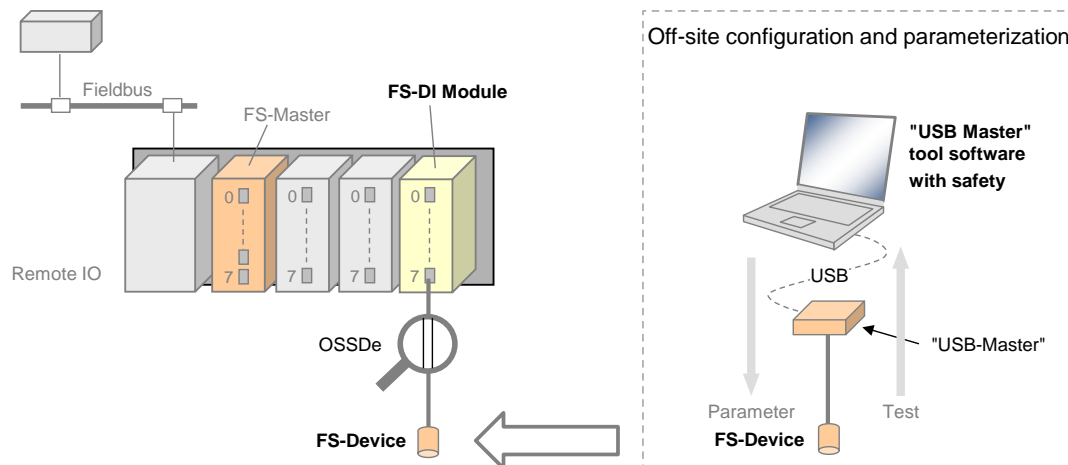


Figure 11 – Off-site configuration and parameterization

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 parameterization, and conventional FS-DI modules connected to FSCPs.

Table 2 – Interoperability matrix of safety devices

Device type	FS-Master			"USB-Master" with safety parameterization	FS-DI module (FSCP)
	Communication "a"	OSSDe tolerated "b"	OSSDe supported "c"		
Sensor with OSSDe	-	-	OSSDe	-	OSSDe
Sensor with OSSDe and SDCI	-	-	OSSDe	SDCI ^a	OSSDe
Sensor with OSSDe and SDCI-FS	SDCI-FS	SDCI-FS	OSSDe or SDCI-FS	SDCI	OSSDe
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		USB = Universal Serial Bus, currently the most common interface amongst possible others for offsite parameterization tools due to fast communication combined with power supply			
a Pin layout may differ					

4.2 Positioning within the automation hierarchy

Figure 12 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.

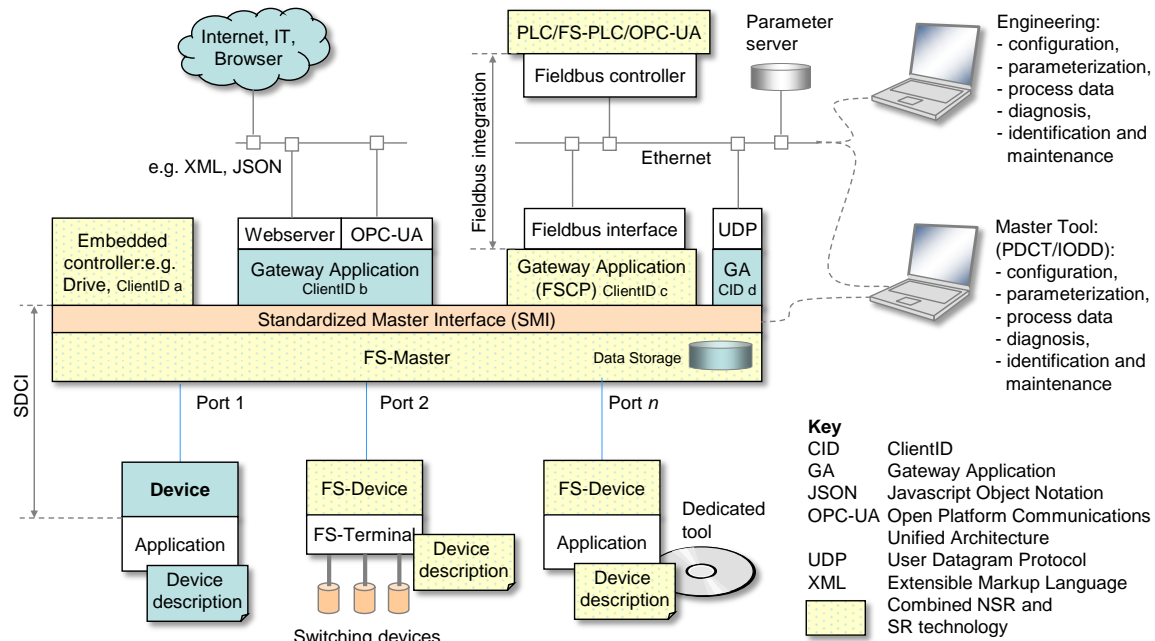


Figure 12 – SDCI-FS within the automation hierarchy

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.

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 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/FS-DI 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 13 shows the adapted physical layer of an FS-Master (class A).

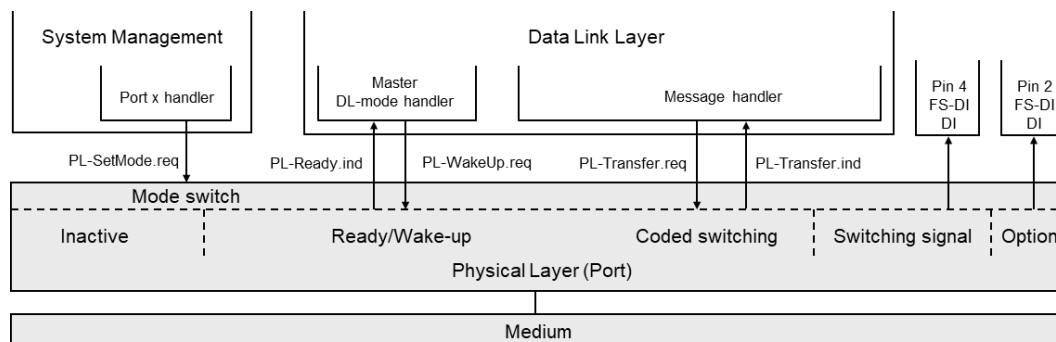


Figure 13 – The SDCI physical layer of an FS-Master (class A)

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

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

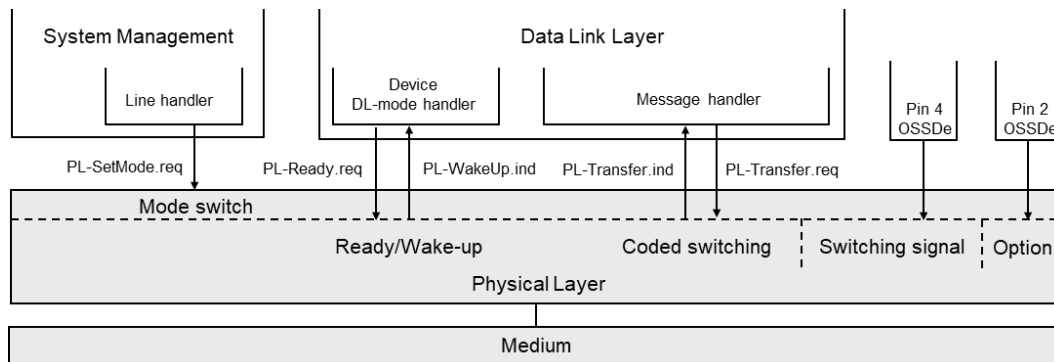


Figure 14 – The physical layer of an FS-Device (class A)

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 "FS-DI" (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 15 shows the cross compatibility between OSSD based safety sensors and OSSDe based FS-Devices.

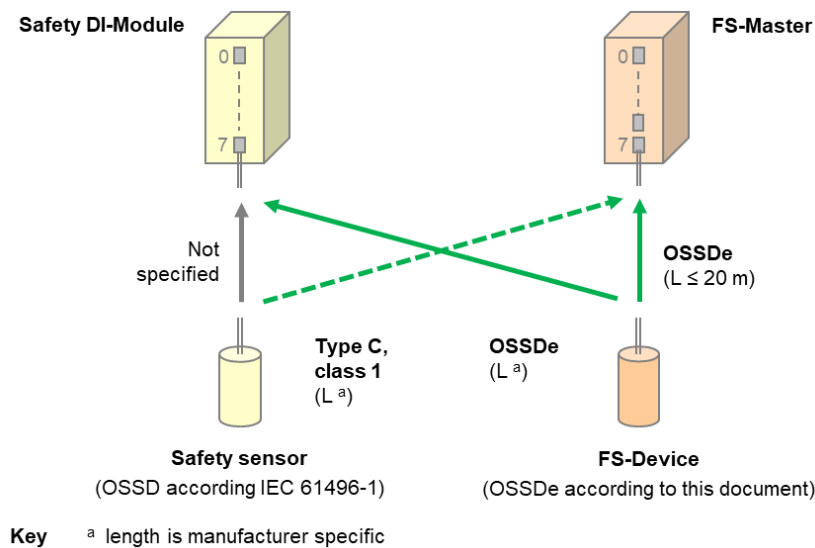


Figure 15 – Cross compatibility OSSD and OSSDe

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 FS-DI 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 an exactly specified interface.
- The FS-DI 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.

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.

Figure 16 shows a corresponding reference model 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.

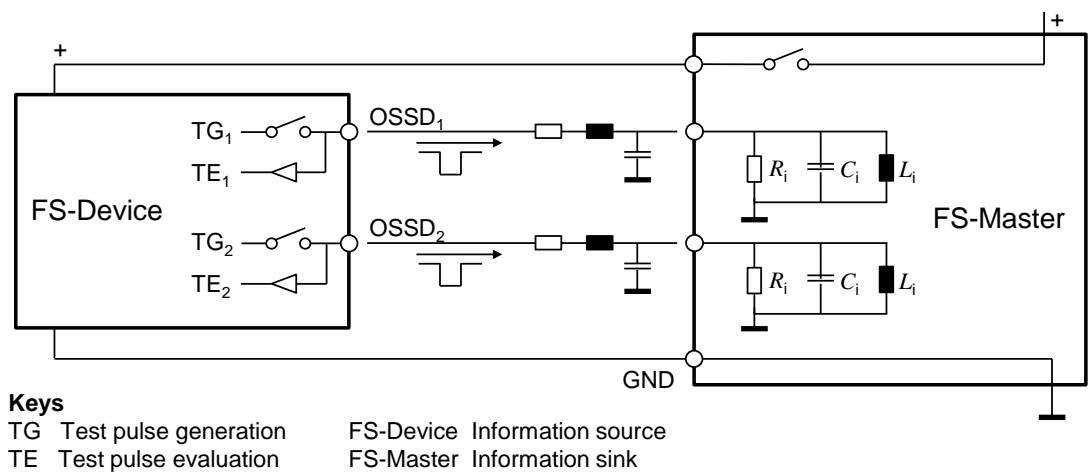


Figure 16 – Principle OSSDe function

The worst-case values for the line resistance and capacitance are defined in Table 8. 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 7.

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

Table 4 – OSSDe 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	> 2 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 17 shows the test pulses approach for the detection of cross connection faults. A cross fault occurs if at least one channel detects signal high during the test pulse.

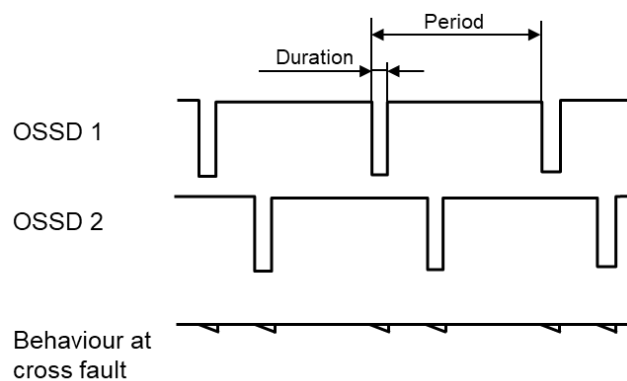


Figure 17 – Test pulses to detect cross connection faults

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 18).

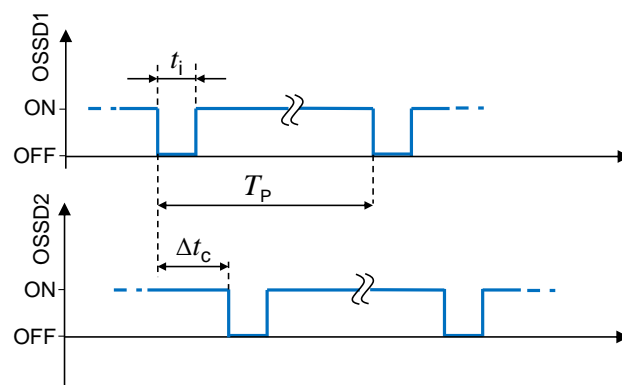


Figure 18 – OSSD timings

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)

FS-Devices shall implement a test pulse length $t_i \leq 1000 \mu s$ (see Table 6).

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

Figure 19 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.

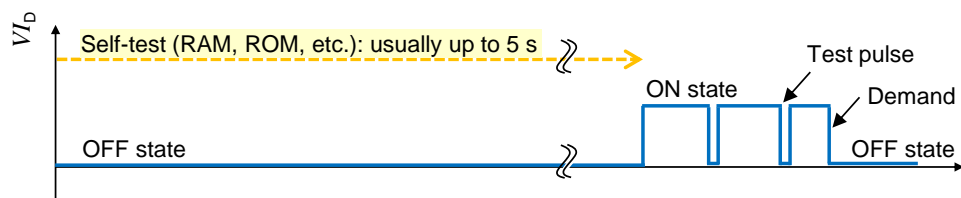


Figure 19 – Typical start-up of an OSSD sensor

Figure 20 shows the start-up of an FS-Device with OSSDe capability connected to a classic FS-DI module.

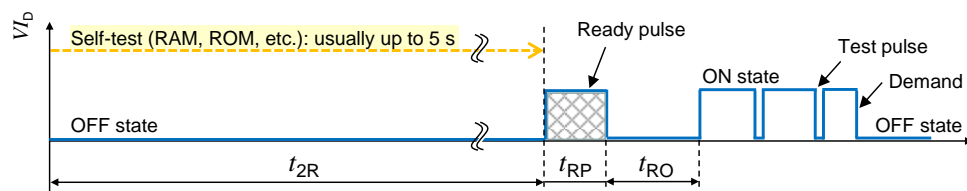


Figure 20 – Start-up of an FS-Device

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 Extensions to electric characteristics of a receiver in FS-Device and FS-Master

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

5.4 Extensions to electric and dynamic characteristics of an FS-Device

In general, the specified values and ranges of IEC 61131-9 apply.

The electric and dynamic parameters for the OSSDe interface of an FS-Device are specified in accordance with IEC 61131-9. Extensions see Table 6.

Table 6 – Extension to electric and dynamic characteristics of the FS-Device (OSSDe)

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
IS_D	DC current on L+	n/a	n/a	1000	mA	See 5.9 See NOTE 3
t_{2R}	Time to Ready pulse	n/a	n/a	5 (default), > 5 to 327,67 as exception	s	See Figure 20 and A.2.11
t_{RP}	Duration of Ready pulse	500	n/a	1000	µs	See Figure 20
t_{RWD}	End of Ready pulse to ready for Wake-up	n/a	n/a	50	µs	See Figure 23
t_{RO}	End of Ready pulse to OSSD mode	1,1	n/a	Data sheet	s	See Figure 20
T_P / t_i	Test pulse period factor	100	n/a	n/a	–	See Figure 18 NOTE 4
t_i	Test pulse duration	n/a	n/a	1000	µs	See Figure 18 NOTE 5
Δt_C	Time-shift	0	n/a	Data sheet	ms	See Figure 18
t_{disD}	Discrepancy time	n/a	n/a	Data sheet	–	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 interface type 1 according to IEC 61131-2						
NOTE 3 The average DC current on L+ must not exceed 1000 mA. During the switching slopes (0.2 * Tbit) on C/Q an overcurrent on L+ must not exceed 2000 mA. Transients on the L+ current are in general allowed to be 1500 mA for a maximum of 5 µs.						
NOTE 4 If Test pulse period factor is less than 100 it shall be described in the data sheet.						
NOTE 5 Characteristics assume OSSD type "C", class "1" in [6]						

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)).

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

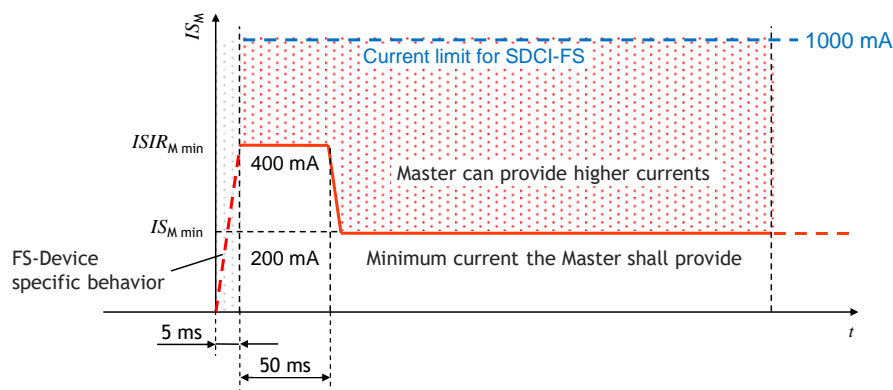
5.5 Extensions to electric and dynamic characteristics of an FS-Master Port (FS-DI)

In general, the specified values and ranges of IEC 61131-9 apply. The definitions in Table 7 extend these for the electrical characteristics of an FS-Master Port.

Table 7 – Extensions to electric and dynamic characteristics of the Port interface

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
I_{SM}	Supply current for FS-Devices (DC current on L+)	200	n/a	1000	mA	Rules in 5.9. shall be considered
t_{RWM}	End of Ready pulse to ready for Wake-up	50	n/a	500	μs	See Figure 23
t_i	Test pulse duration	n/a	n/a	1	ms	See Figure 18 NOTE 1 NOTE 2
NOTE 1 Test pulses are generated by the OSSD device. The FS-Master has to consider this value for an appropriate input filter design. (See 5.6)						
NOTE 2 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 21). After 50 ms the current limitations for I_{SM} in Table 7 apply.

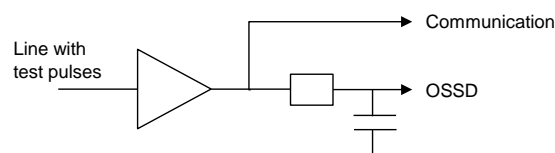
**Figure 21 – Charge capability at power-up**

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 22.

**Figure 22 – FS-DI 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 7.

The following standards shall apply

- ISO 13849

- IEC 62061

5.7 Wake-up coordination

Figure 23 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 19).

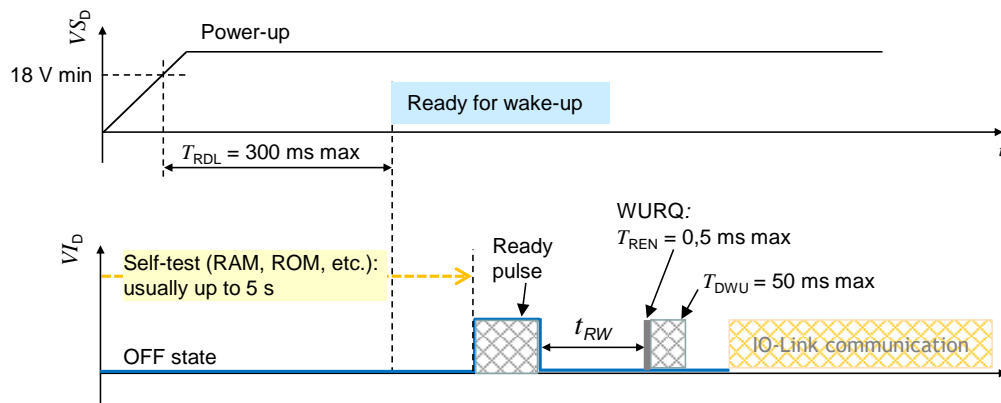


Figure 23 – Start-up of an FS-Device

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 24 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.

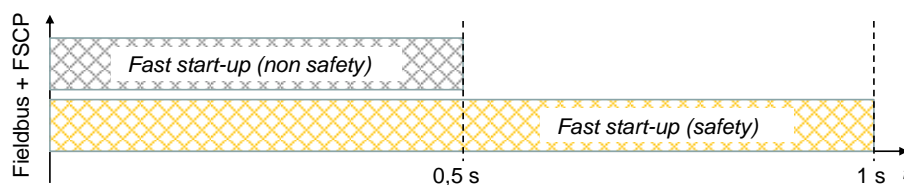


Figure 24 – Required fast start-up timings

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 6 shall be considered. The overall cable loop resistance shall be not more than 1,2 Ω .

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 the FS-Master port and the FS-Device port are permitted.

5.10.2 Connectors

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

5.10.3 Cable characteristics

Table 8 shows the cable characteristics for SDCI-FS and non-safety Devices if higher power supply currents than 200 mA are applied.

Table 8 – 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 28 and Figure 30).

For FS-Devices the meaning of SIO differs from IEC 61131-9. SIO means OSSDe without the IO-Link fallback mechanism. To achieve this the following rules apply:

- FS-Devices which implement OSSDe shall provide this information in the IODD using the attribute `sioSupported="true"`.
- Independent from OSSDe support all FS-Devices shall report "SIO mode not supported" to the IO-Link Master as described in B.1.6 of IEC 61131-9 to avoid fallback.

7 Extensions to the data link layer (DL)

7.1 Overview

Figure 28 and Figure 30 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 25 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.

1220 The maximum waiting time is t_{2R} as defined in Table 6. Whenever the time expired, the FS-
 1221 Master shall run a power-OFF/ON cycle for the connected FS-Device in order to initiate a retry
 1222 for another Ready-pulse.

1223 The criterion to use the extra path is the guard [safety], which is derived from the new Port
 1224 configuration "FS_PortModes" (see 10.4.2).

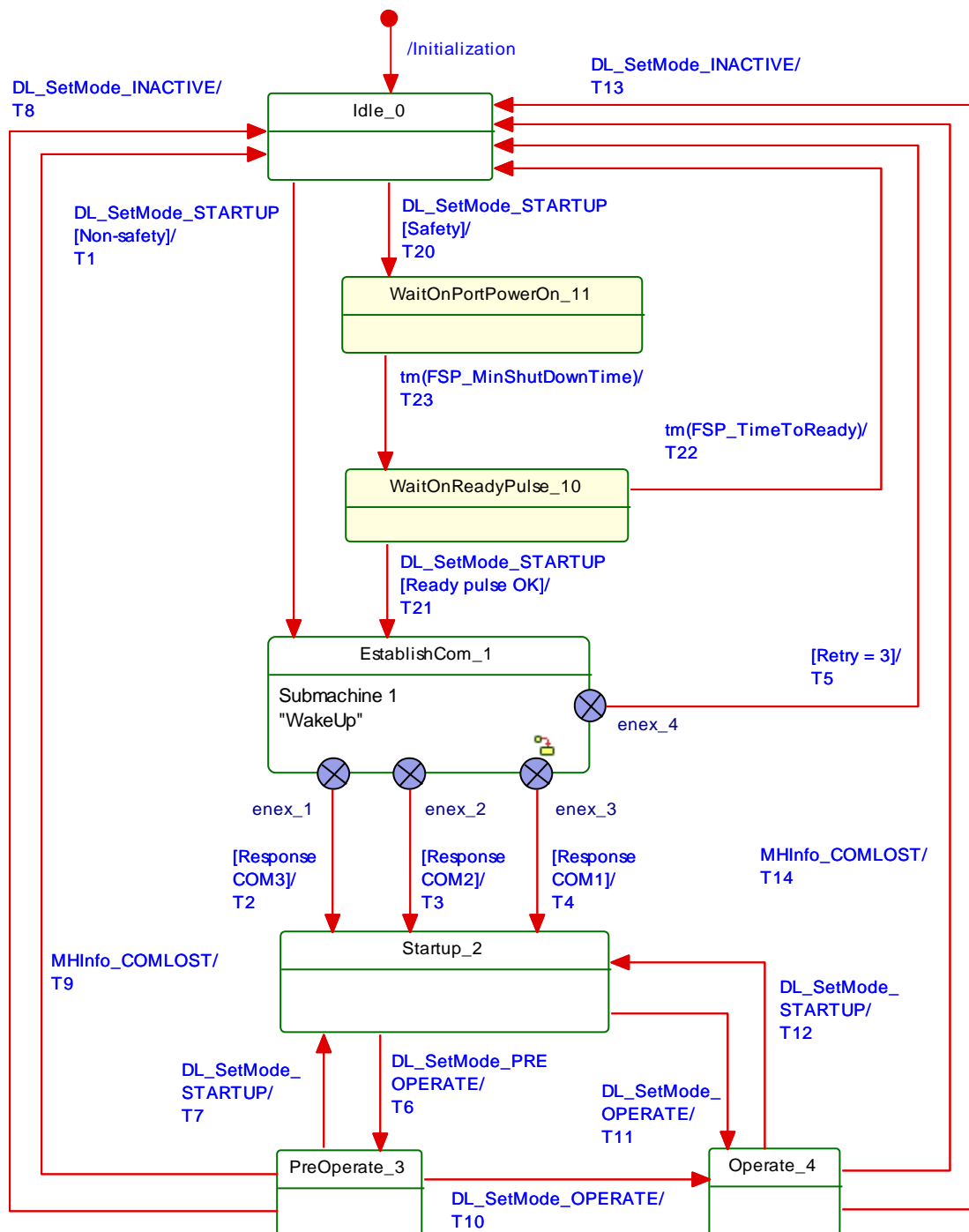


Figure 25 – State machine of the FS-Master DL-mode handler

Table 9 shows the additional state and transitions as well as internal items considering the Ready-pulse feature.

1229

Table 9 – State transition tables of the FS-Master DL-mode handler

1230

1231

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).	
WaitOnPortPowerOn_11		Wait for PortPowerOn after FSP_MinShutDownTime	
TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1 to T19	*	*	See Table 42 in IEC 61131-9:2022
T20	0	11	This path is taken only if the new configuration parameter "Safety" has been assigned to "SafetyCom" respectively and issue PortPowerOffOn (FSP_MinShutDownTime)
T21	10	1	Set Retry = 0.
T22	10	0	FS-Master was not able to detect a Ready-pulse within FSP_TimeToReady
T23	11	10	Wait for FSP_MinShutDownTime
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"
Ready pulse OK		Guard	Ready pulse detected

1232

1233

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

1234

Figure 26 shows the modifications of the FS-Device DL-mode handler versus the Device DL-mode handler in IEC 61131-9.

1235

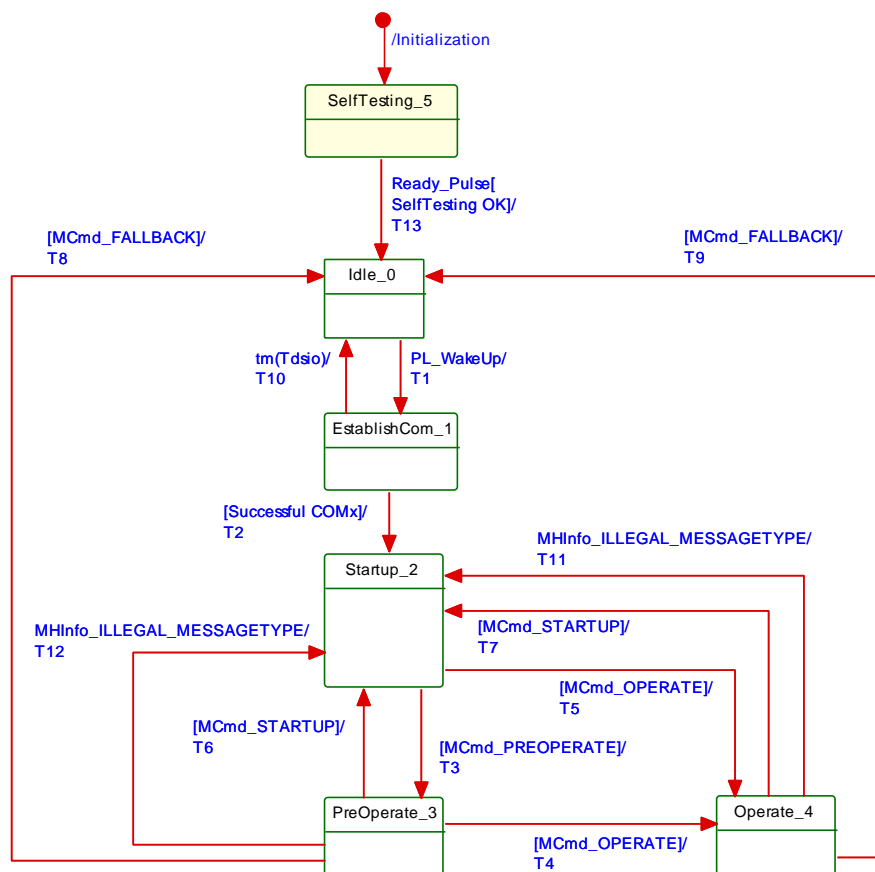


Figure 26 – State machine of the FS-Device DL-mode handler

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 10 – 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 6).
INTERNAL ITEMS		TYPE	DEFINITION
T_{RDL}		Time	See Table 10 in IEC 61131-9:2022
t_{RP}		Time	See Table 6
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 11.8.4 and shown in Figure 55. This requires an extension to the Configuration Manager as shown in Figure 27.

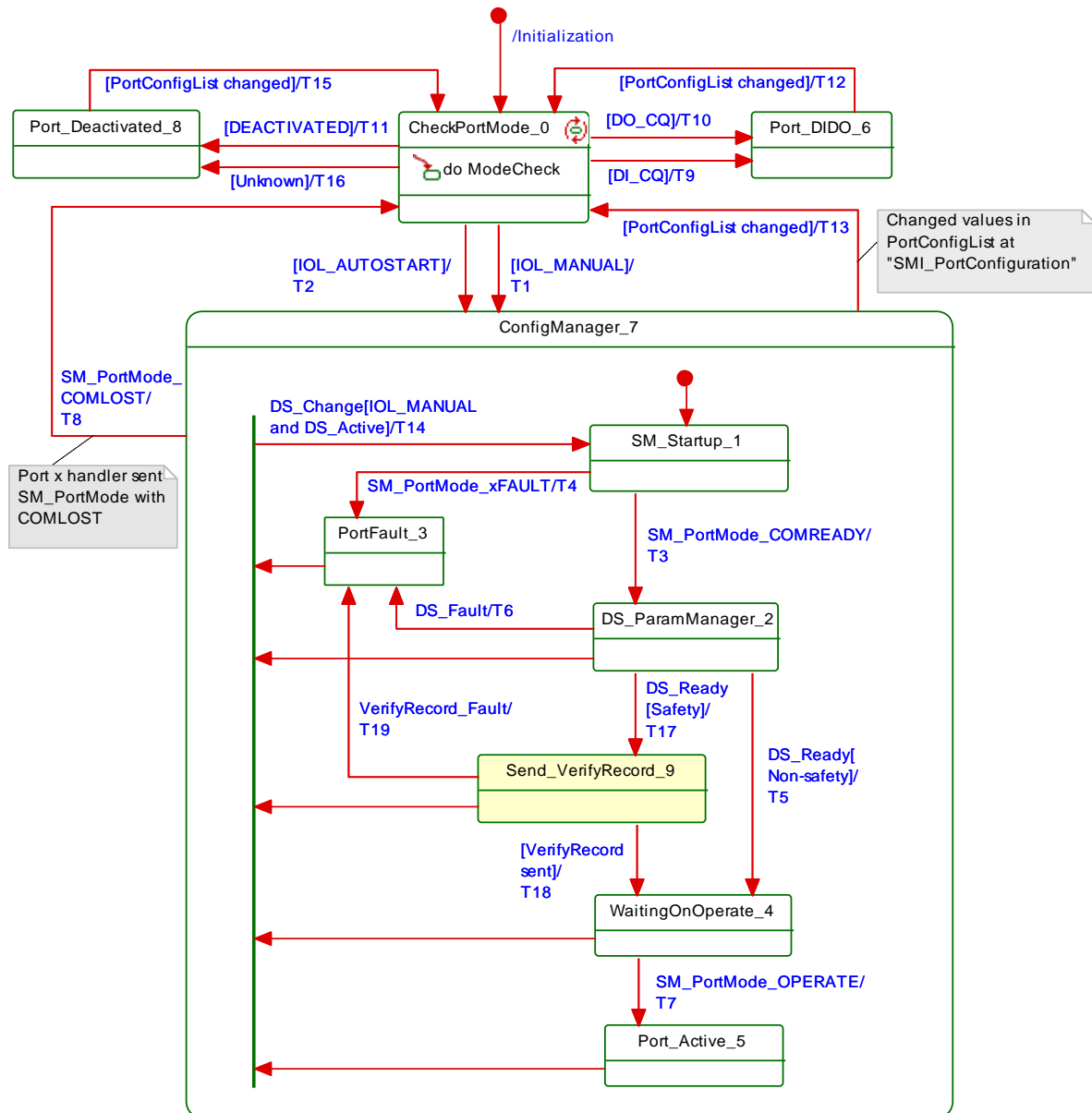


Figure 27 – Extension to the Configuration Manager (VerifyRecord)

A new state "Send_VerifyRecord_9" considers the requirement for the FS-Master to send the VerifyRecord and Table 11 the additional state, transitions, and internal items.

Table 11 – 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
T19	9	3	Update parameter elements of "PortStatusList": - PortStatusInfo = PORT_DIAG: Launch SMI_PortEvent with error 0x2007 - RevisionID = (real) RRID - Transmission rate = COMx - VendorID = (real) RVID - DeviceID = (real) RDID - Port QualityInfo = invalid

INTERNAL ITEMS	TYPE	DEFINITION
Safety	Guard	FS-Device in FS mode
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 28 shows the principle architecture of the FS-Device. It does not include safety measures for implementation such as redundancy for the safety-related parts.

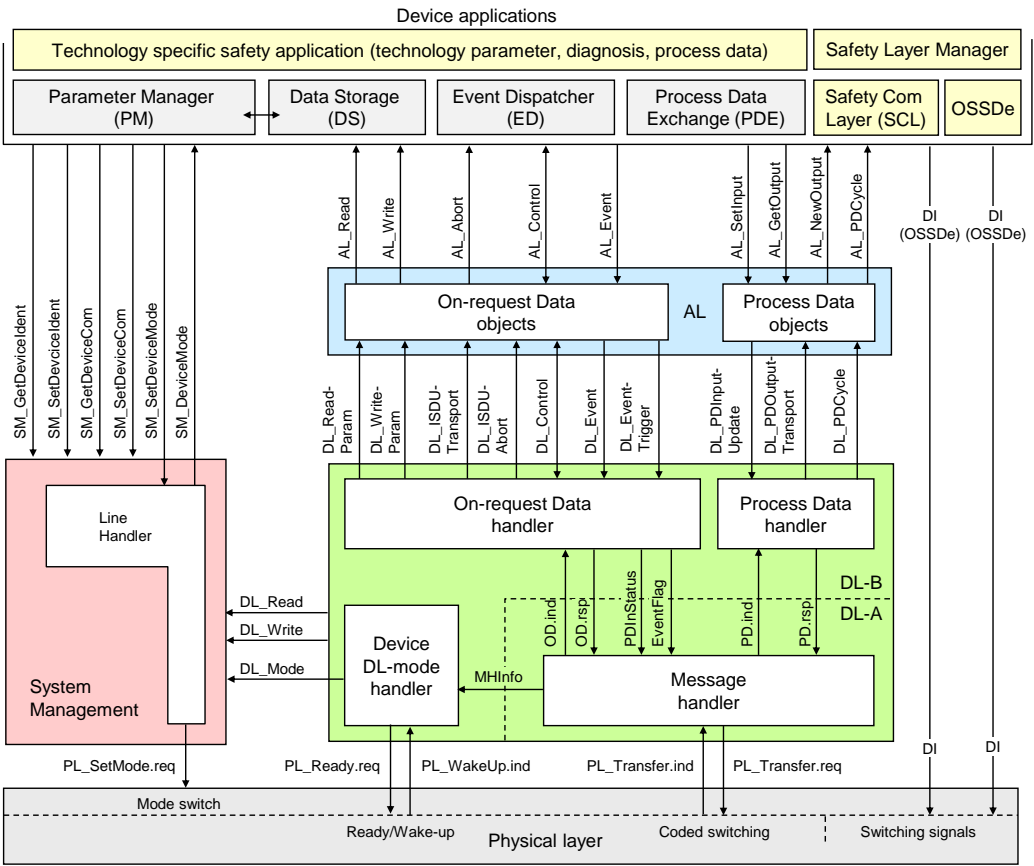


Figure 28 – Principle architecture of the FS-Device

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 29 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 SCL is running), 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 11.8.4). 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 29).

Technology specific parameters (FST) could be handled either in an open manner to a certain extend as standard non-safety parameters (see 11.8.8) or in a protected manner in hidden internal memory (see 11.8.9).

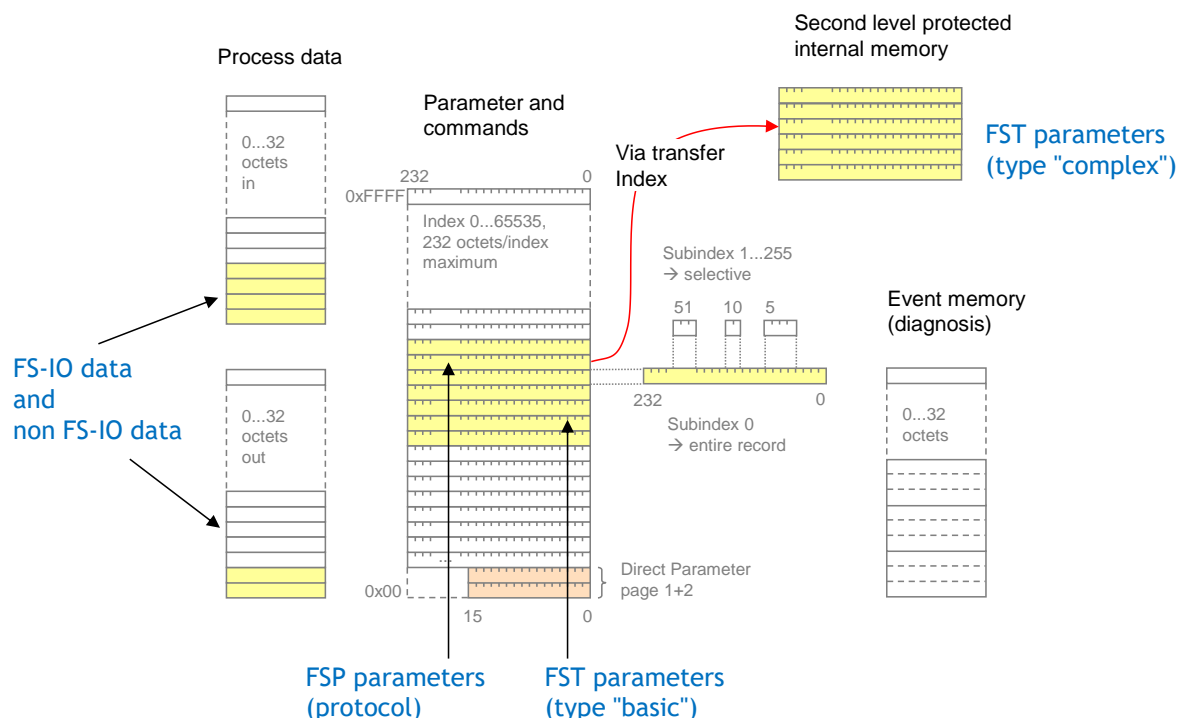


Figure 29 – The FS-Device model

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 T9 in IEC 61131-9:2022 (see [24]), Table 100 considers this check as shown in Table 12.

Table 12 – Extension to Data Storage (DS) state machine

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1 to T8	*	*	See Table 100 in IEC 61131-9:2022
T9	3	2	<p>In case of Safety: if the device is not in "Back-to-box" mode, and the Command was "DS_Download_End", and there was a change in the FSP- and FST-Parameter or the device is configured for commissioning mode:</p> <ul style="list-style-type: none"> - return 0x8036 – Function temporarily unavailable – Device control. - Unlock local parameter access. Set State_Property = "Inactive" <p>In all other cases:</p> <ul style="list-style-type: none"> - Set DS_UPLOAD_FLAG = FALSE, unlock local parameter access - Set State_Property = "Inactive"
T10 to T11	*	*	See Table 100 in IEC 61131-9:2022

9.4.2 Backup levels

Table 13 lists the Data Storage backup levels specified in IEC 61131-9. This Clause describes some specialties to be considered for functional safety.

Table 13 – Data Storage Backup Levels

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

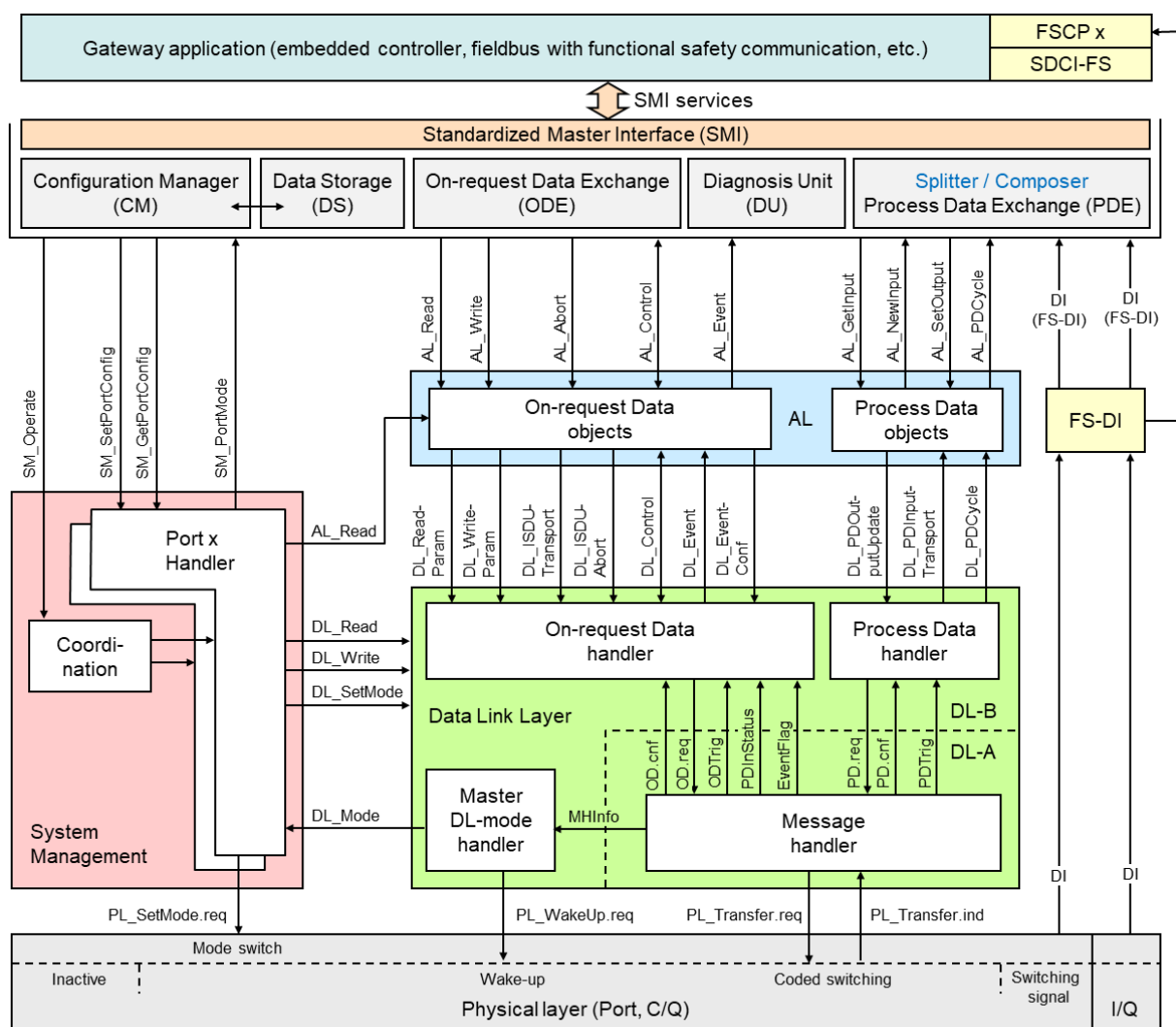
In case of functional safety, commissioning cannot be completed without verification and validation of FSP and FST parameters as well as of entire safety functions according to the relevant safety manuals.

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 30 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. Implementation of the FS-DI is vendor specific.



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

Figure 30 – 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 FS-DI 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 additional SMI services are specified as shown in Table 14 and in Figure 31: 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 14 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 14 – 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
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	See 10.2.6
SMI_PDOut	R	0x1002	See 10.2.7
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 <div style="display: inline-block; width: 20px; height: 10px; background-color: yellow; margin: 0 10px;"></div> yellow marked services are either extended or additional ones for SDCI-FS			

Figure 31 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.

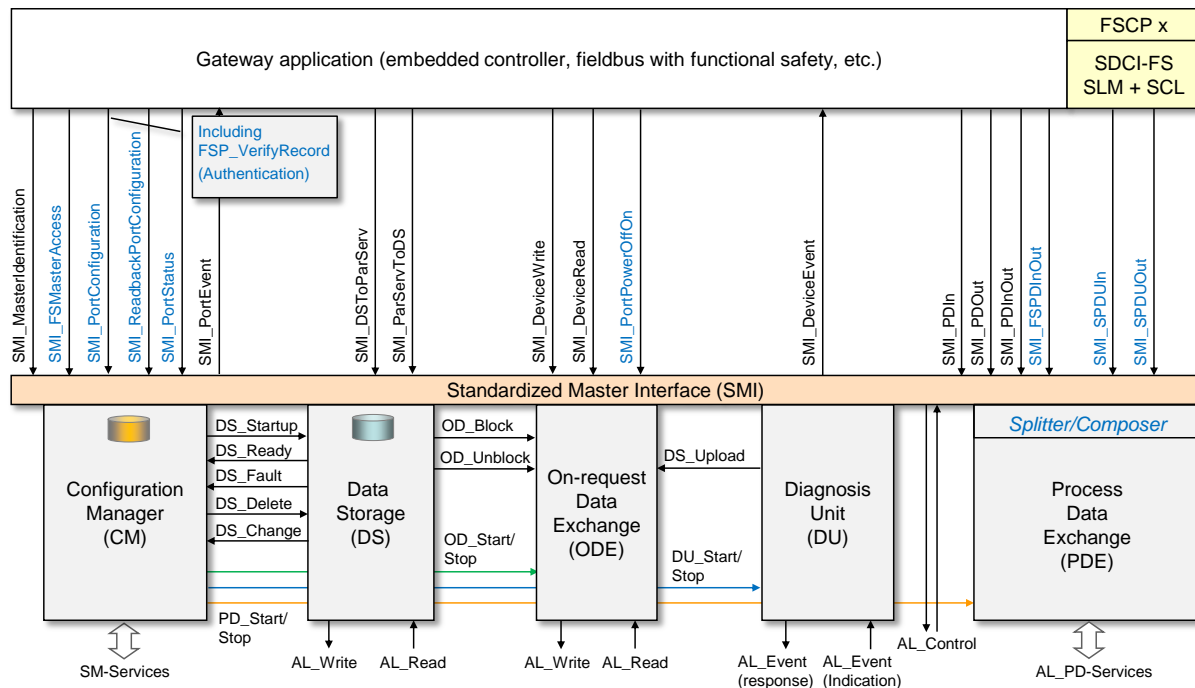


Figure 31 – SMI service extensions

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 31).

Table 15 shows the structure of the service following the rules defined in IEC 61131-9.

Table 15 – 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

Parameter name	.req	.cnf
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0x0100)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

1372

Argument1373 The specific parameters of the service request are transmitted in the argument.
13741375 **ClientID**1376 **PortNumber**

1377 Allowed are 1 to n (n = number of ports; 0 is not allowed)

1378 **ExpArgBlockID**

1379 This parameter contains the ArgBlockID of "FSCPAAuthenticity" (0x0101, see Table 19)

1380 **ArgBlockLength**

1381 This parameter contains the length of the subsequent ArgBlock

1382 **ArgBlock**

1383 This parameter contains the ArgBlock "FSMasterAccess" (0x0100, see Table 18)

1384 **Result (+):**

1385 This selection parameter indicates that the service request has been executed successfully.

1386 **ClientID**1387 **PortNumber**1388 **RefArgBlockID**

1389 This parameter contains as reference the ID of the ArgBlock sent by the request (0x0100)

1390 **ArgBlockLength**

1391 This parameter contains the length of the subsequent ArgBlock

1392 **ArgBlock**

1393 This parameter contains the ArgBlock associated to the ExpArgBlockID (see Table 19)

1394 **Result (-):**

1395 This selection parameter indicates that the service request failed

1396 **ClientID**1397 **PortNumber**1398 **RefArgBlockID**

1399 This parameter contains as reference the ID of the ArgBlock sent by the request (0x0100)

1400 **ArgBlockLength**

1401 This parameter contains the length of the "JobError" ArgBlock

1402 **ArgBlock**

1403 This parameter contains the ArgBlock "JobError" (0xFFFF, see IEC 61131-9)

1404 Permitted values in prioritized order:

1405 PORT_NUM_INVALID (incorrect Port number)

1406 ARGBLOCK_NOT_SUPPORTED (ArgBlock unknown)

1407 ARGBLOCK_LENGTH_INVALID (incorrect ArgBlock length)

1408 ARGBLOCK_INCONSISTENT (incorrect ArgBlock content type)

1409 SERVICE_TEMP_UNAVAILABLE (Master busy)

1410 **10.2.3 SMI_SPDUIn**1411 This service allows for cyclically reading Safety Protocol Data Units (SPDU) from an FSInBuffer
1412 (see 10.5) and shall only be available for internal use by the SCL. Coding of this SMI service
1413 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 16 shows the structure of the service.

Table 16 – SMI_FSPDInOut

Parameter name	.req	.cnf
Argument		
ClientID	M	
PortNumber	M	
ExpArgBlockID (0x1103)	M	
ArgBlockLength	M	
ArgBlock (VoidBlock: 0xFFFF0)	M	
Result (+)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFFF0)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M
Result (-)		S
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock 0xFFFF0)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

Argument

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

ClientID

PortNumber

This parameter contains the port number. Allowed values are 1 to n, 0 is not allowed.

ExpArgBlockID

This parameter contains the ArgBlockID of "FSPDInOut" (0x1103, see Table 24)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock

ArgBlock

This parameter contains the ArgBlock "VoidBlock" (0xFFFF0)

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 (0xFFFF0)

ArgBlockLength

This parameter contains the length of the subsequent ArgBlock

ArgBlock

This parameter contains the ArgBlock associated to the ExpArgBlockID (see Table 24)

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 (0xFFFF0)

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)

10.2.6 SMI_PDIn

This service allows for cyclically reading Process Data from an InBuffer (see 10.5). In IO-Link PortMode = "SAFETYCOM" this service contains only the NSR data of the Process Data In. In IO-Link PortMode = "IOL_MANUAL" or PortMode = "IOL_AUTOSTART" this service contains the complete PDU.

The ArgBlockID is "0x1001".

10.2.7 SMI_PDOut

This service allows cyclically writing Process Data to an OutBuffer (see 10.5). In IO-Link PortMode = "SAFETYCOM" this service contains only the NSR data of the Process Data Out. In IO-Link PortMode = "IOL_MANUAL" or PortMode = "IOL_AUTOSTART" this service contains the complete PDU.

The ArgBlockID is "0x1002".


10.3 ArgBlock extensions

10.3.1 Overview

Table 17 shows new ArgBlock types for FS-Masters: "FSMasterAccess", "FSCPAAuthenticity", "FSPortConfigList", "FSPortStatusList", "SPDUIn", "SPDUOut", and "FSPDInOut".

Table 17 – ArgBlock types and ArgBlockIDs

ArgBlock type	ArgBlockID	Remark
FSMasterAccess	0x0100	See 10.3.2
FSCPAAuthenticity	0x0101	See 10.3.3
PDIn	0x1001	–
PDOut	0x1002	–
PDInOut	0x1003	–
SPDUIn	0x1101	See 10.3.6

ArgBlock type	ArgBlockID	Remark
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 marked ArgBlocks are additional ones for SDCI-FS		

1479

1480 **10.3.2 FSMasterAccess**

1481 The ArgBlock "FSMasterAccess" in Table 18 shows the password for FS-Master access and
 1482 the corresponding password to reset the entire FS-Master project including the existing
 1483 password.

1484

Table 18 – FSMasterAccess

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x0100
2	FSMasterPassword	Reserved. Default: 0x00000000 FSMasterPassword shall be left on the default value.	Unsigned32	–
6	FSResetMasterPW	This element contains the password for resetting the entire FS-Master project including the FSMasterPassword. Default is 0x00000000.	Unsigned32	–
10	FSUserRole	Reserved. Default: 0x00	Unsigned8	–

1485

1486 **10.3.3 FSCPAuthenticity**

1487 The ArgBlock "FSCPAuthenticity" in Table 19 shows FSCP authenticity codes assigned to the
 1488 FS-Master port as used in the PortConfiguration through the upper-level FSCP engineering tool
 1489 or via DIP switches.

1490

Table 19 – 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	–

1491

1492 **10.3.4 FSPortConfigList**

1493 Table 20 shows the ArgBlockType "FSPortConfigList" suitable for FS-Masters. It considers
 1494 additional PortModes and expands by Safety PDU lengths, the FSP_VerifyRecord (see 10.3.3
 1495 and A.2.10) as well as the FS I/O data structure description (see 0 and Table A.4).

1496

Table 20 – FSPortConfigList

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x8100

Offset	Element name	Definition	Data type	Values
2	PortMode	<p>This element contains the Port mode expected by the SMI client, e.g. gateway application. All modes are mandatory, except FS-DI. They shall be mapped to the Target Modes of "SM_SetPortConfig" (see IEC 61131-9:2022).</p> <p>0: DEACTIVATED (SM: INACTIVE → Port is deactivated; input and output Process Data are "0"; Master shall not perform activities at this Port)</p> <p>1: IOL_MANUAL (SM: CFGCOM → Target Mode based on user defined configuration including validation of RID, VID, DID)</p> <p>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)</p> <p>3: DI_C/Q (Pin 4 at M12) ^b (SM: DI → Port in input mode SIO)</p> <p>4: DO_C/Q (Pin 4 at M12) ^b (SM: DO → Port in output mode SIO)</p> <p>5 to 48: Reserved for future versions</p> <p>49: SAFETYCOM (implying IOL_MANUAL behavior)</p> <p>50: Reserved for future versions</p> <p>51: FS_DI (Pin 2 + Pin 4 at M12) (Not for FS-Masters of type Port Class B; Values in offset 16 to 37 are don't care; SPDULength in offset 40 = 1 octet; value in offset 41 is don't care)</p> <p>52 to 96: Reserved for extensions such as Safety or Wireless)</p> <p>97 to 255: Manufacturer specific</p>	Unsigned8	0x00 to 0xFF
3	Validation&Backup	<p>This element contains the InspectionLevel to be performed by the Device and the Backup/Restore behavior.</p> <p>0: No Device check</p> <p>1: Type compatible Device V1.0</p> <p>2: Type compatible Device V1.1</p> <p>3: Type compatible Device V1.1, Backup + Restore</p> <p>4: Type compatible Device V1.1, Restore</p> <p>5 to 255: Reserved</p>	Unsigned8	0x00 to 0xFF
4	I/Q behavior (Manufacturer or profile specific)	<p>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.</p> <p>0: Not supported</p> <p>1: Digital Input</p> <p>2: Digital Output</p> <p>3 to 255: Reserved</p>	Unsigned8	0x00 to 0xFF
5	PortCycleTime	<p>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"</p> <p>0: AFAP (As fast as possible – SM: FreeRunning → Port cycle timing is not restricted. Default value in Port mode IOL_MANUAL)</p> <p>1 to 255: TIME (SM: For coding see Table B.3 in IEC 61131-9. Device shall achieve the</p>	Unsigned8	0x00 to 0xFF

Offset	Element name	Definition	Data type	Values
		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	0x0001 to 0xFFFF
8	DeviceID	This element contains the 3 octets long DeviceID expected by the SMI client	Unsigned32	0x000001 to 0xFFFFFFFF
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	0x0001 to 0x7FFF (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	0x0064 to 0x03E8 (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	0x01 to 0xFF
25	FSP_AuthentCRC	CRC signature across complete authentication	Unsigned16	–
27	FSP_ProtVersion	Version of the SDCI-FS protocol	Unsigned8	0x01 to 0xFF
28	FSP_ProtMode	SDCI-FS protocol mode	Unsigned8	–
29	FSP_WatchdogTime	Watchdog time of FS-Master and FS-Device	Unsigned16	0x0001 to 0xFFFF
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	FS-DI data (1 octet) or length of incoming SPDU (<i>m</i>); see 10.5 and Table A.4	Unsigned8	1 or 4 to (32 – <i>n</i>) octets
41	TotalOfInBits	Set of input BooleanT (bits)	Unsigned8	0x00 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 (<i>o</i>); see 10.5 and Table A.4	Unsigned8	4 to (32 – <i>p</i>) octets
46	TotalOfOutBits	Set of output BooleanT (bits)	Unsigned8	0x00 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 "I/Q behavior".				

1497

1498 **10.3.5 FSPortStatusList**

1499 Table 21 shows the ArgBlockType "FSPortStatusList" suitable for FS-Masters. Content of
 1500 "FSPortStatusInfo" shall be derived from "PortMode" in IEC 61131-9).

1501

Table 21 – FSPortStatusList

Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x9100

Offset	Element name	Definition	Data type	Values
2	PortStatusInfo	<p>This element contains status information of the Port.</p> <p>0: NO_DEVICE No communication (COMLOST). However, Port configuration IOL_MANUAL or IOL_AUTOSTART was set.</p> <p>1: DEACTIVATED Port configuration DEACTIVATED is set.</p> <p>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.</p> <p>3: Reserved</p> <p>4: OPERATE This value to be set if the Device is in OPERATE, even in case of Device error.</p> <p>5: DI_C/Q Port configuration "DI" is set.</p> <p>6: DO_C/Q Port configuration "DO" is set.</p> <p>7: FS_DI (FS-DI at C/Q and I/Q)</p> <p>8: SCL_ENABLED (Port ready for safety data exchange)</p> <p>9 to 253: Reserved</p> <p>254: PORT_POWER_OFF Shutdown of Port is active caused by SMI_PortPowerOffON</p> <p>255: NOT_AVAILABLE (Port status currently not available)</p>	Unsigned8 (enum)	0x00 to 0xFF
3	PortQualityInfo	<p>This element contains status information on Process Data</p> <p>Bit0: 0 = VALID 1 = INVALID</p> <p>Bit1: 0 = PDOUTVALID 1 = PDOUTINVALID</p> <p>Bit2 to Bit7: Reserved</p>	Unsigned8	–
4	RevisionID	<p>This element contains information of the SDCI protocol revision of the Device</p> <p>0: NOT_DETECTED (No communication at that Port)</p> <p><>0: Copied from Direct parameters page, address 4</p>	Unsigned8	0x00 to 0xFF
5	TransmissionRate	<p>This element contains information on the effective Port transmission rate.</p> <p>0: NOT_DETECTED (No communication at that Port)</p> <p>1: COM1 (transmission rate 4,8 kbit/s)</p> <p>2: COM2 (transmission rate 38,4 kbit/s)</p> <p>3: COM3 (transmission rate 230,4 kbit/s)</p> <p>4 to 255: Reserved for future use</p>	Unsigned8	0x00 to 0xFF
6	MasterCycleTime	<p>This element contains information on the Master cycle time. For coding see IEC 61131-9 B.1.3.</p>	Unsigned8	–

Offset	Element name	Definition	Data type	Values
7	InputDataLength	This element contains the input data length as number of octets of the Device provided by the PDIn service	Unsigned8	0x00 to 0x20
8	OutputDataLength	This element contains the output data length as number of octets for the Device accepted by the PDOOut service	Unsigned8	0x00 to 0x20
9	VendorID	This element contains the 2 octets long VendorID expected by the SMI client	Unsigned16	0x0000 to 0xFFFF
11	DeviceID	This element contains the 3 octets long DeviceID expected by the SMI client	Unsigned32	0x000001 to 0xFFFFFFFF
15	NumberOfDiags	This element contains the number <i>x</i> of diagnosis entries (DiagEntry0 to DiagEntryx)	Unsigned8	0x00 to 0xFF
16	DiagEntry0	This element contains the "EventQualifier" and "EventCode" of a diagnosis (Event).	Struct Unsigned8/16	–
19	DiagEntry1	Further entries up to <i>x</i> if applicable...	...	–

10.3.6 SPDUIIn

Table 22 shows the structure of the ArgBlock "SPDUIIn" as illustrated in 10.5.

Table 22 – SPDUIIn

Offset	Element name	Definition	Data type
0	ArgBlockID	0x1101	Unsigned16
2	SPDUIIn0	Safety Protocol Data Unit in (octet 0)	Unsigned8
3	SPDUIIn1	Safety Protocol Data Unit in (octet 1)	Unsigned8
...			
SPDUIInLength + 2	SPDUIIn m	Safety Protocol Data Unit in (octet m)	Unsigned8

10.3.7 SPDUIOut

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

Table 23 – SPDUIOut

Offset	Element name	Definition	Data type
0	ArgBlockID	0x1102	Unsigned16
2	SPDUIOut0	Safety Protocol Data Unit out (octet 0)	Unsigned8
3	SPDUIOut1	Safety Protocol Data Unit out (octet 1)	Unsigned8
...			
SPDUIOutLength + 2	SPDUIOut o	Safety Protocol Data Unit out (octet o)	Unsigned8

10.3.8 FSPDInOut

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

Table 24 – 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	SPDUInLength	This element contains the length of the Device's SPDUIn.	Unsigned8	0x00 to 0x1F
5	PDILength	This element contains the length of the Device's non-safe input Process Data (PDI0 to PDI n).	Unsigned8	0x00 to 0x20
6	SPDUIn0	Safety Protocol Data Unit in (octet 0)	Unsigned8	0x00 to 0xFF
7	SPDUIn1	Safety Protocol Data Unit in (octet 1)	Unsigned8	0x00 to 0xFF
...				
$m+6$	SPDUIn m	Safety Protocol Data Unit in (octet m)	Unsigned8	0x00 to 0xFF
$(m+6) + 1$	PDI0	Input Process Data (octet 0)	Unsigned8	0x00 to 0xFF
$(m+6) + 2$	PDI1	Input Process Data (octet 1)	Unsigned8	0x00 to 0xFF
...				
$(m+6) + (n+1)$	PDI n	Input Process Data (octet n)	Unsigned8	0x00 to 0xFF
$(m+6) + (n+1) + 1$	SPDUOutLength	This element contains the length of the Device's SPDUOut.	Unsigned8	0x00 to 0x1F
$(m+6) + (n+1) + 2$	PDOLength	This element contains the length of the Device's non-safe output Process Data (PDO0 to PDO p).	Unsigned8	0x00 to 0x20
$(m+6) + (n+1) + 3$	SPDUOut0	Safety Protocol Data Unit out (octet 0)	Unsigned8	0 to 0xFF
$(m+6) + (n+1) + 4$	SPDUOut1	Safety Protocol Data Unit out (octet 1)	Unsigned8	0x00 to 0xFF
...				
$(m+6) + (n+1) + (o+3)$	SPDUOut o	Safety Protocol Data Unit out (octet o)	Unsigned8	0x00 to 0xFF
$(m+6) + (n+1) + (o+3) + 1$	PDO0	Output Process Data (octet 0)	Unsigned8	0x00 to 0xFF
$(m+6) + (n+1) + (o+3) + 2$	PDO1	Output Process Data (octet 1)	Unsigned8	0x00 to 0xFF
...				
$(m+6) + (n+1) + (o+3) + (p+1)$	PDO p	Output Process Data (octet p)	Unsigned8	0x00 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 FS-DI 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 11.8.4) 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 30 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)

The FS-Master shall support the additional port mode SAFETYCOM and can optionally support port mode FS_DI.

SAFETYCOM

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

FS_DI

This setting enables FS-DI 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 32 illustrates some use cases related to the FSP parameters (see A.1).

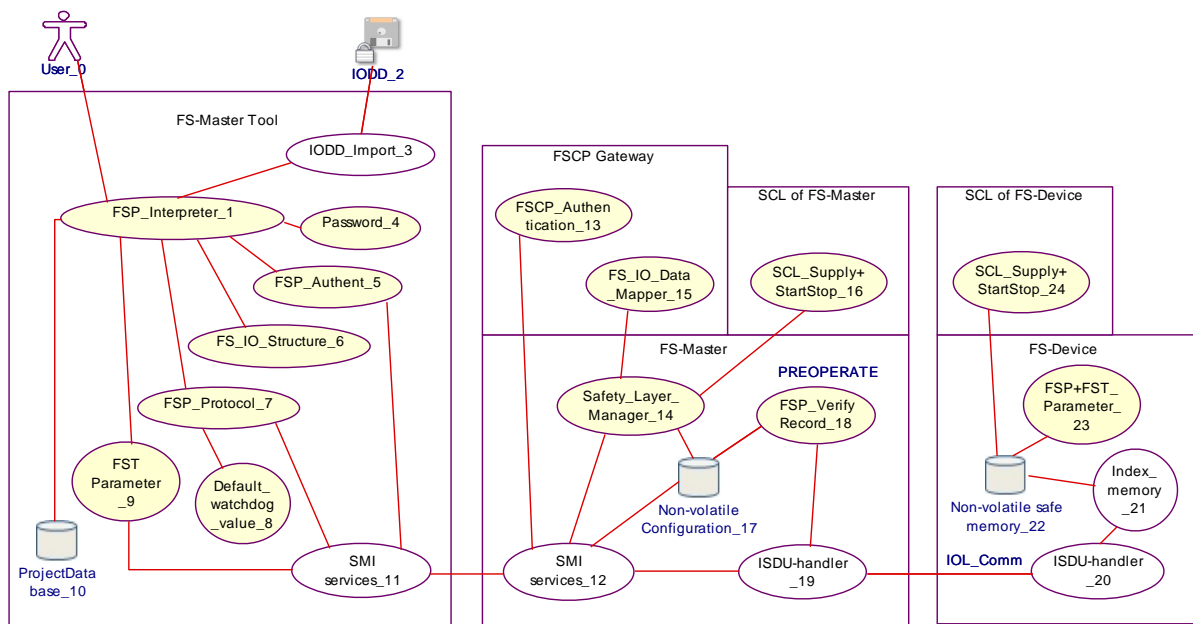


Figure 32 – FSP parameter use cases

Table 25 shows a listing of the items in Figure 32 and references clauses within this document or IEC 61131-9.

Table 25 – 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 56	–
2	IODD (secured)	Device description	Annex E	–
3	IODD_Import	Activity	Annex E	–
4	Password	Activity	10.4.3.2	Access protection
5	FSP_Authent	Activity	11.8.5	–
6	FS_IO_Structure	FS I/O description	Clause A.1	–
7	FSP_Protocol	Activity	11.8.6	–
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	11.8.5	–
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	11.8.4	–
19	ISDU-Handler	FS-Master DL	IEC 61131-9	Standard SDCI
20	ISDU-Handler	FS-Device DL	IEC 61131-9	Standard SDCI

No.	Item	Type	Reference	Remarks
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	–

1550

1551 In the following, a typical parameterization session of a project in the ProjectDatabase is
 1552 described, where a new FS-Device is planned, configured, and parameterized for a particular
 1553 Port. After installation of IODD and associated Dedicated Tool, the user of an FS-Master Tool
 1554 opens the parameter tab page (see illustration in Figure 56). After entry of the password for
 1555 safety projects (see 10.4.3.2), FSP parameters are enabled to be displayed and Dedicated
 1556 Tools are enabled to be launched.

1557 When online, the FS-Master Tool uses the Standardized Master Interface (SMI) to the FS-
 1558 Master specified in IEC 61131-9. Any transmission error (see Table 26) can falsify the message
 1559 bits and thus, each FSP parameter record is secured by CRC signature.

1560 The choice of the SMI service call technology is the responsibility of the respective integration
 1561 into a fieldbus (see IEC 61131-9).

1562 The *authenticity parameter* values carry "0" as default within the IODD of an FS-Device. FS-
 1563 Master Tool acquires FSCP Authenticity values with the help of the SMI_FSMasterAccess
 1564 service (see 10.2.2) and suggests these as actual values. For details see 10.4.3.3.

1565 The IODD contains the *I/O data structure description* of the safety Process Data as a record
 1566 secured by CRC signature (see A.2.7 and E.5.6). This information is used for the mapping to
 1567 FSCP I/O data and checked by FS-Device (see 0).

1568 Most of the *protocol parameter* values are preset by default values provided by the FS-Device
 1569 manufacturer within the IODD, except for the value of FSP_TechParCRC, which has a particular
 1570 responsibility. A value of "0" followed by a Port power OFF/ON means commissioning/test (see
 1571 Annex G). The consequences are:

- 1572 • changes of FST parameters become effective right upon acceptance by the FS-Device;
- 1573 • no Data Storage backup.

1574 From now on the SDCI-FS system can run in "monitored operational mode". That means
 1575 personnel are required to watch the machine.

1576 The user can now enter and test the technology specific parameters (see illustration in Figure
 1577 56). After verification and validation, the user launches the Dedicated Tool, confirms the value
 1578 assignments, and transfers the CRC signature to the FSP_TechParCRC field. The
 1579 corresponding SMI_PortConfiguration cares for the FSP_VerifyRecord within the FS-Master.
 1580 With an FSP_TechParCRC value of \neq "0", the system can be armed:

- 1581 • Data Storage;
- 1582 • Verification of authenticity, protocol, and technology parameters, as well as IO data
 1583 description at start-up.

1584 After parameter assignment, the FSP and FST parameter instance values can be stored in the
 1585 ProjectDatabase.

1586 Another Port power OFF/ON will cause the FS-Device to perform safety selftests prior to
 1587 communication and the FS-Master to transmit the FSP_VerifyRecord to the FS-Device. Its
 1588 Safety Layer Manager verifies all parameters, passes relevant protocol parameters, and
 1589 launches the SCL. In case of mismatch a corresponding Event is activated and the SCL will not
 1590 operate.

1591 The SLM propagates the I/O structure description to the FS_IO_DataMapper. The
 1592 FSP_VerifyRecord is propagated to the local FS-Master safety communication layer (SCL). It
 1593 verifies all parameters, passes relevant protocol parameters, launches the SCL, and sets

1594 PortStatusInfo to "8" (SCL_ENABLED). In case of mismatch a corresponding Event is activated
1595 and the SCL will not operate.

1596 **10.4.3.2 Protection**

1597 An FS-Master Tool creates and maintains safety projects and the FS-Master stores safety
1598 parameters within configuration data. Both require protection from easy manipulation. FS-
1599 Master Tool can use password parameters within the SMI_FSMasterAccess service (10.2.2) for
1600 that purpose. Usage depends on the security concept of the upper-level system and on the
1601 inheritance concept of password protection. It is optional.

1602 Dedicated Tools may have password mechanisms for their FS-Devices independent from the
1603 FS-Master (see 10.2.2).

1604 **10.4.3.3 FSP authenticity parameter record**

1605 FSP authenticity parameters are specified in Clause A.1. The FSP authenticity record includes
1606 the FSCP authenticity code, a Port number, and a CRC signature. An FS-Master Tool shall
1607 always update the CRC signature when changes occur and only write entire consistent records.

1608 For stand-alone FS-Masters the entry of unique and unambiguous values per FS-Master is
1609 required per machine or production center if there is a possibility to misconnect FS-Devices
1610 amongst different FS-Masters.

1611 **10.4.3.4 FSP protocol parameter record**

1612 FSP protocol parameters are specified in Clause A.1. Manufacturer/vendor presets values and
1613 defines ranges within the IODD for protocol version and mode, Port mode, watchdog, and
1614 TechParCRC.

1615 Manufacturer/vendor shall determine the preset value for the watchdog timer considering the
1616 FS-Device response time at the indicated transmission rate (COMx). The FS-Master Tool can
1617 calculate and suggest a value based on the performance data of the used FS-Master and on
1618 the preset value from the IODD.

1619 An FS-Master Tool shall always update the CRC signature when changes occur and only write
1620 entire consistent records.

1621 **10.4.3.5 FS I/O data structure description**

1622 With the help of this information, the mapping process within the FSCP gateway can be
1623 controlled or monitored (see 0 and A.2.7).

1624 The FS-Device shall check the validity of its implemented safety PDin/PDout structure via the
1625 FSP_IO_StructCRC signature provided by the FSP_VerifyRecord.

1626 **10.4.3.6 Verification record**

1627 The FS-Master takes the FSP_VerifyRecord from the SMI_PortConfiguration service (see
1628 10.3.3 and 11.8.4).

1629 Default value of the FSP_VerifyRecord is all octets "0". "0" means that the FSP_VerifyRecord
1630 is not valid. If the element FSP_TechParCRC is "0" the FSP_VerifyRecord shall not be stored.
1631 After a PowerOffOn on the FS-Master, the FSP_VerifyRecord shall have the last valid value
1632 without FSP_TechParCRC "0" or the default value with all octets "0" if there was no valid
1633 FSP_VerifyRecord before. This applies not only for the FSP_VerifyRecord but for the complete
1634 PortConfiguration.

1635 **10.5 Process Data Exchange (PDE)**

1636 The FS-Master application Process Data Exchange (PDE) provides additional features called
1637 "Splitter" and "Composer".

1638 Figure 33 shows the mechanism of splitting the SPDUI part and the Input Data from the
1639 complete SR and NSR data. The SR part is stored within an FSInBuffer and the NSR part within
1640 the InBuffer.

- 1641 In case of PortConfiguration "FS_DI", the signal status of C/Q is stored as "FS_DI1" in Bit "0"
 1642 of octet "0", and signal status of I/Q is stored as "FS_DI2" in Bit "1" of octet "0".
- 1643 See IEC 61131-9 for definitions of "PQI" and "PQ". PDIn valid/invalid only refer to non-safety
 1644 data.

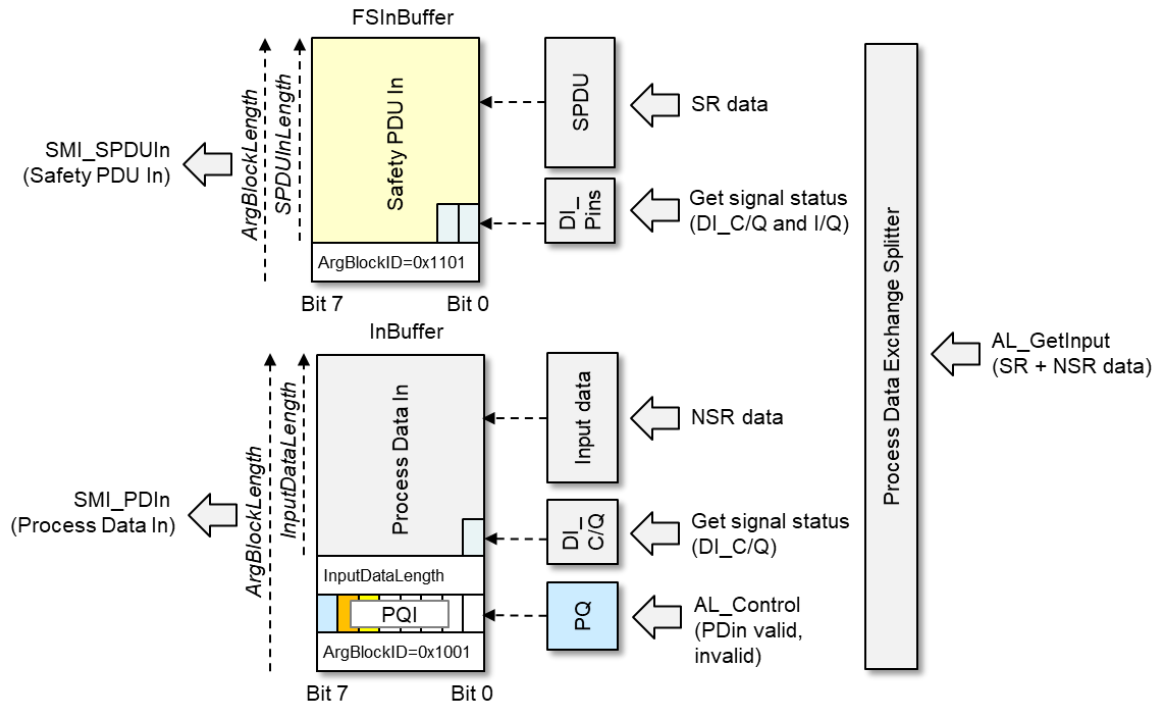


Figure 33 – PDE Splitter (only in IO-Link PortMode = "SAFETYCOM")

- 1647 Figure 34 shows the mechanism of composing the complete SR and NSR data. Both are
 1648 prepared for the AL_SetOutput service out of the SPDU Out part from the FSOutBuffer and out
 1649 of the Process Data Out from the OutBuffer.
- 1650 See IEC 61131-9 for definitions of "OE" and "OE detect".

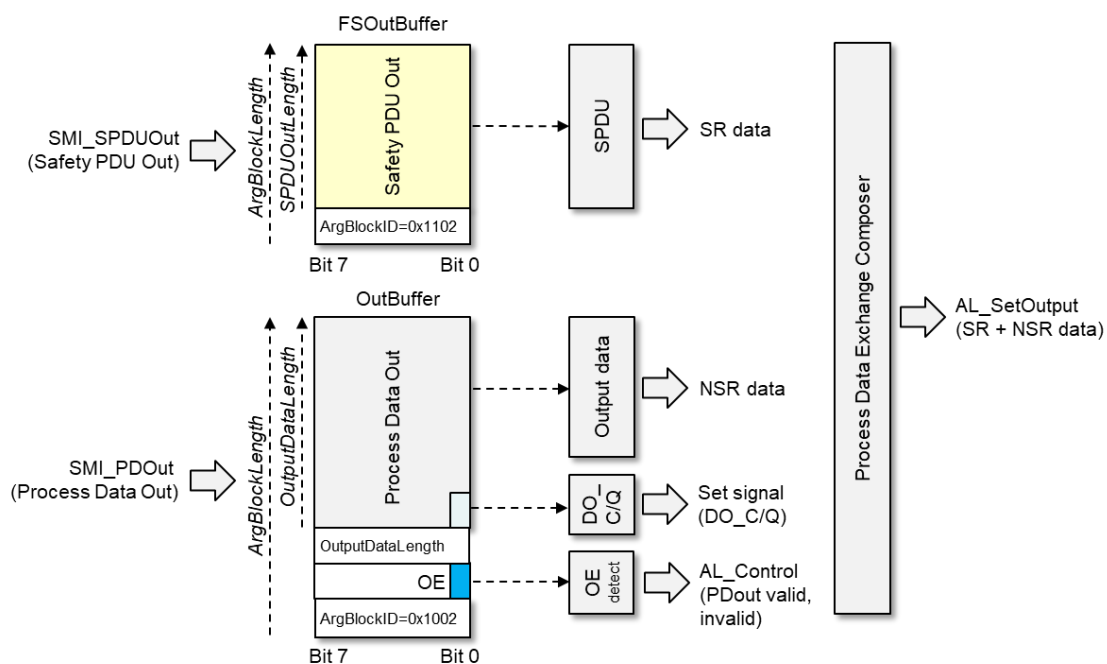


Figure 34 – PDE Composer (only in IO-Link PortMode = "SAFETYCOM")

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 [8]. 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 26 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 26 – 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	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.

11.3 SCL services

11.3.1 Positioning of safety communication layers (SCL)

Figure 35 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).

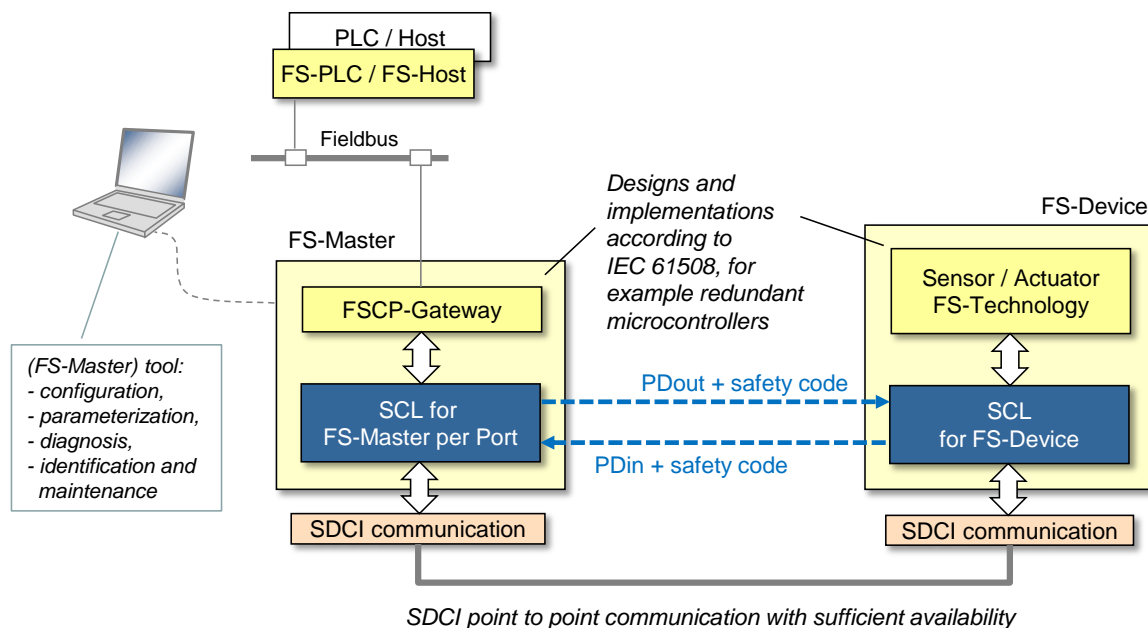


Figure 35 – Positioning of the SDCI-FS Safety Communication Layer (SCL)

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 36 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.

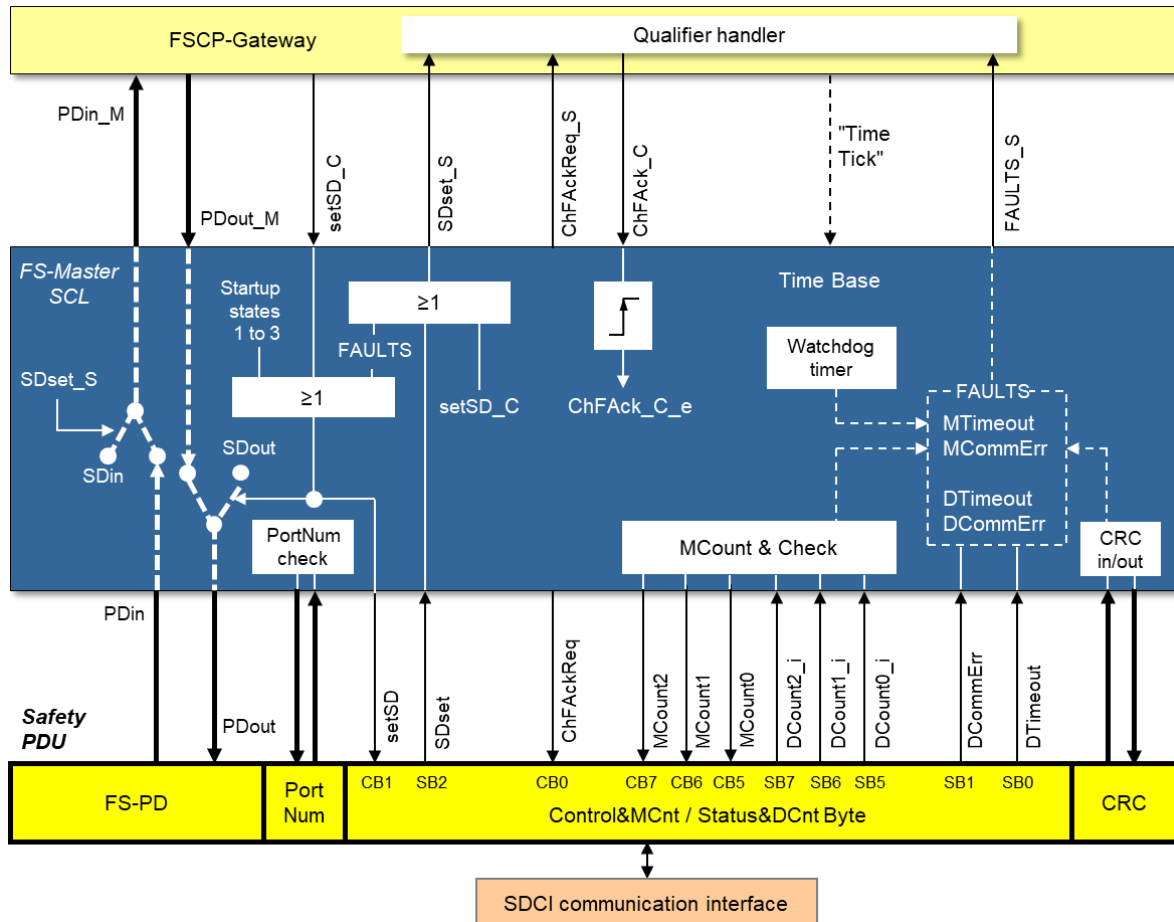


Figure 36 – FS-Master Safety Communication Layer services

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 27 shall be available to the FSCP gateway or to a programmer of an FS-Master system.

Table 27 – 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 11.12.4). In addition, it causes the SCL to send SDin to the FSCP gateway (PDin_M).

Service/signal	Definition
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 37.
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 11.12.4).

1724

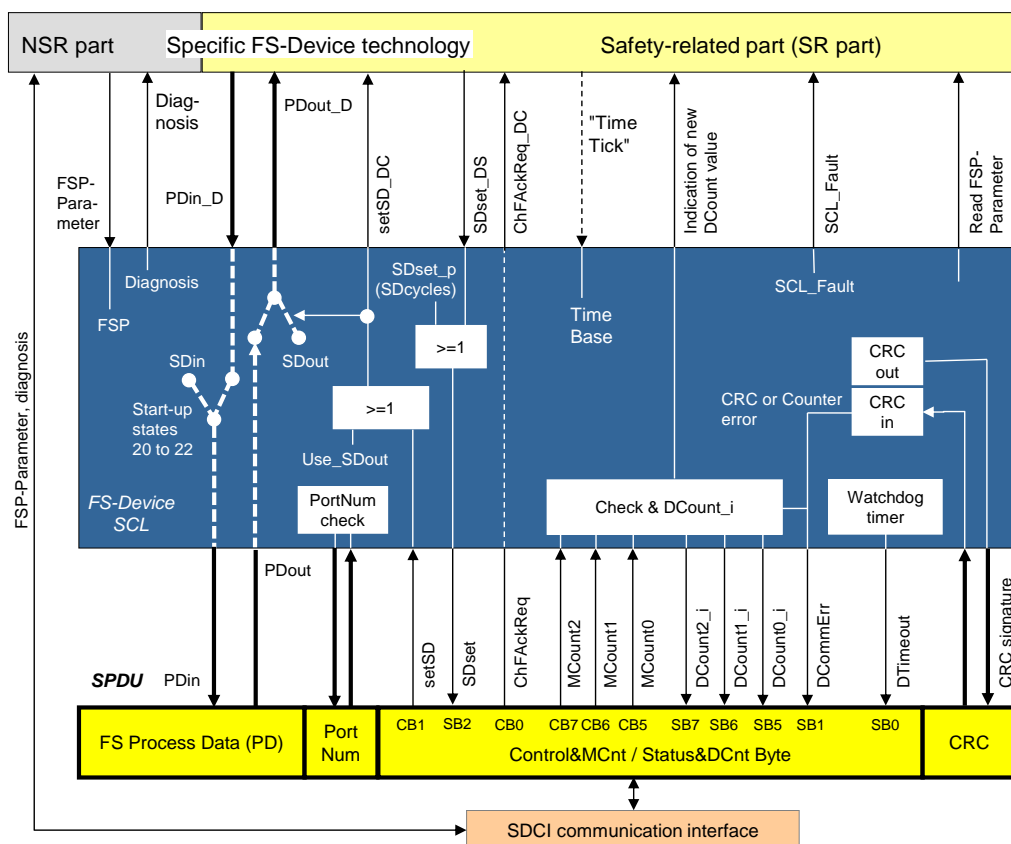
1725 The lower part of the figure shows a combined input and output safety PDU specified in 11.4.3
 1726 and 11.4.5.

1727 11.3.3 FS-Device SCL services

1728 Figure 37 shows the FS-Device Safety Communication Layer services depicted by arrows in
 1729 the upper part of the figure.

1730 A service name carries either an extension "_DC" (Device Control) if it controls the FS-Device
 1731 technology or an extension "_DS" (Device Status) if it is reporting its status.

1732 Some of the service names correspond to the signal names of the Control Byte or Status Byte
 1733 (see lower part of the figure and 11.4.5). That means they are correlated, but there is some
 1734 control logic of the SCL in between. This control logic is time discrete and not continuous even
 1735 if it is depicted as logic OR (" \geq ") box. Definitive are the state charts and the state transition
 1736 tables of the SCL (see 11.5.3).



1737

1738

Figure 37 – FS-Device Safety Communication Layer services

The following services in Table 28 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 28 – 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.
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 37 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 38 shows the principle protocol phases to consider for the design according IEC 61784-3:2021.

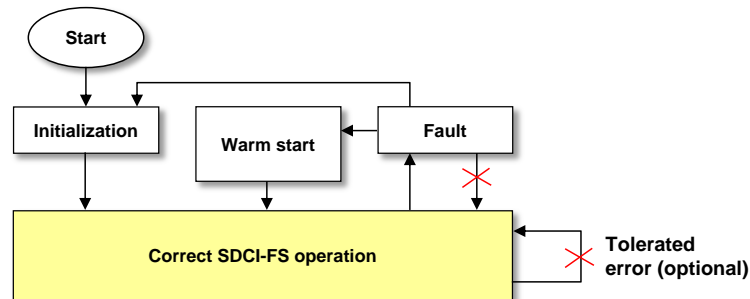


Figure 38 – Protocol phases to consider

The principle protocol phases and the corresponding requirements are listed in Table 29.

Table 29 – 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 and SCL 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 - 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 39 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.

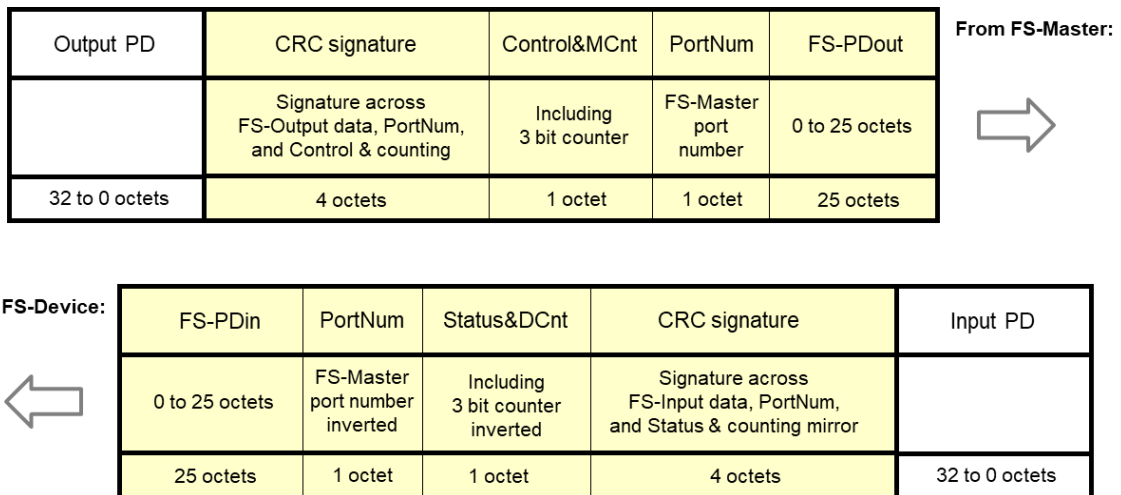


Figure 39 – Safety PDUs of FS-Master and FS-Device

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.

PD validity shall be applied only on the non-safe part of the process data (Output PD, Input PD).

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 33.

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. The port number is the physical port number of the master and is identically in the SDCI black channel.

11.4.6 Status and control

One octet is used in both transmission directions for the protocol flow of SDCI-FS. Table 30 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 30 – 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 31 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 31 – 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	Communication fault: Timeout
DCount2_i	DCount1_i	DCount0_i	–	–	SDset	DCommErr	DTimeout

Table 32 shows the values of MCount and DCount_i during protocol operation.

Table 32 – 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 SDCl. 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 40. 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 40 shows the so-called 1 % share rule of the IEC 61784-3:2021. For SDCl-FS it means, the residual error rate of an SDCl-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 $10^{-9}/h$.

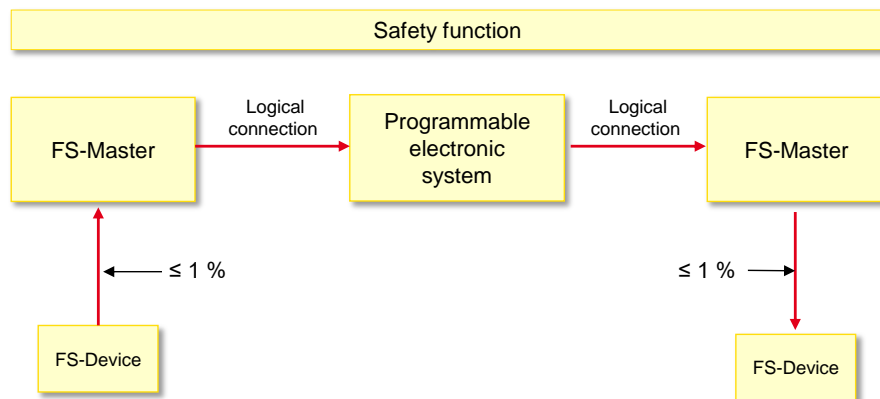


Figure 40 – The 1 % share rule of IEC 61784-3:2021

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

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

It is highly recommended that the 32-bit CRC is chosen, due to the stronger requirements in the IEC 61784-3 for bit error probability.

11.4.8 TADI safety considerations (normative)

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 40).

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 0.5 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 33 lists the three permitted data types (see Annex C).

Table 33 – 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 ($\approx \pm 2$ km, resolution 1 μ m)

11.4.9.3 Qualifier

FS-Devices normally do not require qualifiers (see 11.12.2). 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 11.12.3) the rules in Table 34 for the layout of binary and digital data and their qualifier bits apply.

Table 34 – 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 35 shows the ranking of values and qualifiers.

Table 35 – 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 26 and the protocol signals in 11.4.5.

11.5.2 SCL state machine of the FS-Master

Figure 41 shows the FS-Master state machine for wired SDCI point-to-point communication.

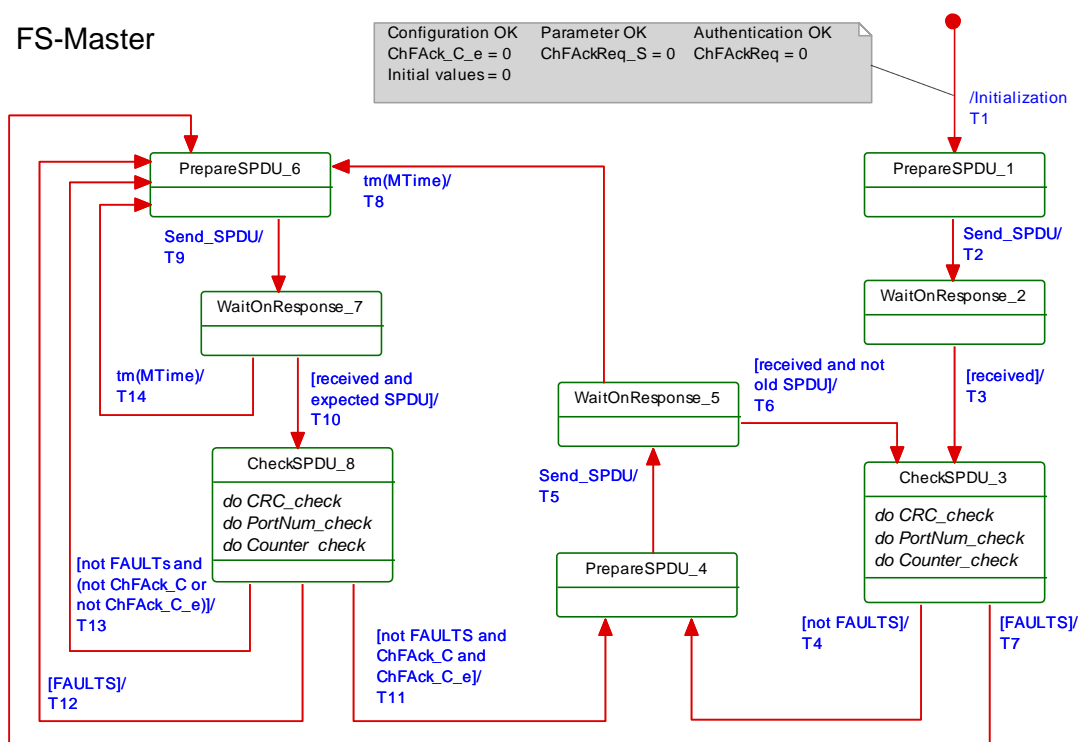


Figure 41 – SCL state machine of the FS-Master

The terms used in Figure 41 are defined in Table 36.

1894

Table 36 – 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

1895

Table 37 – 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 (<i>regular</i>) 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 (<i>regular</i>) 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 (<i>exceptional</i>) 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 36). Hint: A delay time may be required avoiding the impact of an occasional system shutdown.

1896

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 "seed" 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 42 shows the corresponding FS-Device state machine.

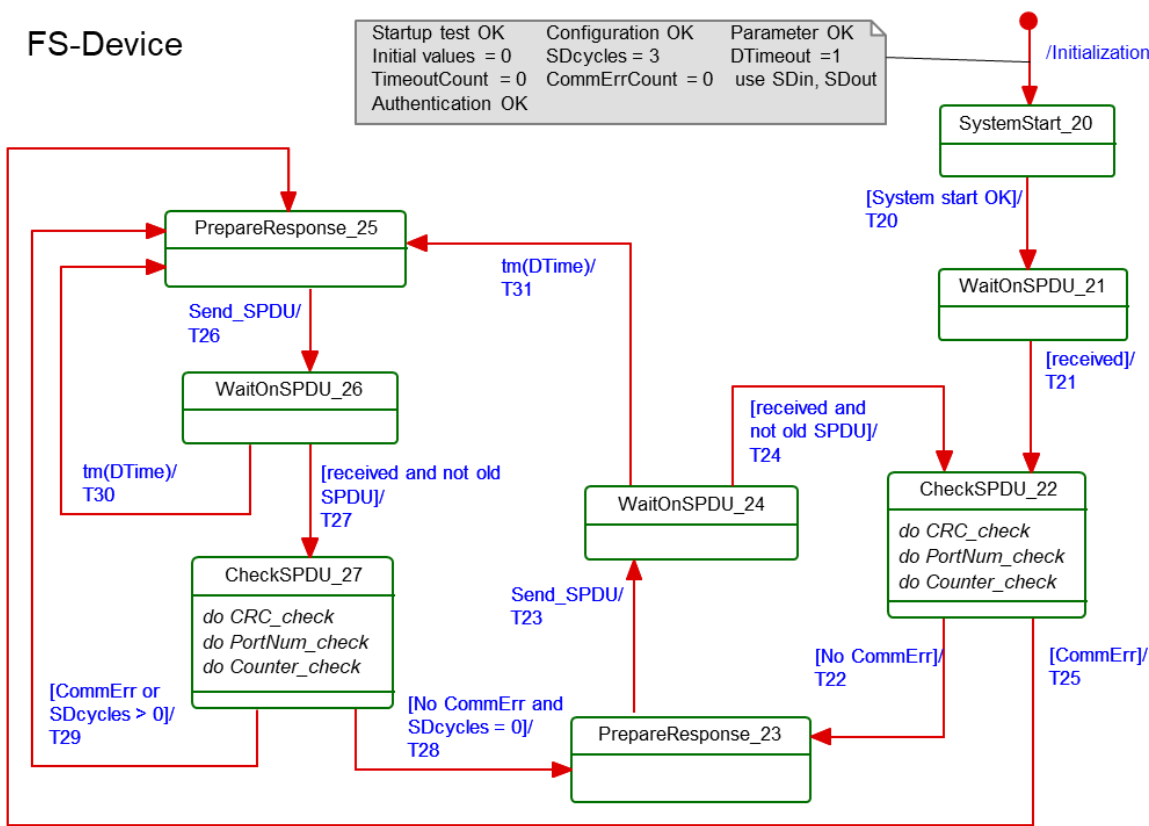


Figure 42 – SCL state machine of the FS-Device

The terms used in Figure 42 are defined in Table 38.

Table 38 – 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 39
SDcycles	See INTERNAL ITEM in Table 39
DTimeout	FSP_WatchdogTime expired
TimeoutCount	See INTERNAL ITEM in Table 39

Table 39 – 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 ChAckReq_DC = ChAckReq. When guard "No CommErr" = true → T22. When guard "CommErr" = true → T25		
23 PrepareResponse	Preparation of (<i>regular</i>) SPDU response for the FS-Master (response message)		
24 WaitOnSPDU	SCL is waiting on next (<i>regular</i>) SPDU from FS-Master not carrying the previous MCount. After FSP_WatchdogTime expired → T31. When SPDU received and guard "MCounter_incremented" = true → T24 (<i>regular</i> cycle)		
25 PrepareResponse	Preparation of (<i>exceptional</i>) 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		
27 CheckSPDU	Check received SPDU from FS-Master for CRC errors; set ChAckReq_DC = ChAckReq. 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	–

TRAN-SITION	SOURCE STATE	TARGET STATE	ACTION
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, 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 “seed” value = "0" and compares with received CRC signature
do PortNum_check		Activity	SCL checks whether PortNum carries the correct FS-Master Port number

INTERNAL ITEM	TYPE	DEFINITION
do Counter_check	Activity	SCL checks whether MCount carries "0" (first SPDU or MTimeout) or the expected subsequent value (all other SPDUs)
NOTE Variables within ACTIONS are defined in 11.3		

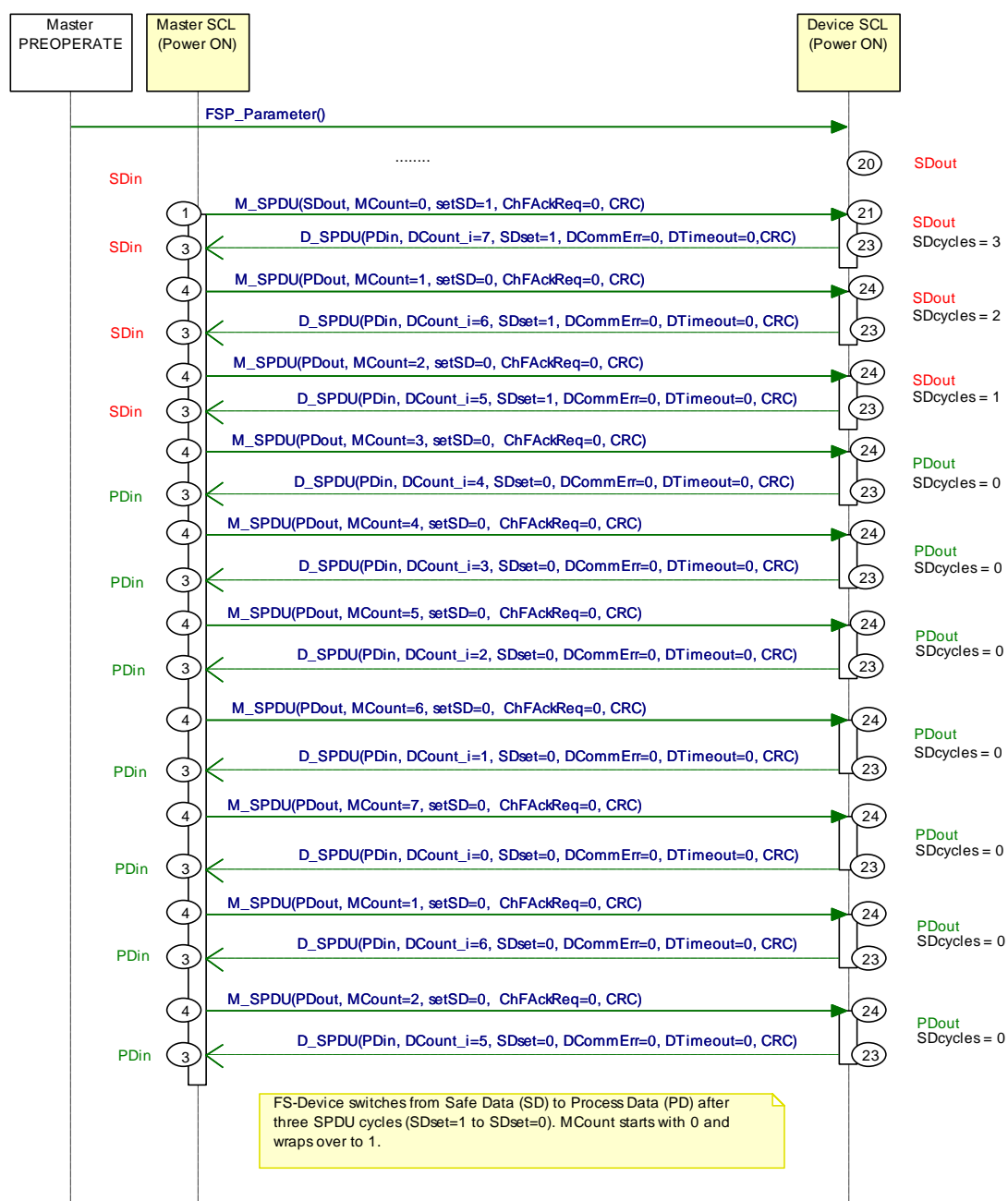
1909

1910 It is very unlikely for an FS-Device to receive SPDUs with all octets "0". The SCL within the FS-
 1911 Device shall ignore such an SPDU. Normally, at least the CRC signature will be "1" if Process
 1912 Data and Control Byte are "0" according to the rules in 11.4.7.

1913 11.5.4 Sequence charts for several use cases

1914 11.5.4.1 FS-Master and FS-Device both with power ON

1915 Figure 43 shows the sequence chart of a regular start-up of both FS-Master and FS-Device.



1916

1917

Figure 43 – 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 44 shows the sequence chart of regular operation while FS-Master has been switched OFF and ON again.

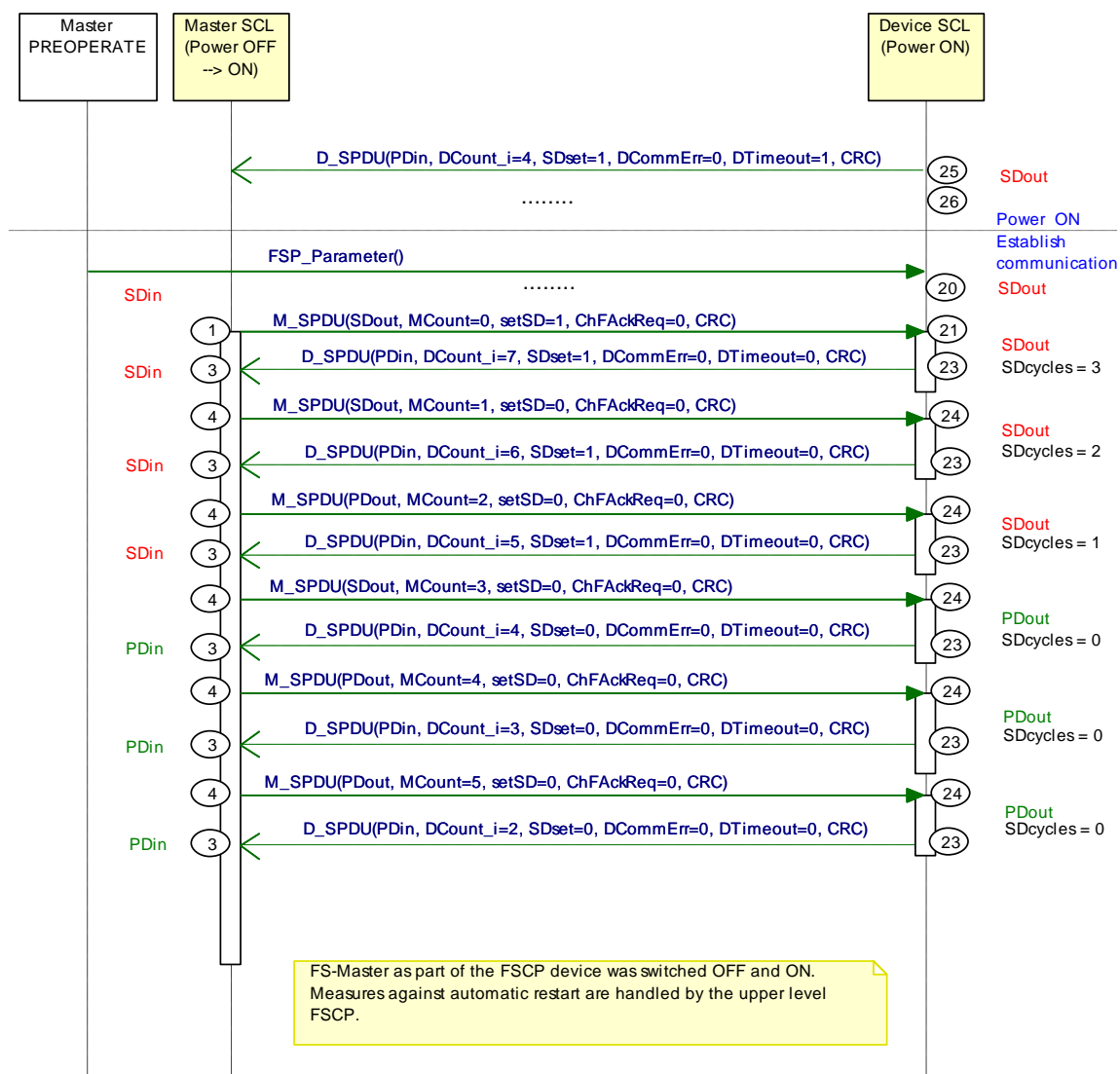


Figure 44 – FS-Master power OFF → ON

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

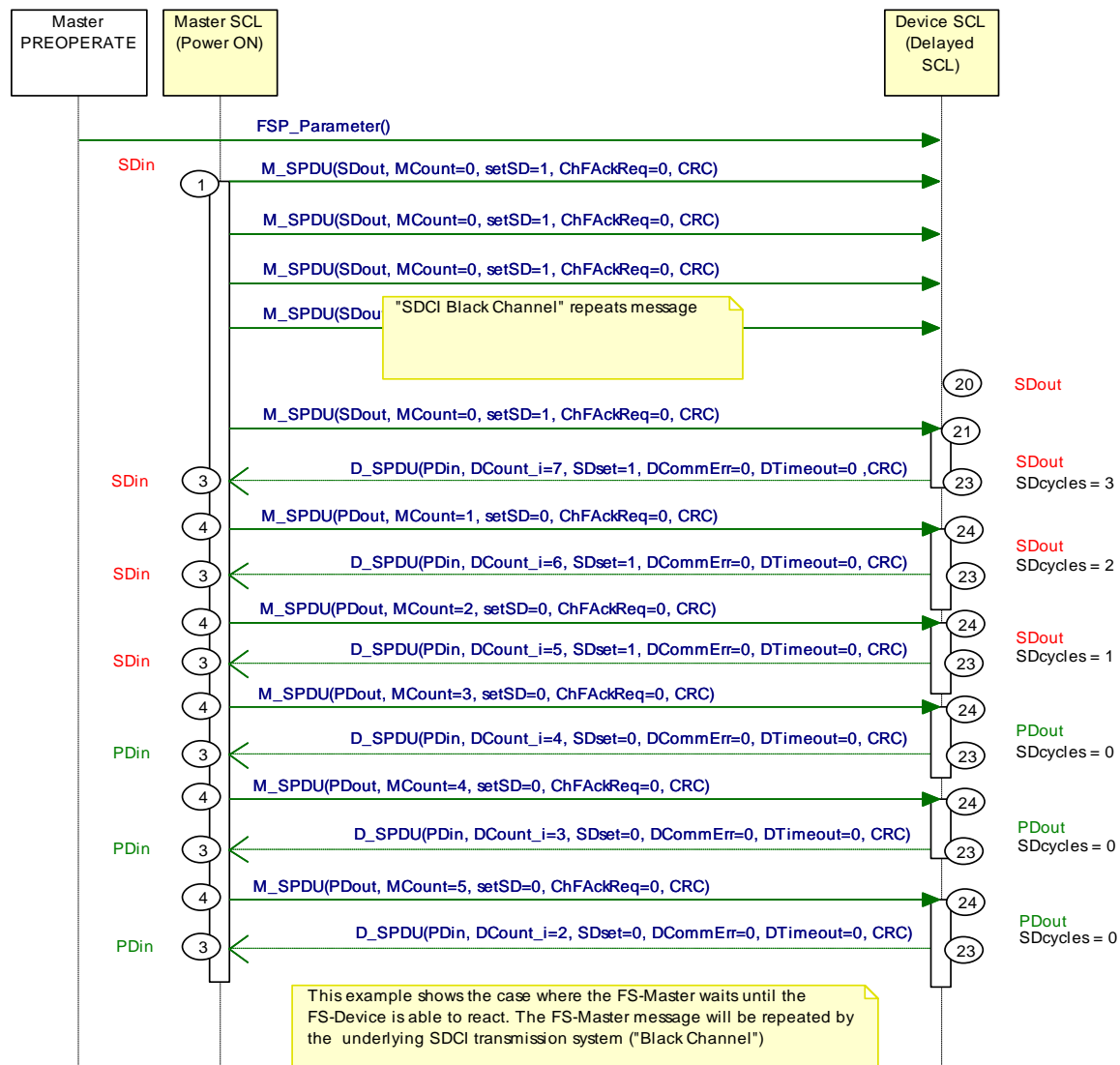


Figure 45 – FS-Device with delayed SCL start

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 SDCl is reserved for the non-safety part of the entire message.

11.5.4.4 FS-Device with power OFF and ON

Figure 46 shows the sequence chart when the FS-Device switches power OFF and ON again.

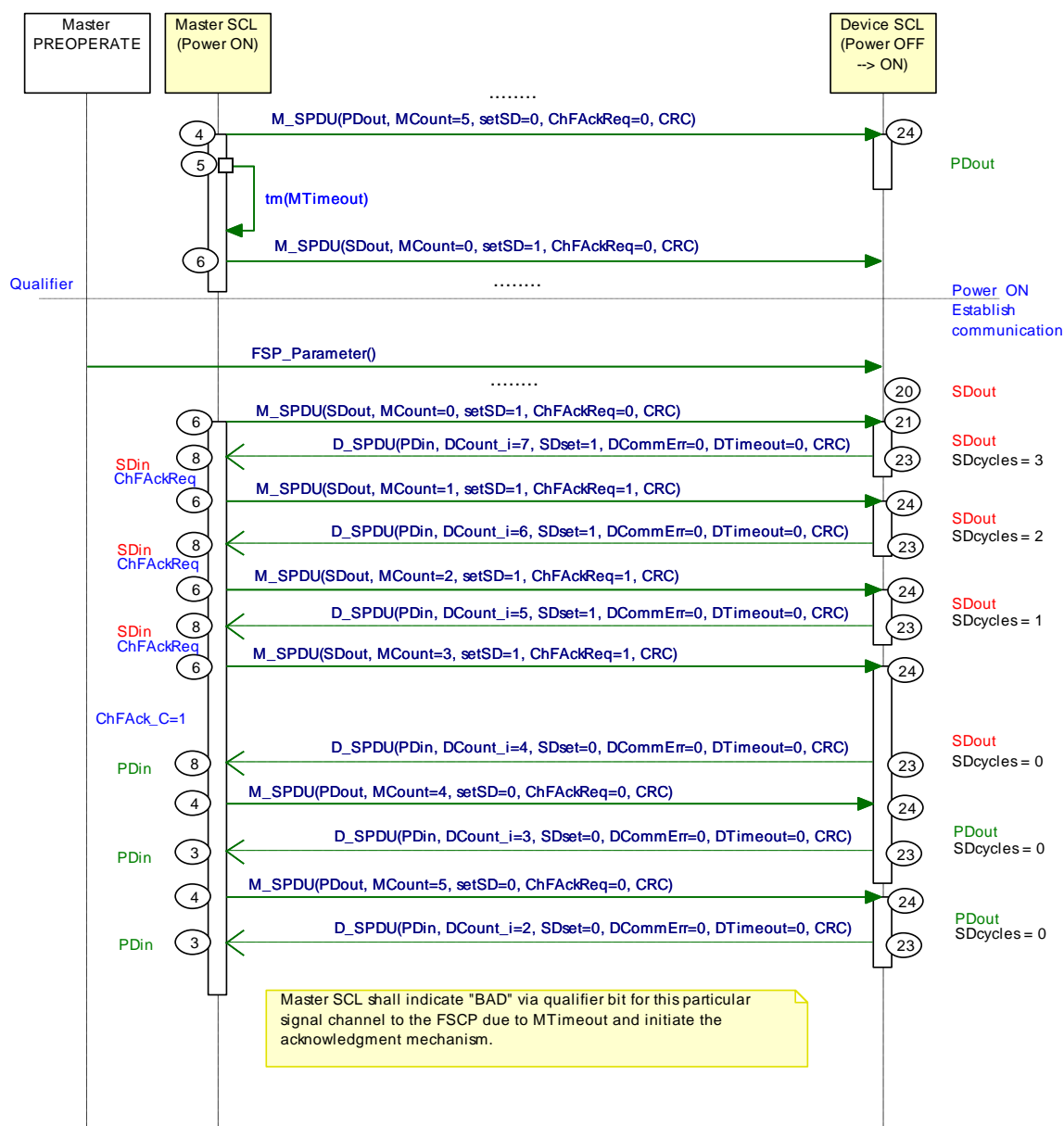


Figure 46 – FS-Device with power OFF and ON

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.

11.5.4.5 FS-Master detects CRC signature error

Figure 47 shows the sequence chart when the FS-Master detects a CRC signature error.

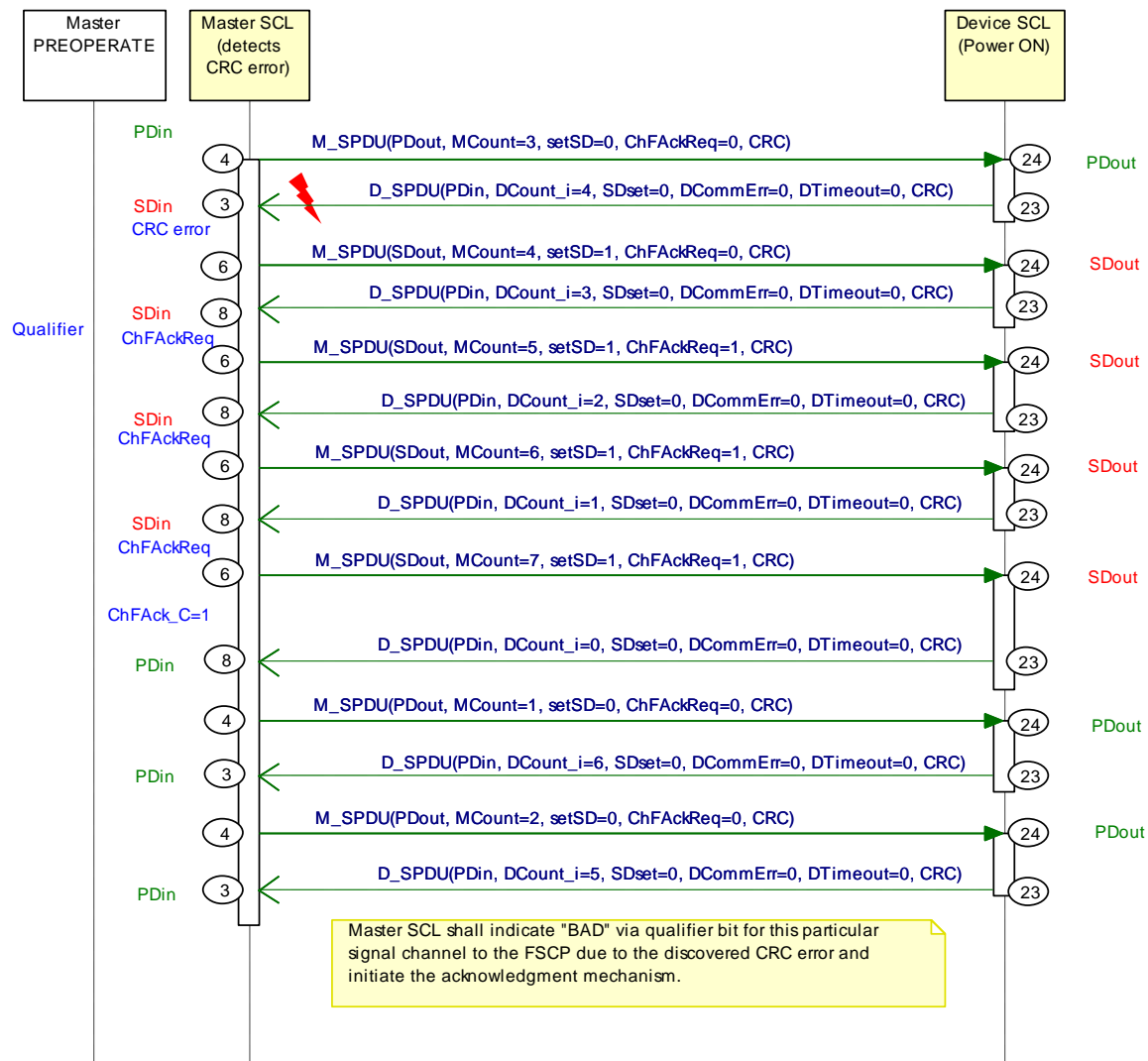


Figure 47 – FS-Master detects CRC signature error

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.

11.5.4.6 FS-Device detects CRC signature error

Figure 48 shows the sequence chart when the FS-Device detects a CRC signature error.

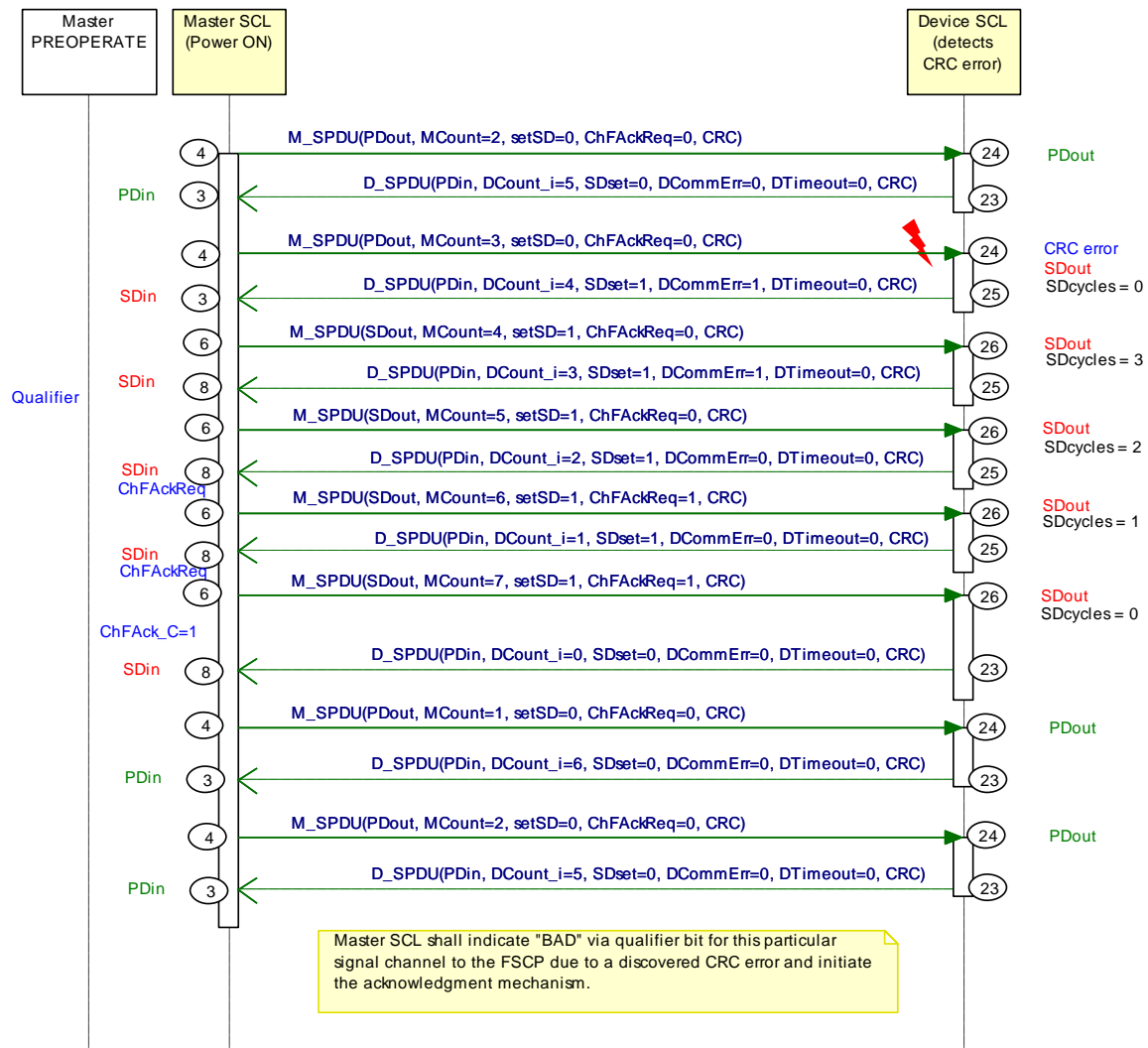


Figure 48 – FS-Device detects CRC signature error

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 ChAckReq signal. This signal is indicated also to the FS-Device via ChAckReq (=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 ChAck_C (=1) arrived.

11.5.5 Monitoring of safety times

Figure 49 illustrates SDCI times and safety times.

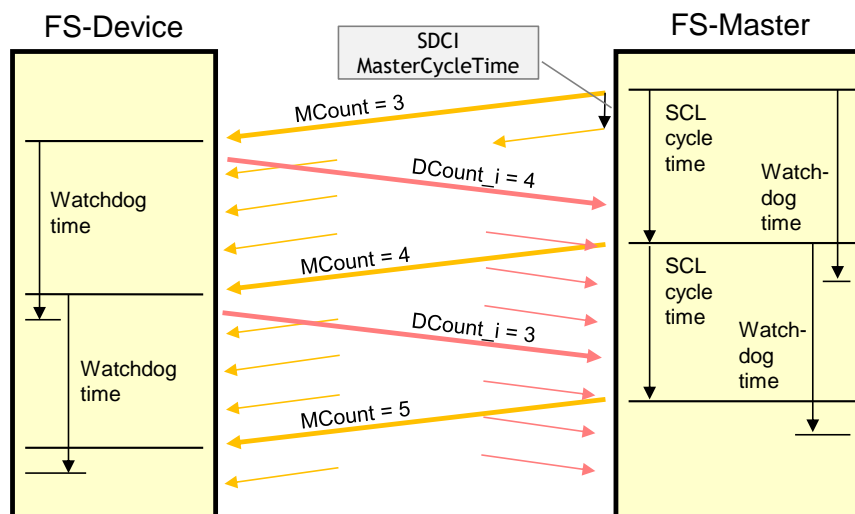


Figure 49 – Monitoring of the SCL cycle time

The base SDCI system ("black channel") transmits SPDUs within the MasterCycleTime 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 40 shows timing constraints.

Table 40 – Timing constraints

Item	Constraints
Synchronization	At each start-up and after an FS-Master timeout, the FS-Master SCL uses MCount = 0
SCL cycle time	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 49)
Watchdog time	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).
Counter check	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".
Repetition	Repetition in case of detected incorrect CRC signatures is not provided
PFH-Monitor	The FS-Master holds the information about the reliability of both SPDU transmissions from the FS-Device and to the FS-Device (see Table 31, 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). The PFH value shall be specified for the FS-Master e.g. in the manual.
PFH-Monitor time (h)	10

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 26 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

2031 communication participants sending messages in such a manner that services are delayed or
2032 denied (for example FIFOs in switches, bridges, routers).

2033 NOTE 1 SDCI provides a point-to-point communication interface with defined message sequences and thus the
2034 probability for congestion and storage of messages is extremely low.

2035 *Countermeasures:*

2036 A consecutive counter in each message (MCount/DCount) together with a watchdog timer
2037 (MTime/DTime) will detect unacceptable delays.

2038 **11.5.6.7 Insertion**

2039 Due to a fault or interference, a message is received that relates to an unexpected or unknown
2040 source entity.

2041 NOTE 1 These messages are additional to the expected message stream, and because they do not have expected
2042 sources, they cannot be classified as Correct, Unintended repetition, or Incorrect sequence.

2043 NOTE 2 SDCI provides a point-to-point communication interface (Port) and thus the probability for insertion of
2044 messages is extremely low.

2045 *Countermeasures:*

2046 The receiver will detect any incorrect sequence due to the stringently sequential expectation of
2047 the MCount and DCount values.

2048 **11.5.6.8 Masquerade**

2049 Due to a fault or interference, a message is inserted that relates to an apparently valid source
2050 entity, so a misdirected non-safety related message may be received by a safety related
2051 participant, which then treats it as safety related correct message.

2052 NOTE 1 Communication systems used for safety-related applications can use additional checks to detect
2053 Masquerade, such as authorised source identities and passphrases or cryptography.

2054 NOTE 2 SDCI provides a point-to-point communication interface (Port) and thus the probability for insertion of
2055 messages is extremely low.

2056 *Countermeasures:*

2057 In case of NSR data instead of a regular SPDU, the CRC signature mechanism of the SCL will
2058 detect this incident.

2059 After changes of wiring, the FS-Devices can detect misconnections through the
2060 FSP_Authenticity1/2 and FSP_Port parameters (see A.2.1 and A.2.2) at start-up.

2061 **11.5.6.9 Addressing**

2062 Due to a fault or interference, a safety related message is delivered to the incorrect safety
2063 related participant, which then treats reception as correct. This includes the so-called loopback
2064 error case, where the sender receives back its own sent message.

2065 NOTE 1 The probability of not detecting a misdirected non-safety related message is lower than the probability of
2066 not detecting a misdirected safety related message since the SPDU structures are similar due to the
2067 shared protocol procedures.

2068 NOTE 2 SDCI provides a point-to-point communication interface (Port) and thus the probability for insertion of
2069 messages is extremely low. However, FS-Master may use internal bus mechanisms to address Ports.

2070 *Countermeasures:*

2071 Port addressing errors can be detected by the Port number (PortNum) within the SPDU.

2072 After changes of wiring, the FS-Devices can detect misconnections through the
2073 FSP_Authenticity1/2 and FSP_Port parameters (see A.2.1 and A.2.2) at start-up.

2074 **11.5.6.10 Loop-back**

2075 A special addressing error is the so-called loopback error case, where the sender receives back
2076 its own sent message.

2077 *Countermeasures:*

2078 SDCI-FS provides for inverted values of MCount as DCount and inverted values of the Port
2079 number (PortNum) returned from the FS-Device.

2080 **11.5.7 Start-up (communication)**

2081 An FS-Device starts always after an FS-Master since the FS-Master shall be the only one to
2082 power-up at least the communication part of the FS-Device. Both devices usually require time
2083 for safety self-tests that may exceed the standard timings defined in IEC 61131-9.

2084 Due to the initial behavior of an FS-Device as an OSSDe, the start-up is coordinated and
2085 specified in 5.7, 7.2, and 7.3.

2086 The start-up of the underlying SDCI communication system is specified in IEC 61131-9 and
2087 illustrated in an abstract and simplified manner in Figure 55 for easier comprehension. Any
2088 deviating FSP authenticity or protocol parameter CRC signature shall lead to a safe state of the
2089 particular FS-Master Port and prevent the SCL from being started.

2090 **11.6 SCL management**

2091 **11.6.1 Parameter overview (FSP and FST)**

2092 Annex A specifies several functional safety related parameters for communication protocol
2093 services (FSP) as well as for the handling and integrity purposes of FS-Device technology
2094 parameters (FST).

2095 The parameters are subdivided into 4 groups:

- 2096 • Authenticity
- 2097 • Safety communication
- 2098 • FS-I/O structure description
- 2099 • Auxiliary parameters

2100 The authenticity parameters combine the safety connection ID ("A-Code") of the FS-Master
2101 (assigned by the upper-level FSCP system) with the Port number of the connected FS-Device.
2102 Due to the point-to-point nature of the FS-Device communication with its Master, a one-time
2103 check at start-up is sufficient to ensure authenticity (see 11.8.4).

2104 The Safety Communication Layers (SCL) require parameters for protocol versions, protocol
2105 modes such as CRC-32, watchdog for timeliness, CRC signature to secure technology
2106 parameters, and a CRC signature to secure the safety communication parameters.

2107 The next group contains a description of the FS I/O data structure, the FS-Device wants to
2108 exchange with the FSCP-Host. This description facilitates the mapping to the description which
2109 some FSCP systems require for set-up. During the mapping process the FS-Master Tool
2110 appends the qualifier bits, which are necessary for Port-selective passivation.

2111 Auxiliary parameters are specified for several purposes. For example, to secure the functional
2112 safety parameter description within the IODD, to support the automatic calculation of approxi-
2113 mate values of safety function response times, and to inform about the start-up self-testing time
2114 of an FS-Device until the Ready pulse appears.

2115 Figure 50 shows an overview of the components and the activities around parameterization.

2116 An FS-Master as a gateway comes with an associated description file for the upper-level system
2117 (FSCP). With the help of an engineering tool and these parameters, the FS-Master receives
2118 during commissioning for example its FSCP connection ID ("A-Code" for authenticity) and its
2119 FSCP watchdog time ("T-Code" for timeliness). Thus, the FSCP communication cycles are
2120 independent from the SDCI-FS communication cycles between FS-Master and FS-Device.

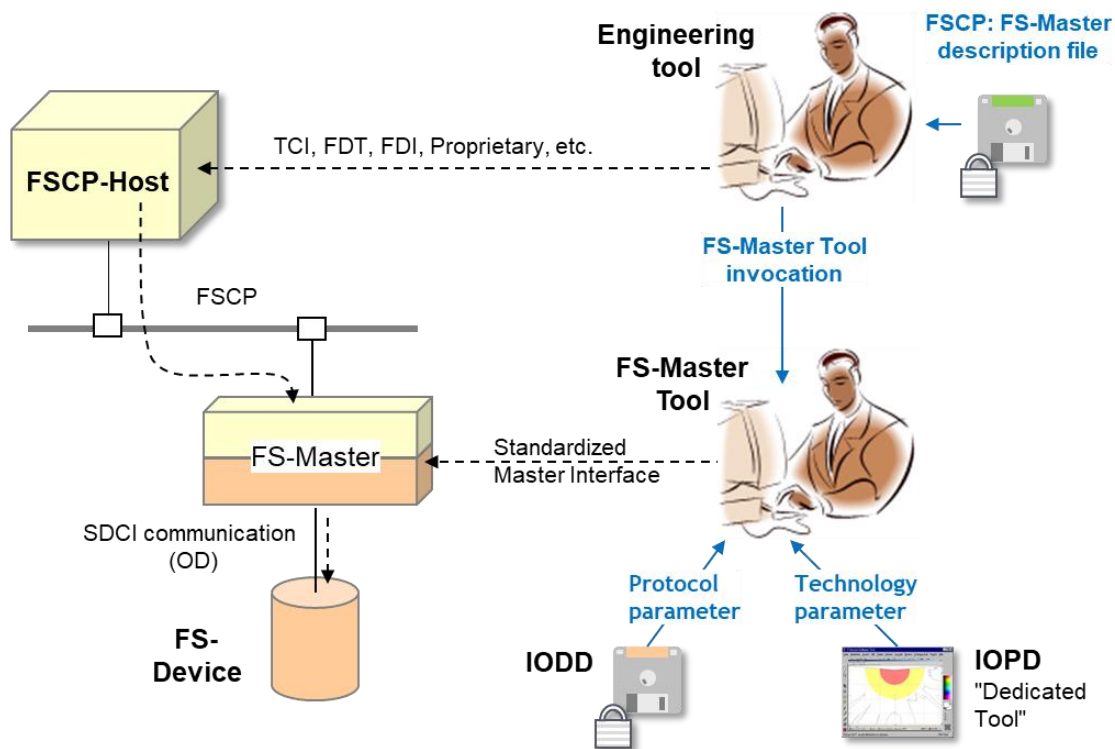


Figure 50 – Parameter types and assignments

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 50 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.

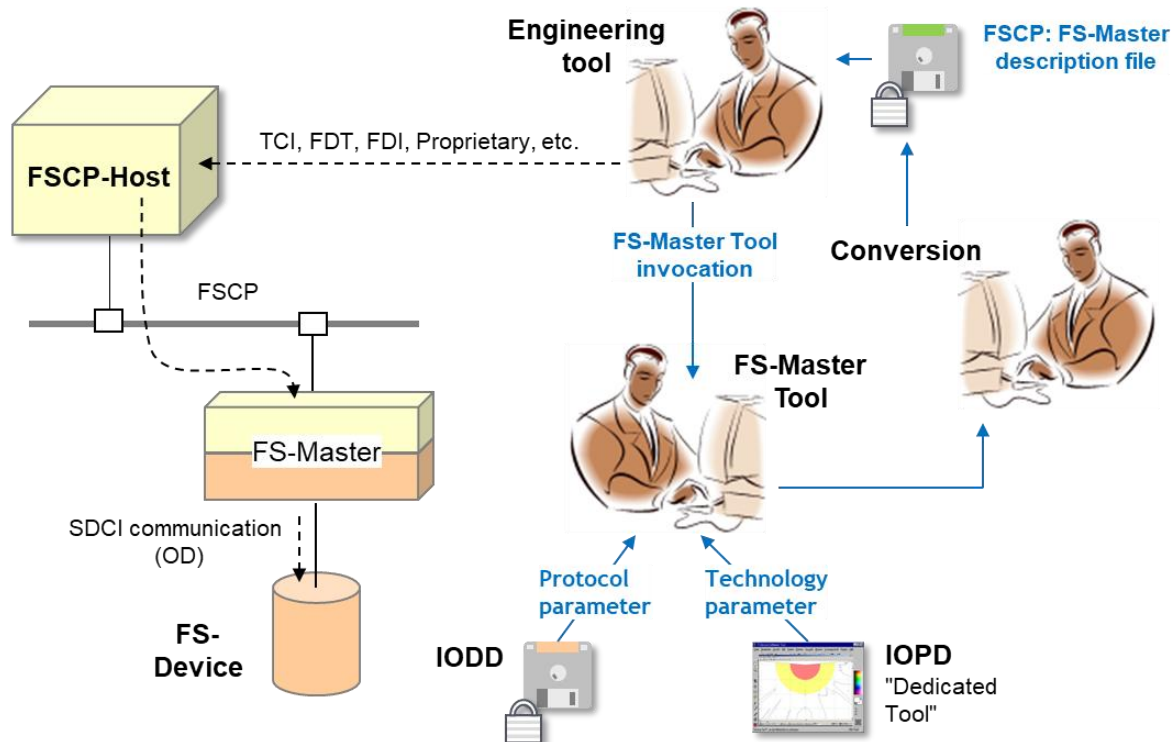


Figure 51 – FSCP-Host-centric system

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.

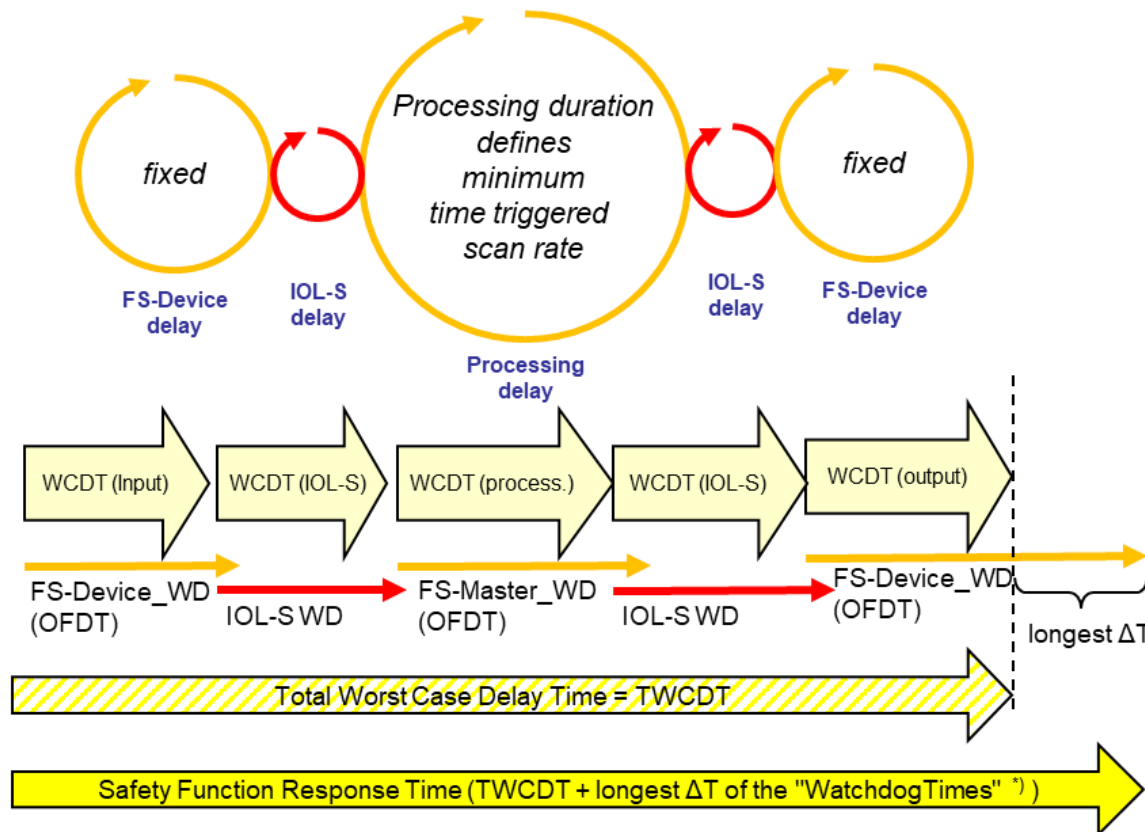
11.7 Safety function response time

11.7.1 General concepts and accuracies

Figure 52 illustrates the effects of the worst-case delay times (WCDDT) and one fault delay times (OFDDT) of the components involved in a safety function based on a pure FS-Master and FS-Device system.

Therefore, since it is mandatory for all components to provide WCDDT and OFDDT in user manuals, FS-Master tools are enabled to provide values for the total worst case delay time (TWCDT) and safety function response time.

An FS-Master shall also provide values for FS-Master_WD (OFDDT), usually derived from program processing duration and for IOL-S WD for the output side.



Key WCDT = Worst Case Delay Time OFDT = One Fault Delay Time *) not necessarily the output device!

Figure 52 – SFRT of a stand-alone FS-Master with processing

Only one fault shall be assumed per trip. The watchdog time or OFDT with the largest impact on the safety function response time (SFRT) shall be considered for a safety function. For a machine usually an overtravel measurement (usually at least 10 measurements) is performed.

Table 41 shows the accuracies and tolerances to be used for timings.

Table 41 – Accuracies and tolerances for timings

Item	Accuracy	Remarks
Measurement accuracy	+/- 1 %	–
Permitted watchdog time tolerance	+/- 10 %	–

Figure 53 illustrates the effects of the worst-case delay times (WDCT) and one fault delay times (OFDT) of the components involved in a safety function based on FS-Master and FS-Devices integrated in a fieldbus functional safety communication profile (FSCP), see for example [26].

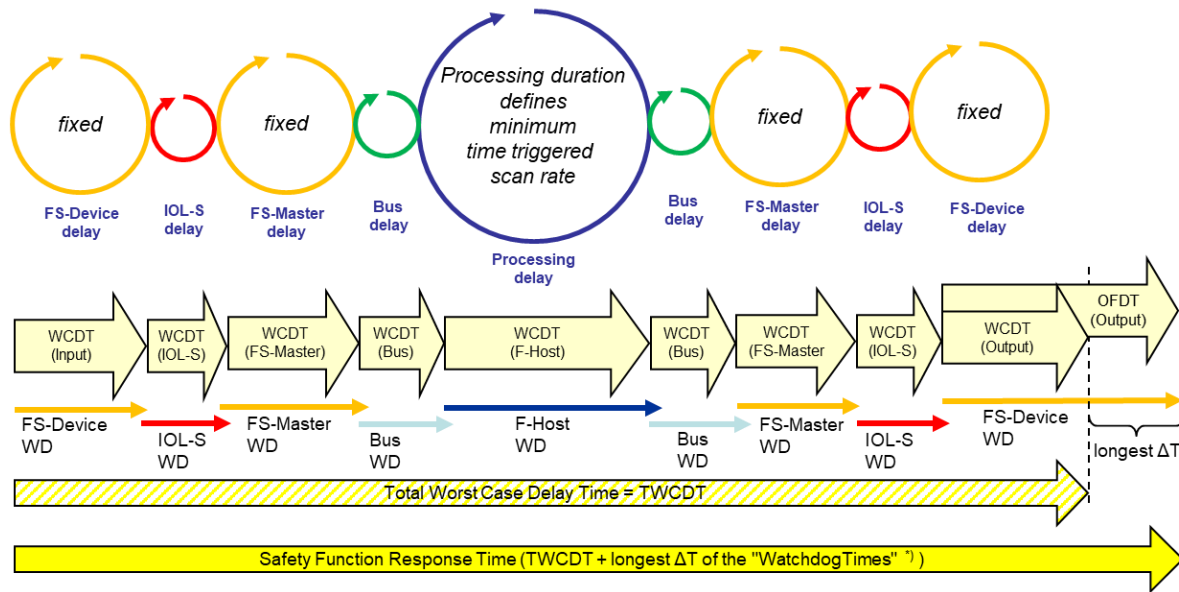


Figure 53 – SFRT including IOL-S and FSCP

The manufacturer of the IO-Link Device specifies the MinCycleTime of the Device.

$$\text{MinCycleTime} = t_{bc,min}$$

In addition, the Manufacturer of the IO-Link FS-Device specifies the io-update time $t_{io,update}$, (see A.2.6). Based on the description the io-update time shall cover the processing time of the SCL layer in the FS-Device plus the transmission time of the black channel including all repetitions and some synchronization delay. Thus the IO-update time shall fulfill the following requirement.

$$t_{io,update} \geq 6.6 t_{bc,min} + t_{scl,d}$$

Where $t_{scl,d}$ is the processing time of the FS-Device, which is the time between the capturing of an update of the SPDU-Out from the FS-Master by the SCL and the updating of the SPDU-IN forwarded to the FS-Master. The io-update time is published in the IODD of the FS-Device as DefaultValue of the parameter FSP-Watchdog. With the Master tool, the value $t_{io,update}$ is extended by the processing time of the FS-Master $t_{scl,m}$ and the MinCycleTime $t_{bc,min}$, which is replaced by the selected nominal master cycle time.

$$t_{bc,nom} \geq t_{bc,min}$$

The resulting Watchdog timeout t_{WD} is stored in the FSP-Watchdog parameter.

$$t_{WD} > 6.6 t_{bc,nom} + t_{scl,d} + t_{scl,m}$$

The SCL Master cycle time $t_{scl,m}$ is a property of the FS-Master that must be made available by the Manufacturer of the FS-Master to the Master tool and the FS-Master tester. The duration of $t_{scl,m}$ covers the processing time between the capturing an SPDUIn from the FS-Device by the SCL and the update of the SPDUOut by the FS-Master.

11.7.2 Integration Aspects

In 11.7.1, the general concepts are explained also for a more complex FS-Master integrated in a fieldbus's functional safety communication profile (FSCP) according to the IEC 61784-3 series. In this case, usually the FS-Master plays only the role of a mapper of Process Data from one safety communication system to the other.

The designer/manufacturer of such a mapping FS-Master/Gateway shall provide WCDT and OFDT for the mapping part to enable computer-aided approximation of a safety function

2205 response time. Integration specifications to FSCPs should comprise definitions and descriptions
2206 how to achieve these values.

2207 **11.8 Integrity measures**

2208 **11.8.1 IODD integrity**

2209 The parameters specified in Annex A are coded in an IODD file using XML. In order to protect
2210 the safety parameter description within this file, a CRC signature ("FSP_ParamDescCRC") shall
2211 be calculated across its safety-related parts (see Annex D and Annex E.5.6). Usually, the IODD
2212 file travels many ways and the CRC signature helps detecting potentially scrambled bits.

2213 **11.8.2 Tool integrity**

2214 When opening the IODD, the FS-Master Tool (interpreter of the IODD file) shall calculate the
2215 CRC signature across the safety-related parts and compare the result with the parameter
2216 "FSP_ParamDescCRC".

2217 During the data manipulations within the FS-Master Tool as well as within Device Tools/IOPDs
2218 ("Dedicated Tools") such as display, intended modification, storage/retrieval, and down/upload,
2219 parameter values could become incorrect. It is the responsibility of the designer to develop the
2220 software tools according to the software safety level requested in ISO 13849-1 or IEC 61508-
2221 3.

2222 **11.8.3 Transmission integrity**

2223 Since communication between the FS-Master Tool and the FS-Device is proprietary, it is the
2224 responsibility of the FS-Master Tool to ensure transmission integrity and authenticity, for
2225 example through CRC signatures and/or read back (see Table 26 and D.3.1).

2226 **11.8.4 Verification record**

2227 In either the FS-Master-centric or in the FSCP-Host-centric approach an FSP_VerifyRecord of
2228 parameter data is stored in the FS-Master per Port/FS-Device as shown in Figure 54.

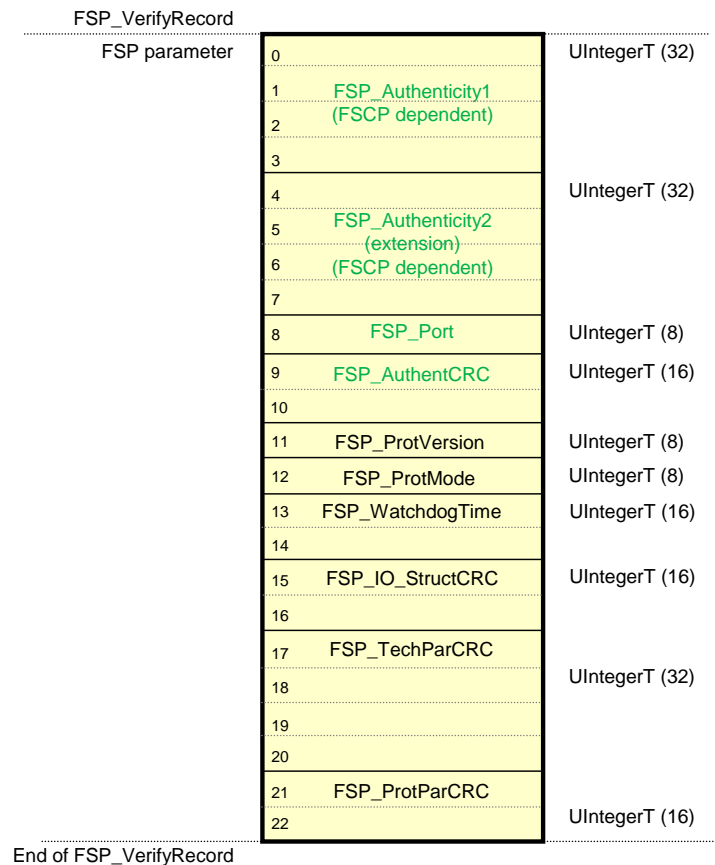


Figure 54 – Structure of the FSP_VerifyRecord

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.

11.8.5 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.

11.8.6 Storage integrity

Both records (authenticity and protocol) of Figure 54 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 55 and described in 10.4.3.1 and A.2.10. Figure 55 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 32).

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

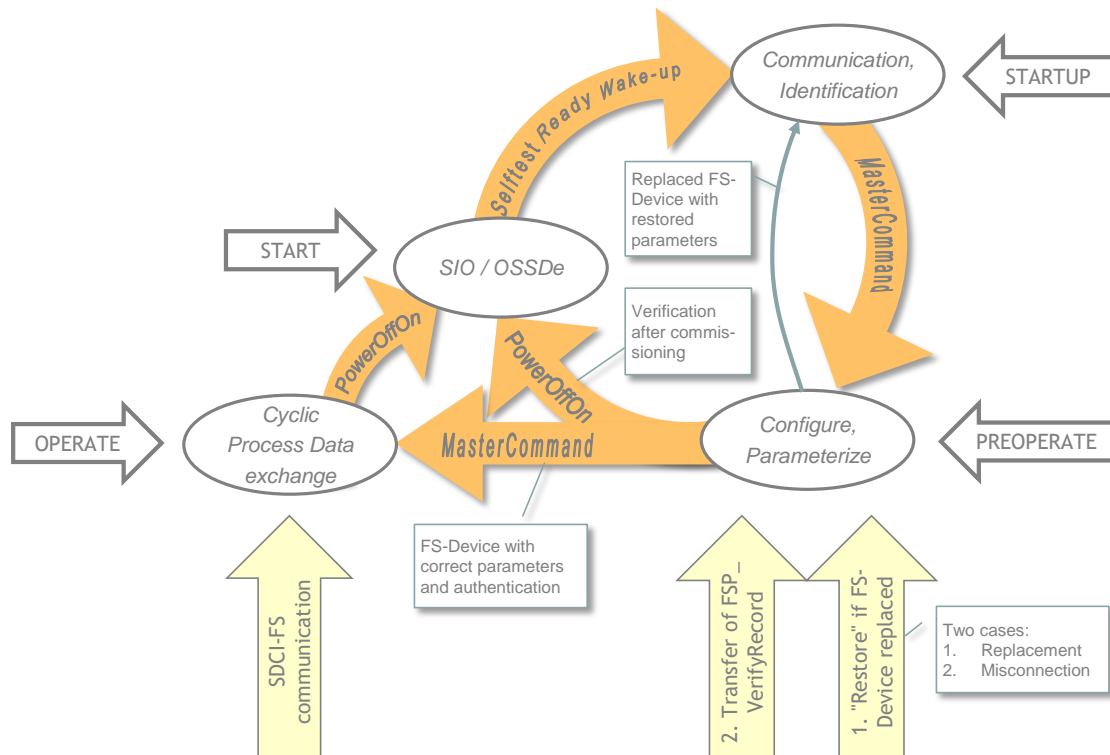


Figure 55 – Start-up of SDCI-FS

In case the FS-Device has been replaced, all parameters including the FST and the FSP parameters are "restored" from Data Storage if

- the FS-Device carries all authenticity parameters = "0" (Back-to-box), or
- all of the FSP- and FST-Parameter are identical.

If Authenticity is not "0", and any of the FSP- and FST-Parameter differs, the FS-Device shall reject with error code 0x8022 and keep all the existing parameters (see 9.4, E.5.7, and step 1. in Figure 55).

Update of standard (non safe) parameters, FST and FSP parameters are different (between Data Storage and FS-Device) and authenticity is not "0" → accept all parameters from Data Storage.

Replace with another FS-Device, authenticity is "0" (Back-to-box) → accept all parameters from Data Storage.

Misconnection if there are differences, authenticity is not "0" and FST or FSP parameters are different → reject Data Storage with error code 0x8022.

Table 42 – restore from Data Storage

Authenticity (FSP_Authenticity1, FSP_Authenticity2, FSP_Port, FSP_AuthenticCRC)	FST / FSP parameters different (between Data Storage and FS-Device)	restore from Data Storage
= 0	yes	yes
= 0	no	- (NOTE)
!= 0	yes	no
!= 0	no	yes
NOTE This combination is not possible and is only listed for the sake of completeness.		

11.8.7 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).

11.8.8 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 56, FST-Parameters section). Values can be assigned as for non-safety parameters only during "commissioning-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.

Topology	Identification	Parameters	Observation	Diagnosis	Device Library
Toplevel ... - Master - Port 1: Device aa - Port 2: Device b - ... - Port n: Device xxx	SDCI-FS protocol parameters (FSP) FSP_Authenticity_1 FSP_Port FSP_ProtMode FSP_Watchdog FSP_TechParCRC: 0x34FA2C43 Device Tool: THC26				Vendor 1 - Device a V1.03 - Device b V1.23 - Device c V2.00 ... Vendor 2 - Device aa V0.99 - Device bb V1.12 ...
	Technology parameters (FST) Filter: 26 Discrepancy: 5 Redundancy: yes Test cycle: 3				

Two hand control THC26

Filter	26
Discrepancy	5
Redundancy	yes
Test cycle	3
CRC signature	0x34FA2C43

③ Confirm values
Confirm

Figure 56 – Securing of FST parameters via Dedicated Tool

After test and validation, the Dedicated 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 Dedicated Tool then calculates the CRC signature across the instance data of the FST parameters (see "CRC signature" in Figure 56), 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.

11.8.9 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 57 shows an example.

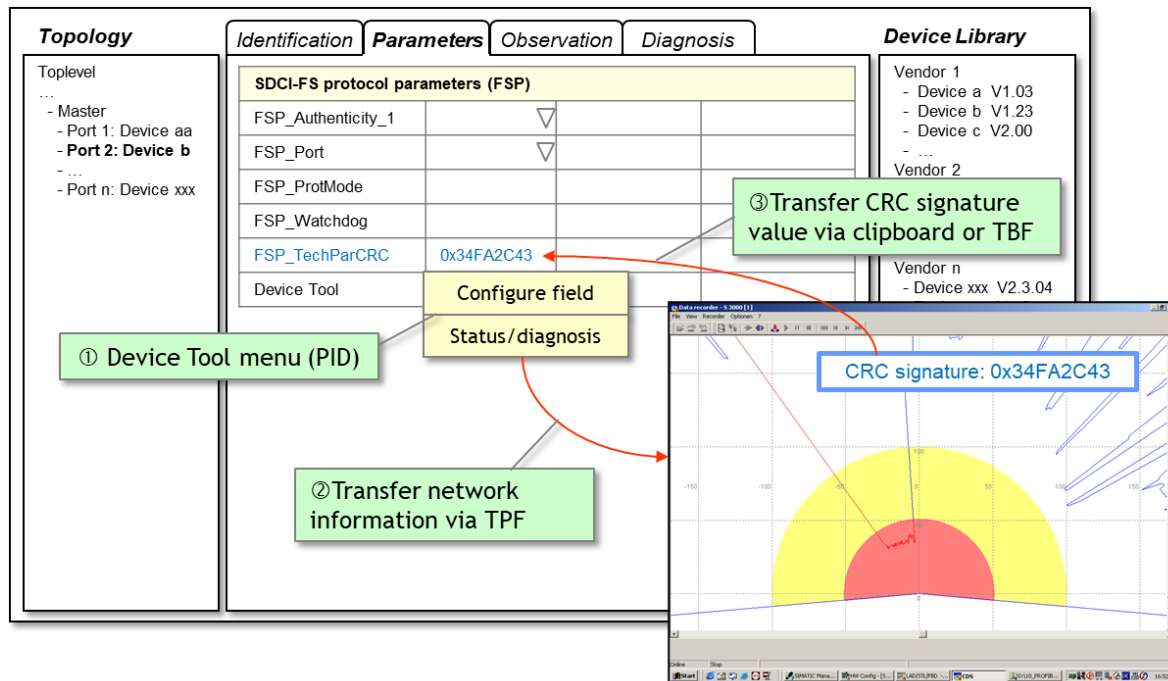


Figure 57 – Modification of FST parameters via Dedicated Tool

Such a Dedicated Tool requires communication with its particular FS-Device and therefore access to a Communication Interface (see Annex F.4). 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.

11.9 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 58.

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 58). 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 58).

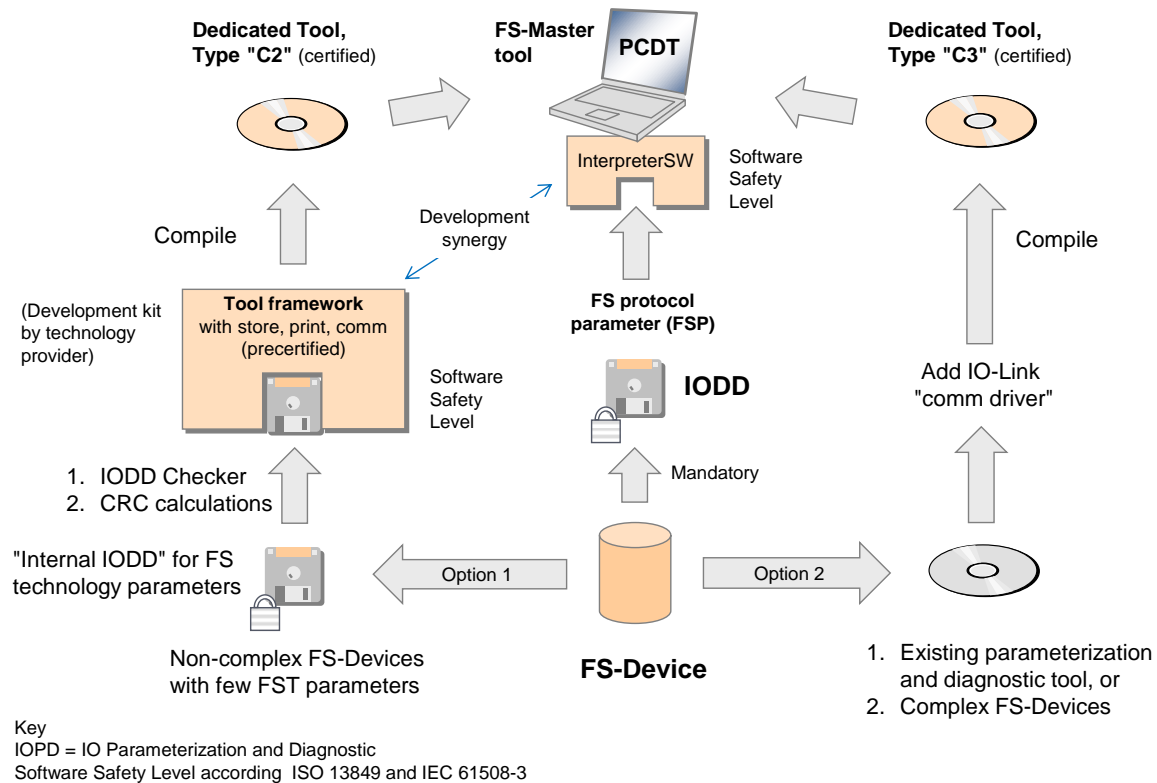


Figure 58 – Creation of FSP and FST parameters

In any case, the dedicated tool (IODD) 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.

11.10 Integration of Dedicated Tools (IODD)

11.10.1 IODD 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 (IODD) as shown in Figure 58.

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

11.10.2 Standard interfaces

Usually, Master Tools are integrated using existing standards such as FDT, the FDI (see [9]), 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 59 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-Dedicated Tool (IODD) does not have to know about the fieldbus environment it is connected to. The example in Figure 59 illustrates how it sends a "Read Index 0x4231" service

2356 and how the FS-Master Tool packs this service into a fieldbus container and passes it to the
2357 fieldbus Communication Interface.

2358 The addressed FS-Master is connected to the fieldbus and receives this container. It unpacks
2359 the SDCI Read service and performs it with the addressed FS-Device connected to a Port.

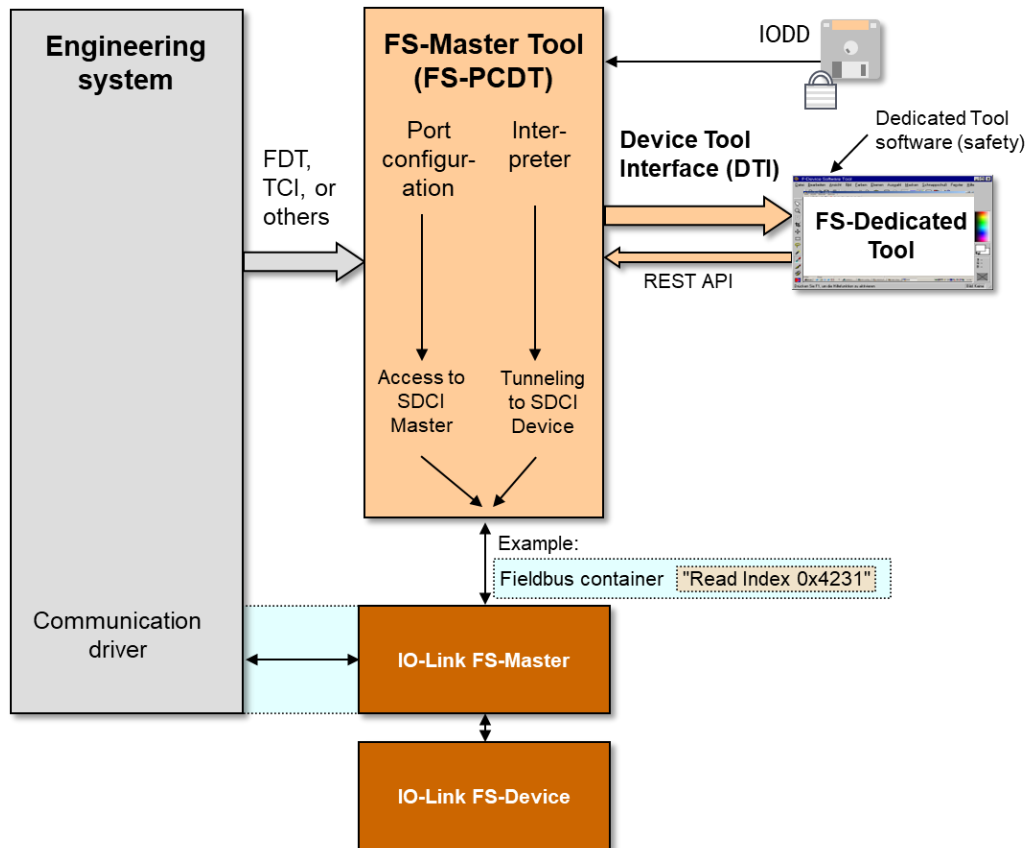


Figure 59 – Example of a communication hierarchy

11.10.3 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 [10] Clause 4.15.5.

F.3.5 and F.7.4 specify the features for this "backward channel".

11.11 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).

11.12 Passivation

11.12.1 Motivation and means

Figure 60 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.

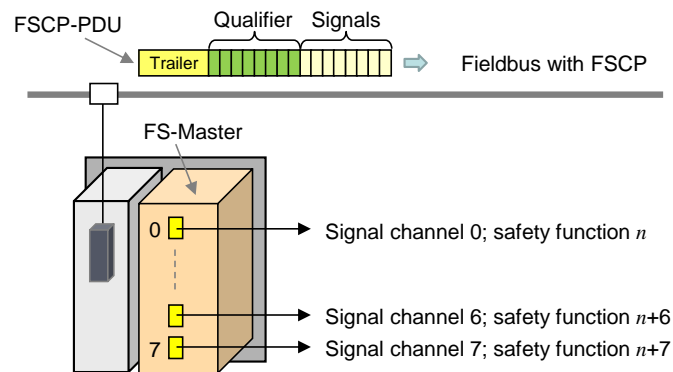


Figure 60 – Motivation for Port selective passivation

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.

11.12.2 Port selective (FS-Master)

In 11.12.1 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 30) shall be used for that purpose. It is not safety related.

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

11.12.3 Signal selective (FS-Terminal)

Figure 12 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.

11.12.4 Qualifier settings in case of communication

Figure 61 illustrates the embedding of the qualifier handler in case of FS_PortModes "SafetyCom" (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.

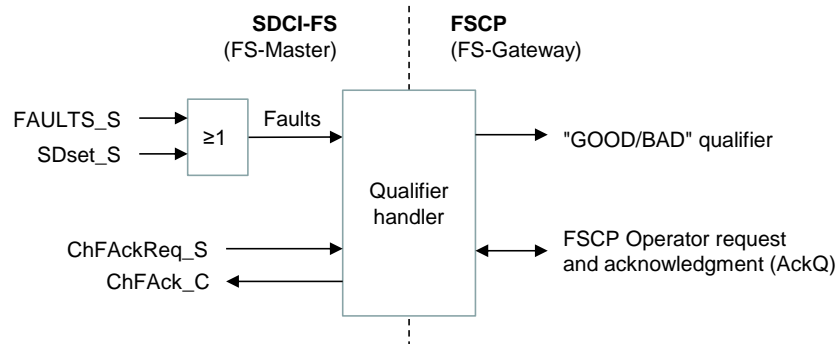


Figure 61 – Qualifier handler (communication)

The qualifier bits "GOOD/BAD" shall be set according to the definitions in Table 43 during the FSCP mapping procedure.

Table 43 – Qualifier bits "GOOD/BAD"

Faults	Qualifier	Signal state
0	GOOD	1
1	BAD	0

11.12.5 Qualifier handling in case of FS-DI

Figure 62 illustrates the embedding of the qualifier handler in case of FS_PortModes "FS_DI" (see 10.4.2). Definitions of Table 43 apply.

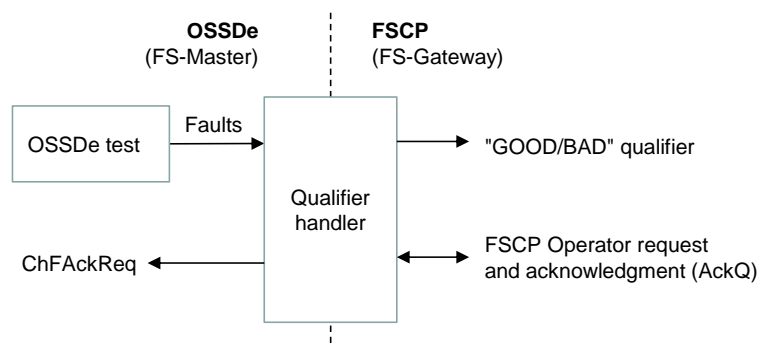


Figure 62 – Qualifier handler (FS-DI)

Figure 63 shows the state machine for the behavior of the qualifier handler (FS-DI).

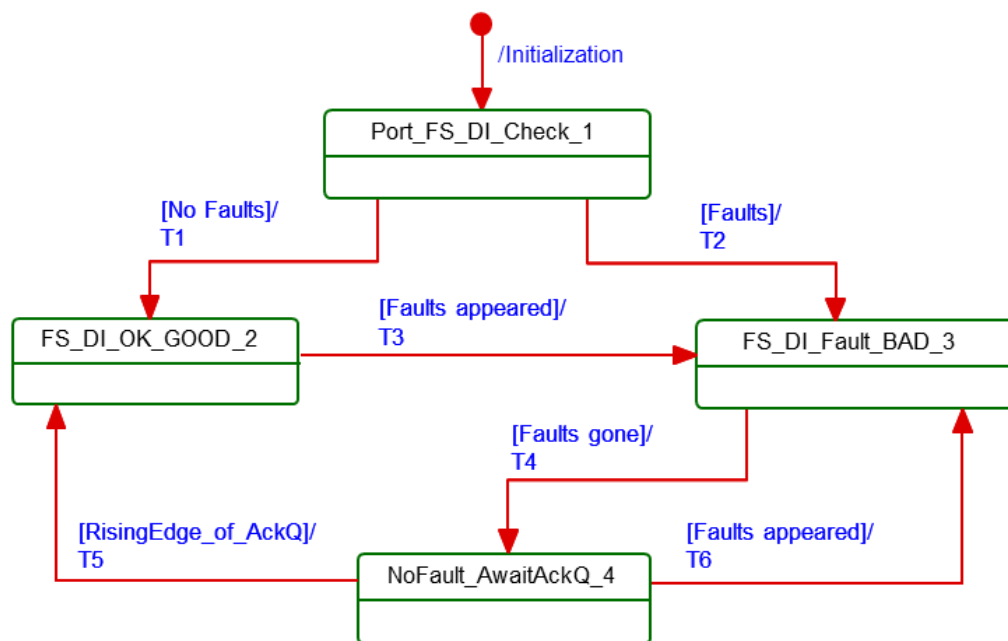


Figure 63 – Qualifier behavior per FS-Master Port

Table 44 shows the state and transition table for the qualifier behavior.

Table 44 – State transition table for the qualifier behavior

STATE NAME		STATE DESCRIPTION	
Initialization		Use SD, Qualifier = BAD, ChFAckReq =0	
1 Port_FS_DI_Check		Perform Port diagnosis to detect Faults	
2 FS_DI_OK_GOOD		Perform Port diagnosis cyclically to detect Faults	
3 FS_DI_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.
Faults		Flag	Any detected fault such as a wire break, short circuit.
ChFAckReq		Flag	Signal set by qualifier handler (see 11.12.2 and H.1)

11.13 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 64 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.

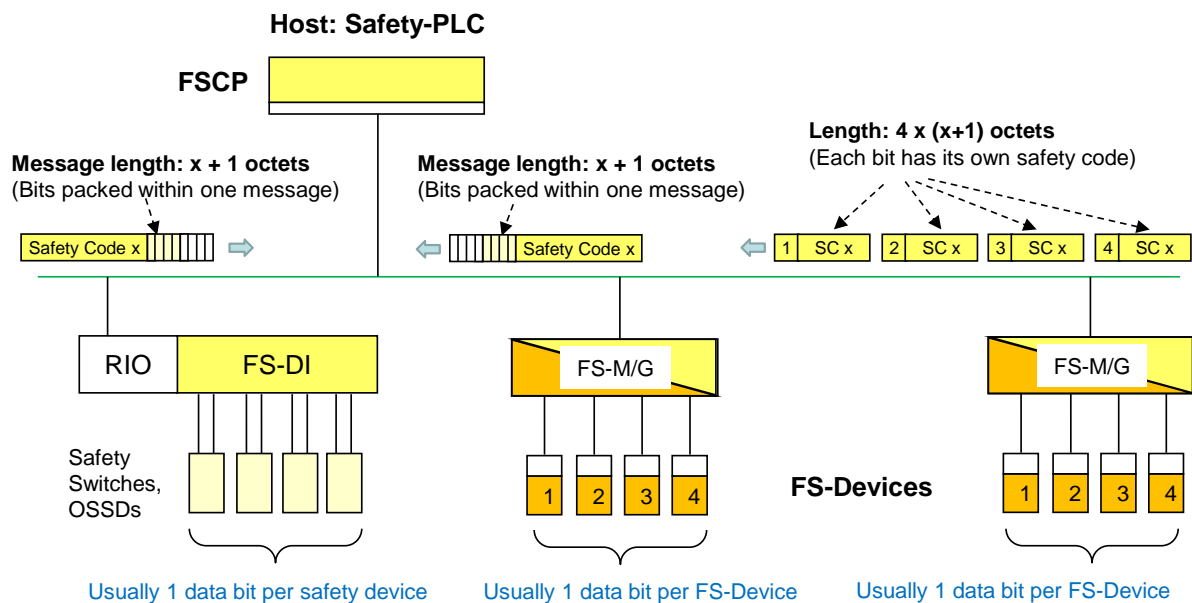


Figure 64 – Mapping efficiency issues

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.

Hint: the rules for parameter checks according to 11.3.4 in IEC 61131-9 apply.

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_AuthentCRC	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 65535 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

Index (dec)	Sub index	Object name	Access	Length	Data type	M/O/C	Purpose/reference
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
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							

2465

2466 A.2 Parameters in detail

2467 A.2.1 FSP_Authenticity

2468 During off-line commissioning of an SDCI-FS project, the default value of this parameter is "0".
 2469 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
 2470 in 10.4.3.1. The FS-Master Tool shall only transfer entire authenticity blocks to the FS-Device
 2471 with correct CRC signature values such that the FS-Device can check plausibility and
 2472 correctness (see A.2.3).
 2473

2474 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
 2475 FSP_VerifyRecord to detect any misconnection (incorrect Port or FS-Master), see Annex G.
 2476

2477 Some FSCPs provide extended authenticity. In those cases, the extended code shall be
 2478 included in this parameter.

2479 The parameters FSP_Authenticity_1 and FSP_Authenticity_2 can be different per port. These
 2480 parameters are to identify each FS-Master port in an application. Responsible for the correct
 2481 routing from the FSCP to the FS-Master port is the gateway application inside the FS-Master.
 2482 Representation of FSP_Authenticity1 and FSP_Authenticity2 in the engineering tool or FS-Master Tool in other number formats than in decimal (for example in hexadecimal) is allowed.
 2483

2484 Padding bits and octets shall be filled with "0".

2485 A.2.2 FSP_Port

2486 The FS-Master Tool identifies the FS-Master PortNumber of the attached FS-Device and stores
 2487 it in this parameter. Storage and checking of the parameter by the FS-Device corresponds to
 2488 A.2.1 and A.2.3. Numbering starts at "1". Thus, the FS-Device shall not accept a "0".

2489 Default PortNumber in IODD is "0" and means PortNumber of a particular Device has not been
 2490 assigned yet.

The port number is the physical port number of the FS-Master where the FS-Device is attached.

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	Reserved
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	Reserved
0xFB to 0xFF	Reserved

It is highly recommended that the CRC-32 is chosen (coding value 0x02) due to the enhancement of the IEC 61784-3.

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 worst case time between the reception of a safety PDU with a new MCount value and the provisioning of the response safety PDU with the appropriate DCount value.

With the help of the parameter default value (I/O update time), the transmission times of the safety PDUs, FS-Master processing times and IO-Link transmission 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).

2522 The safety-related interpreter of the FS-Master Tool transfers the entire instance data together
 2523 with the CRC signature to the FS I/O data mapper as shown in 10.4.3.1.

2524 Table A.4 shows Version "1" of the generic FS I/O data structure description for FS-Devices.
 2525 With the help of this table, individual instances of FS-Device I/O Process Data can be created
 2526 via IODD and, amongst others, used for an automatic mapping of SDCI-FS data to FSCP safety
 2527 data.

2528 **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 6 octets respectively for the safety code (Control/Status, PortNumber, and CRC-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 6 octets respectively for the safety code (Control/Status, PortNumber, and CRC-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)

2529

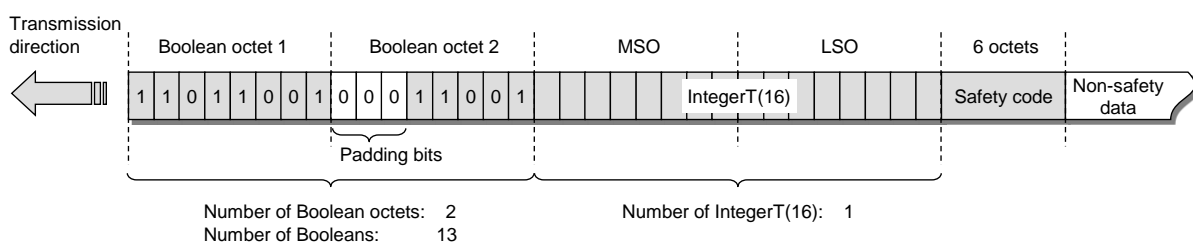
2530 Figure A.1 shows the instance of the FS I/O data description of the example in Figure A.2.

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

2531

2532 **Figure A.1 – Instance of an FS I/O data description**

2533 Figure A.2 shows an example with FS input Process Data and no FS output Process Data.



2534

2535 **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 11.8.8 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 11.8.9 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 \neq "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 11.8.4). The FS-Device uses this diversely handled record for verification of authenticity, protocol, I/O structure, and technology parameters. This takes place during PREOPERATE after a "DS_Change" (see Figure 31 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 [24], 11.2.14 and Clause E.9). In OPERATE the FSP_VerifyRecord is not allowed and be declined with "Service temporarily not available" (0x8022). 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. The checking of the FSP_VerifyRecord content is done by the SCL and could happen delayed to the transfer via the service.

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 6) 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 [7] or [11].

Values greater than 5 s leads to restricted FS-Master behavior, for example no automatic Device detection. The FSP_TimeToReady shall be set slightly larger than the FS-Device requires to overcome tolerances in clock speed. If FS-Device take longer than the FSP_TimeToReady (FS-Device is not ready) the FS-Master cannot start the port.

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). FSP_MinShutDownTime shall be specified longer than the FS-Device requires.

A.2.13 FSP_WCDT

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. The process

2586 data change only considers the SPDU data change in the device memory including device
 2587 internal synchronization. The SCL preparation and transmission times are considered in the
 2588 IOL-S protocol Watchdog.

2589 For an FS-Device (actuator) it is the time from signal change or capturing of an update of the
 2590 SPDU-Out to the actuator's safe state.

2591 The FS-Device designer measures/determines the "Worst-Case Delay Time" values. Several
 2592 different values are possible since FS-Devices can be configured via technology parameters,
 2593 for example high resolution may lead to longer and low resolution to shorter times. When
 2594 reading this parameter, the FS-Device will provide a value corresponding to its current
 2595 parameterization.

2596 **A.2.14 FSP_OFDT**

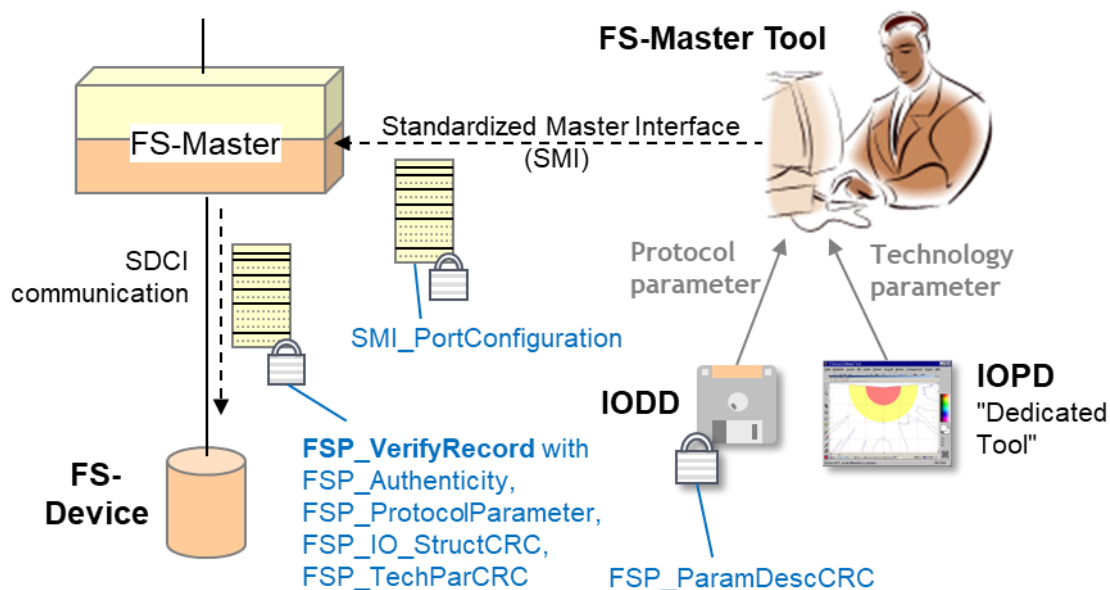
2597 The definition of OFDT is similar to WCDT, however in case of a fault within the FS-Device at
 2598 the time of triggering. Therefore, the value of OFDT is greater than or equal to the value of
 2599 WCDT.

2600 Faults to be considered in the OFDT are device specific and depend on the device architecture.

2601 The FS-Device designer measures/determines the "One Fault Delay Time" values. Several
 2602 different values are possible. When reading this parameter, the FS-Device will provide a value
 2603 corresponding to its current parameterization.

2604 **A.2.15 FSP_ParamDescCRC**

2605 The purpose of this parameter is to secure the relevant descriptions of safety parameters within
 2606 the IODD (see E.5.6) against data falsification as shown in Figure A.3.



2607
 2608 **Figure A.3 – Securing of safety parameters**

Annex B (normative)

Extensions to EventCodes

B.1 Additional FS-Device EventCodes

The FS-Device shall send events in case of error according to 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)	0	Notification
0xB001	Transmission error (Counter)	0	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 range (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	TYPE
0x2000	Transmission error (CRC signature or Counter if 0x2001 is not used)	Notification
0x2001	Transmission error (Counter) [optional]	Notification
0x2002	Transmission error (Timeout)	Error
0x2003	Unexpected authentication code	Error
0x2004	Unexpected authentication port	Error
0x2005	Incorrect FSP_AuthentCRC	Error
0x2006	Incorrect FSP_ProtParCRC	Error
0x2007	ISDU error (FSP_VerifyRecord)	Error
0x2008	Reserved	–
0x2009	Watchdog time out of specification (e.g. "0")	Error

EventCode	Definition and recommended maintenance action	TYPE
0x200A to 0x20EF	Reserved: do not use number; do not evaluate number	–
0x20F0 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 ($\approx \pm 2$ km, resolution 1 μ m)

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".

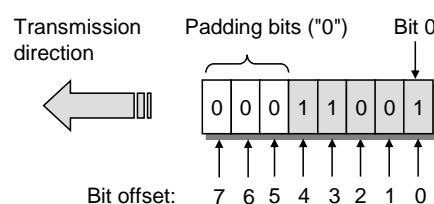


Figure C.1 – Example of a BooleanT data structure

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.

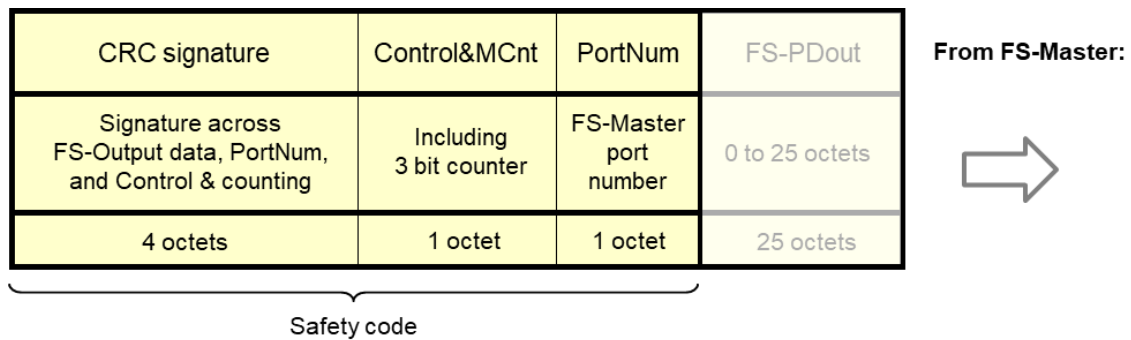


Figure C.2 – Safety Code of an output message

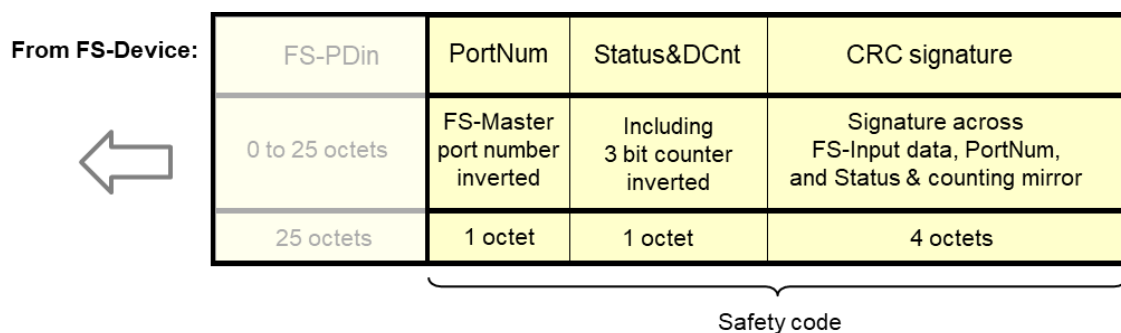


Figure C.3 – Safety Code of an input message

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 $g(x) = p(x) \cdot (1 + x)$ 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 [12]).

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	[13]	Suitable for functional safety
0xF4ACFB13	≤ 32768	≥ 6	≤128 octets	[13]	
	≤ 65534	≥ 4			
NOTE Representation: "Normal": high order bit omitted					

The CRC-16 can be used

- to secure the transfer of up to 16 octets of FSP parameters at start-up or restart.

The CRC-32 can be used

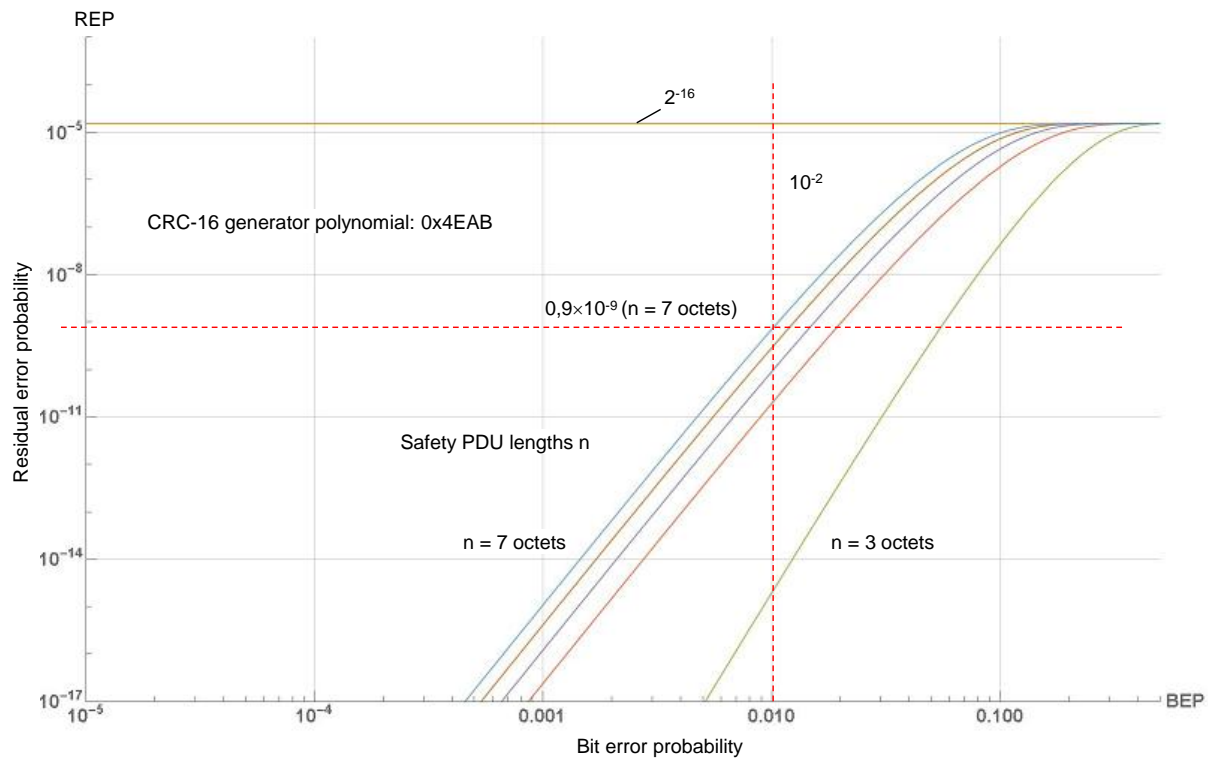
- to secure cyclic Process Data exchange with a total safety PDU length of up to 32 octets, i.e. 25 octets for safety Process Data (see 11.4.4), and 6 octets of safety code, and
- to secure the transfer and data integrity of the entire FST parameter set.

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 when using the CRC-16 generator polynomials is approx. 2×10^{-5} at a BEP of 0.5 for octet length from 3 to 7 which exceeds the required REP of 10^{-9} by far. Thus, the CRC16 polynomial cannot be used to protect SPDU transmission.



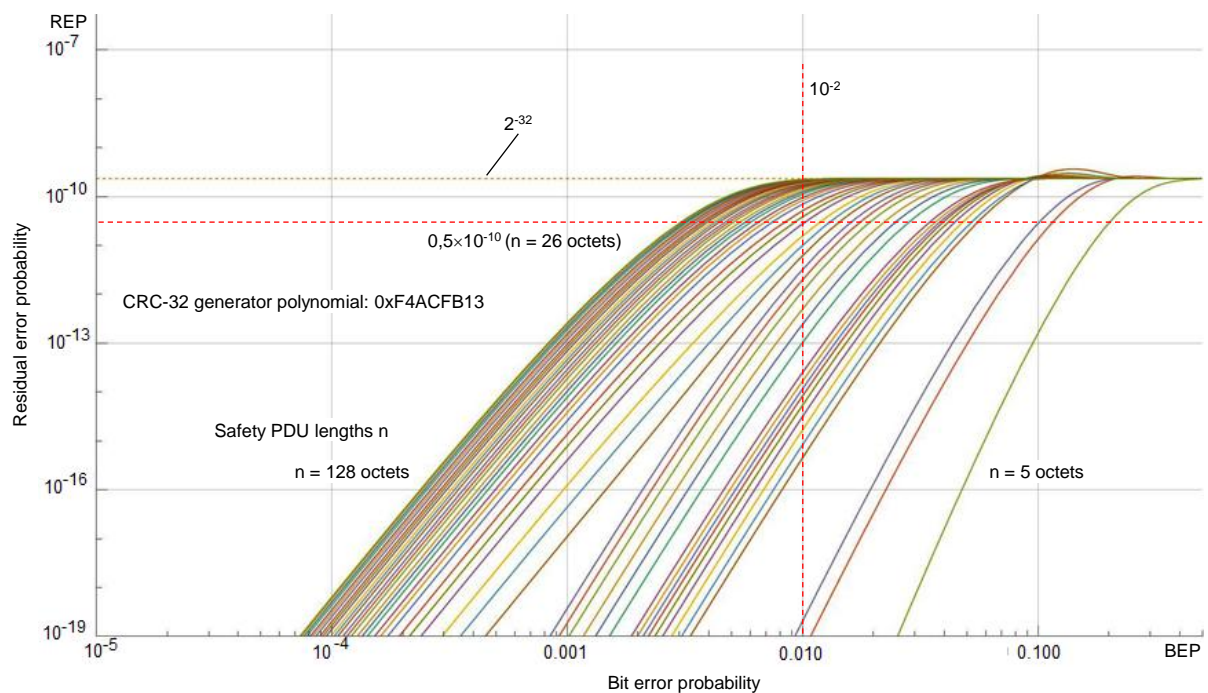
2715

2716

Figure D.1 – CRC-16 generator polynomial

2717 Figure D.2 shows the results of residual error probability (REP) calculations over bit error
 2718 probabilities (BEP) for safety PDU lengths from 5 to 128 octets.

2719 The REP is approx. 5×10^{-10} at a BEP of 0.5 for 26 octets which is less than the required REP
 2720 of 10^{-9} .



2721

2722

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 Octet
*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++.

2746 The function in Figure D.4 uses the lookup in Table D.4.

2747 **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).

2748

2749 **D.3.4 Bit shift algorithm (32 bit)**

2750 For the 32-bit CRC signature, the value 0xF4ACFB13 is used as the generator polynomial. The
 2751 number of data bits may be odd or even. The value generated after the last octet corresponds
 2752 to the CRC signature to be transferred.

2753 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 Octet
*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

CRC-32 lookup table (0 to 255)							
5C281C97	A884E784	41DD11A2	B571EAB1	67C206FD	936EFDEE	7A370BC8	8E9BF0DB
C45441EB	30F8BAF8	D9A14CDE	2D0DB7CD	FFBE5B81	0B12A092	E24B56B4	16E7ADA7
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	1A814BBB	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	EEEEF5C2D
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).							

2770

2771 D.3.6 Seed values

2772 The algorithm for example in Figure D.3 does not mention explicitly any initial value for the CRC
 2773 signature variable in "r". It is implicitly assumed to be "0" by default. This initial value is
 2774 sometimes called "seed value" in literature.

2775 In 11.4.7 a seed value of "1" is required for the cyclic data exchange of safety PDUs. The reason
 2776 for that is the possibility for the FS-PDout or FS-PDin data to become completely "0". Since it
 2777 is a property of CRC-signatures for leading zeros in data strings not to be secured by CRC
 2778 signatures whenever the seed value is "0", the requirement in 11.4.6 is justified. Any value
 2779 instead of "0" could be used. However, a "1" is sufficient and faster since all the operations then
 2780 are shifting and only the last one consists of shifting and XOR processing.

2781 In A.2.3, A.2.9, A.2.7, E.5.1 and E.5.6 , the seed value can be "0" since there are no leading
 2782 zeros within the data strings.

2783 Publicly available CRC signature calculators can be found in [14].

2784 **D.3.7 Octet order for CRC calculation**

2785 The order of octets for the CRC calculation of the SPDU shall start with the seed value followed
2786 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 11.8.1.

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 [15].

E.3 IODD constraints

E.3.1 General rules

Basis for the IODD in SDCI-FS are the definitions of the Common Profile in [16].

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 "IODD-Snippets-IOLS.xml". This XML file is included in the ZIP file of this specification.

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

Table E.1 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.1 – 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)
...				
Key H = highly 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 [15] 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 shall be “0”. The DefaultValues for FSP_TechParCRC and FSP_ProtParCRC shall be “Back-to-box” values. Table E.2 demonstrates the user actions to replace the default values by actual values.

Table E.2 – 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 11.8.8 and 11.8.9)
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.3 lists the RecordItems of FSP_Protocol to be serialized.

Table E.3 – 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.4 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.4 – Sample serialization for FSP_ParamDescCRC

Offset (hex)	Serialization (hex)	IODD items	Expected values (hex)
0000	42 00	index	42 00 (<i>≠ 0</i>)
0002	00 58	bitLength of index	00 58
0004	01	subindex	01 (<i>Authenticity 1</i>)
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
000C	02	subindex	02 (<i>Authenticity 2</i>)
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 (<i>Port</i>)
0015	00 10	bitOffset	00 10
0017	01	xsi:type=UIntegerT, bitLength=8	01
0018	00	RecordItemInfo/@defaultValue	00
0019	04	subindex	04 (<i>AuthentCRC</i>)
001A	00 00	bitOffset	00 00
001C	02	xsi:type=UIntegerT, bitLength=16	02
001D	00 00	RecordItemInfo/@defaultValue	00 00 (<i>dummy CRC</i>)
001F	42 01	index	42 01
0021	00 60	bitLength of index	00 60
0023	01	subindex	01 (<i>ProtVersion</i>)
0024	00 58	bitOffset	00 58
0026	01	xsi:type=UIntegerT, bitLength=8	01
0027	01	RecordItemInfo/@defaultValue	01
0028	01	SingleValue/@value	01 (<i>this document</i>)
0029	02	subindex	02 (<i>ProtMode</i>)
002A	00 50	bitOffset	00 50
002C	01	xsi:type=UIntegerT, bitLength=8	01
002D	02	RecordItemInfo/@defaultValue	(<i>Vendor defined</i>)
002E	02	SingleValue/@value	02 (<i>example 32 bit</i>)
002F	03	subindex	03 (<i>Watchdog</i>)
0030	00 40	bitOffset	00 40
0032	02	xsi:type=UIntegerT, bitLength=16	02
0033	00 64	RecordItemInfo/@defaultValue	(<i>Vendor defined</i>)
0035	00 64	ValueRange/@lowerValue	00 64 (<i>example: 100</i>)

Offset (hex)	Serialization (hex)	IODD items	Expected values (hex)
0037	13 88	ValueRange/@upperValue	13 88 (<i>example: 5000</i>)
0039	04	subindex	04 (<i>IO_StructCRC</i>)
003A	00 30	bitOffset	00 30
003C	02	xsi:type=UIntegerT, bitLength=16	02
003D	9A 28	RecordItemInfo/@defaultValue (see A.2.7)	(<i>Vendor defined</i>)
003F	05	subindex	05 (<i>TechParCRC</i>)
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 (<i>dummy CRC</i>)
0047	06	subindex	06 (<i>ProtParCRC</i>)
0048	00 00	bitOffset	00 00
004A	02	xsi:type=UIntegerT, bitLength=16	02
004B	00 00	RecordItemInfo/@defaultValue	00 00 (<i>dummy CRC</i>)
Calculated 32-bit FSP_ParamDescCRC signature value: 1860635738			See E.5.7

2907

2908 The sample serialization in Table E.4 shows 77 octets to be secured via the CRC-32 polynomial
 2909 listed in Table D.1. FS-Master Tool shall check the signature after import of the IODD. Only a
 2910 few values are variable and "*Vendor defined*" (see offsets: 002D, 002E, 0033 to 0037, and
 2911 003D). FS-Master Tool can compare the remaining values with preset values as an additional
 2912 safety measure.

2913 The "*dummy CRC*" are placeholders to be replaced by the FS-Master Tool once the user
 2914 assigned the actual parameter values.

2915 E.5.7 FST and FSP parameters and Data Storage

2916 FST parameters shall be described within the IODD. A "packed" parameter transfer via one
 2917 ISDU that is not described within the IODD is possible for Data Storage as long as the result in
 2918 the Device/FS-Device is the same as with discrete ISDUs (see 11.8.6). A manufacturer/vendor
 2919 is responsible to guarantee this behavior.

2920 FSP parameters (authenticity and protocol) shall be described within the IODD also and are
 2921 part of Data Storage.

2922 E.5.8 Sample IODD of an FS-Device

2923 A complete sample IODD file of an FS-Device with name "IO-Link-Safety_001201-20251023-
 2924 IODD1.1.xml" is included in the ZIP file of this specification or can be downloaded from the
 2925 website referenced in Annex I. This sample IODD file contains already calculated CRC
 2926 signature values. It refers to the Process Data example in Figure A.2.

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 provided by the Device manufacturer shall be used. Syntax and semantics of these files are standardized (see [15]) such that the Devices can be integrated independently from the vendor/manufacture.

However, functional safety requires a software tool for e.g. parameter setting and validation, at least as a complement to the IODD method.

This Device Tool – or Dedicated Tool shall communicate with its Device and is responsible for the data integrity according to IEC 62061. In the following, the term "Dedicated Tool" is used synonym to the term "Device Tool".

Without a standardized software interface between the Master Tool and the Dedicated Tool, the user would be forced:

- to know which Dedicated Tool is required for a particular Device,
- to enter the communication parameters of the Device both in the Master Tool and in the Dedicated Tool and to keep the parameters consistent,
- to store consistent configuration and parameterization data from both the Master Tool and the Dedicated Tool at one single place to archive project data.

In addition,

- in the case of an Online Channel, the Device manufacturer needs to implement the communication functionality for each supported communication system like a field bus system, including the problem of nested communication whenever the target Device is in a different network.

The solution for these requirements is the Device Tool Interface (DTI) technology specified herein after.

For Master Tools supporting IO-Link Safety, the complete implementation of the DTI including online channel is mandatory. For IO-Link Safety Devices and the associated Dedicated Tool the online channel is optional.

In the case that the IO-Link Safety Device supports parametrization with safety related technology parameters, the manufacturer of the IO-Link Safety Device shall provide a Dedicated Tool according to this specification.

F.2 DTI Overview

The Device Tool Interface (DTI) consists of two parts according to Figure 59:

- An Invocation Interface between Master Tool and Dedicated Tool including a backward interface between Master Tool and Dedicated Tool ("Backchannel");
- A Communication Interface for online communication between Dedicated Tool and a Device via a Master Tool based on a REST-API.

F.2.1 Functions and Elements of the Invocation Interface

- A Master Tool which is supposed to be already installed on a PC running a current Microsoft Windows operating system.

- A Device, configured with the help of the corresponding IODD file of the Device manufacturer. Configuration includes assignment of Port addresses and adjustment of the Device parameters as defined in the IODD.
- Association of Dedicated Tool identification with SDCI Device identification.
- DTI specific mechanisms allowing the Master Tool to find the matching Dedicated Tool for a given Device. These mechanisms may include for example, in the context menu of a selected Device in the Master Tool, an entry that can be used to invoke the Dedicated Tool.
- Identification of the selected Device by the Dedicated Tool on its activation.
- Establishing communication with the matching Device and assign parameter values without the need to provide address information and alike. Assigned values may be returned to the Master Tool using the back channel.
- Mechanisms to store and maintain Device data objects (project data).
- While the Dedicated Tool is launched, the Master Tool shall not allow to edit the FSP parameters and FST parameters of the device.

F.2.2 Functions and elements of the Communication Interface

- A Master Tool providing access to a related SDCI Device implementing the SMI interface.
- A Master Tool acting as a local HTTP server implementing a REST-API.
- A Dedicated Tool acting as HTTP client that connects to respective server.
- REST-API functionality that allows the Dedicated Tool to read and write SDCI parameters and to observe process data.

F.2.3 Security

As a Master Tool developer implementing this API, restrict the REST port so that it will only communicate with clients on localhost/127.0.0.1. This can be done through the selected webserver.

F.3 Dedicated Tool 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 Dedicated Tool. To achieve a high flexibility and to be able to identify different versions of the interface, both the description of the Dedicated Tool capabilities and the invocation parameters are exchanged in XML format.

Figure F.1 shows the principle of the DTI invocation interface part.

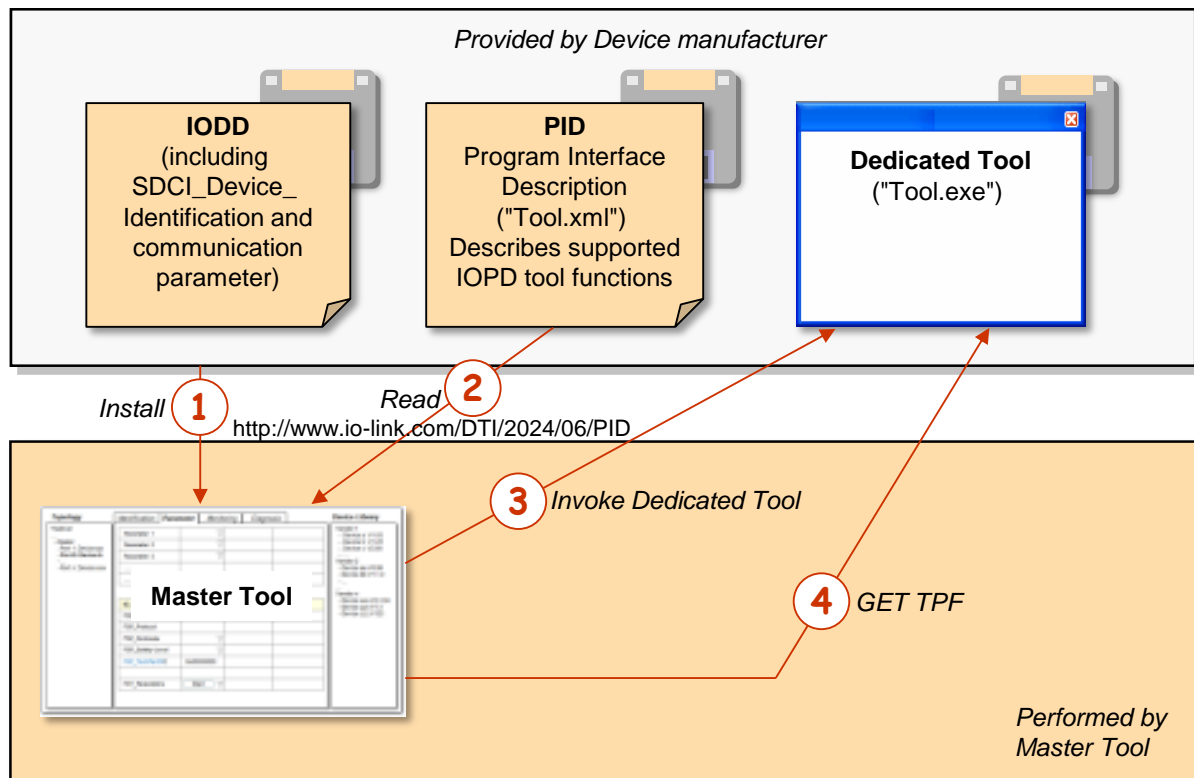


Figure F.1 – Principle of DTI invocation interface

The mechanism visualized in Figure F.1 requires that the Master Tool and all used Dedicated Tools exist on one PC.

For invocation of the Dedicated Tool the following steps are required:

- (1) The Master Tool imports the IODD. The Device is configured, and communication settings are made. With the help of SDCI Device Identification data, the Master Tool can address the installed Dedicated Tool and the "Program Interface Description" (PID). 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) The Master Tool launches the Dedicated Tool and passes configuration parameters for the REST-API as described in Annex F.5.
- (4) The Dedicated Tool issues a GET request to receive the TPF.

F.3.2 Detection of Dedicated Tool

F.3.2.1 Registry structure

DTI uses for identification of the type of an SDCI Device, a specific, unique, and unambiguous SDCI "Device-Identifier" 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.

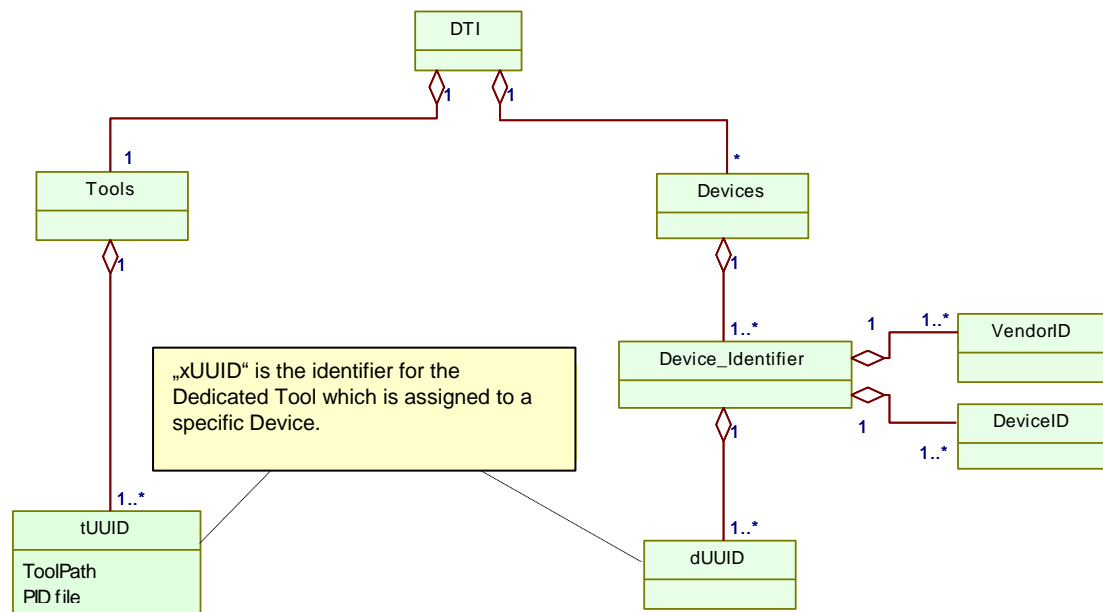


Figure F.2 – Structure of the registry

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 Dedicated Tool specific registry entries

Each version of a Dedicated Tool is represented by one UUID in the system registry.

The installation program of a Dedicated 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 Dedicated Tool file including its file extension (.exe).

Figure F.3 illustrates registry entries for Devices and tools.

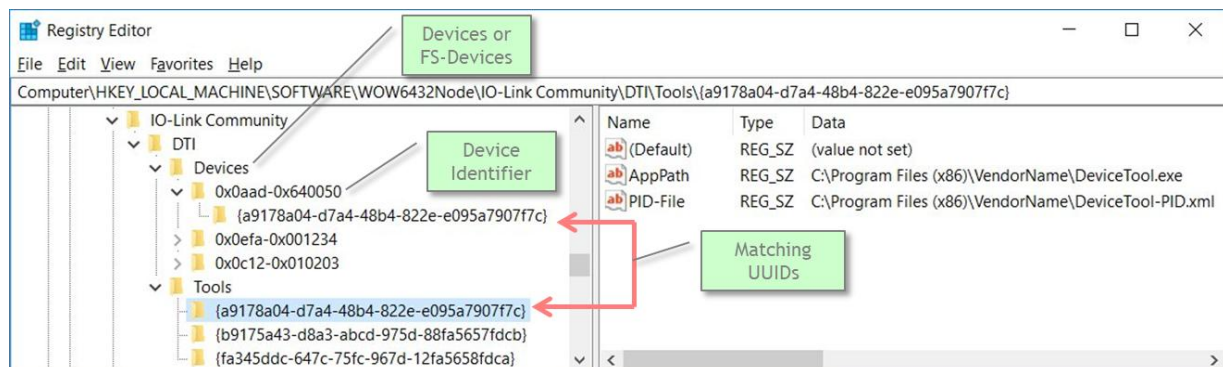


Figure F.3 – Example of a DTI registry

3044 If different versions of a Dedicated Tool for the same Device type exist (same Device Identifier),
 3045 each version requires a separate UUID in the registry. In the PID files of the Dedicated Tools ,
 3046 different version information shall be provided in the attribute "ToolDescription" of the element
 3047 "ToolDescription" (see Table F.1). This leads to multiple items in the context menu of the Master
 3048 Tool, differing in the description text.

3049 NOTE The advantage of a separate entry of the "ToolPath" keyword is a simpler installation procedure for the
 3050 Dedicated Tool. It can install the PID file without a need to modify this file.

3051 The installation program of a Dedicated Tool shall also insert each UUID as key under the
 3052 registry path

3053 `HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link Community\DTI\Devices\<Device Identifier>`

3054 Devices are identified unambiguously via the following items:

- 3055 • VendorID (assigned by SDCI support organization, see Annex I);
- 3056 • DeviceID (assigned by Device/FS-Device manufacturer).

3057 This information is part of the IO Device Description (IODD), which allows the Master Tool to
 3058 work with the Device (data, parameter) without establishing an online connection to the Device.
 3059 The IDs can be found at the following locations within an IODD:

3060 a) `//ISO15745Profile/ProfileBody/DeviceIdentity/@vendorId;`

3061 b) `//ISO15745Profile/ProfileBody/DeviceIdentity/@deviceId.`

3062 With the help of the registry, the Master Tool can read the required information about the
 3063 Dedicated Tool. Location and structure for the entries shall be commonly agreed upon.

3064 All entries shall be provided by the Dedicated Tool under the following registry path:

3065 `HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link Community\DTI\Devices`

3066 Within this path one or more keys can be inserted with the following field structure:

3067 `0xvvvv-0xddddd`

3068 The meaning of the fields is:

3069 `vvvv`: Four-character VendorID in hexadecimal coding

3070 `dddddd`: Six-character DeviceID in hexadecimal coding.

3071 The question mark character "?" can be used in the DeviceID as wildcard to replace one single
 3072 character. The number of question marks is only limited by the size of the field. If wildcards are
 3073 used, the Dedicated Tool is responsible for the check whether it supports the selected object.

3074 The assignment to the tool is made by a string value within this key. The UUID shall be used
 3075 as name for the string value. The number of string values is not limited, which in turn means an
 3076 unlimited number of tools that can be assigned to the same Device.

3077 Examples for valid keys (see Figure F.3):

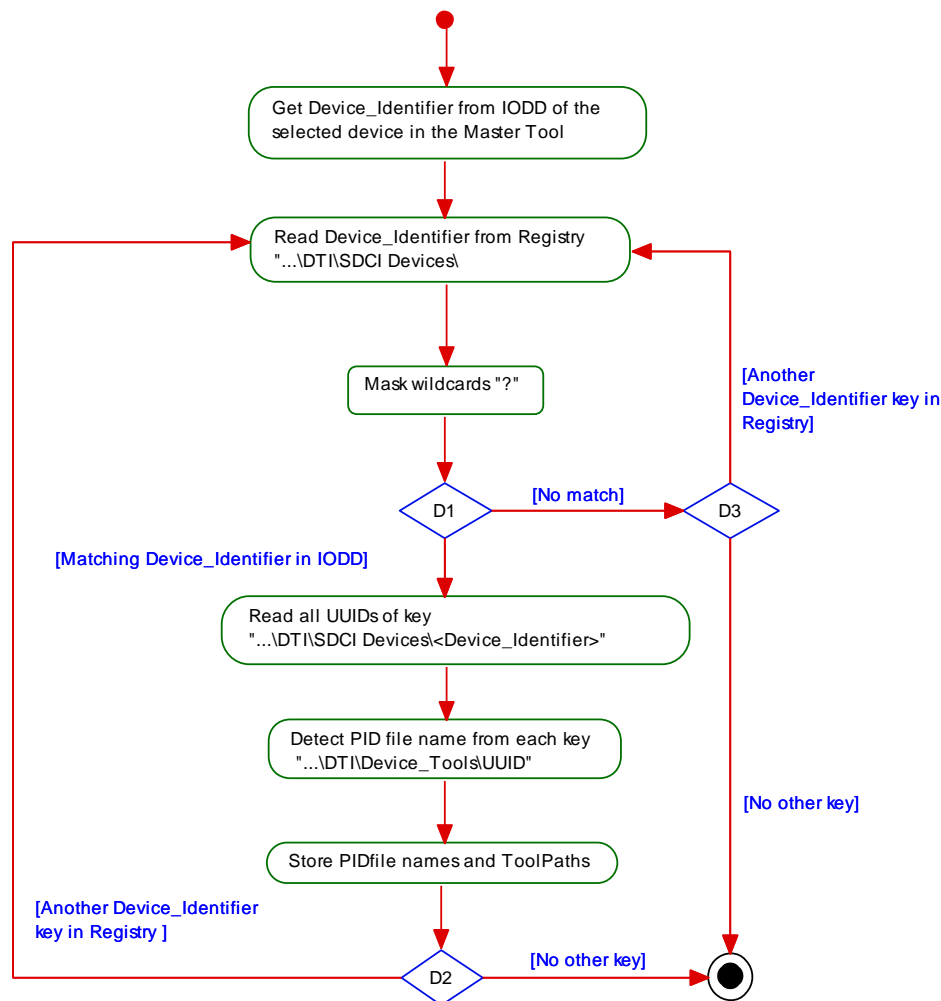
3078 `0x0A99-0x00880D` The tool can be launched in the context of a Device with a DeviceID
 3079 `0x00880D` from the vendor with the VendorID `0x0A99`.

3080 `0x0F3B-0x002B??` The tool can be started in the context of Devices with a DeviceID in the
 3081 range of `0x002B00` to `0x002BFF` from the vendor with the VendorID
 3082 `0x0F3B`.

3083 **F.3.2.3 Processing of the Registry Data**

3084 The installation program of the Dedicated Tool is responsible to insert the keys in the system
 3085 registry as defined in Annex F.3.2.2.

3086 Figure F.4 shows an activity diagram illustrating the detection of a Dedicated Tool in the registry
 3087 via SDCI "Device_Identifier".



3088
 3089 **Figure F.4 – Detection of a Dedicated Tool in registry**

3090 NOTE All registry keys in Figure F.4 are relative to the path HKEY_LOCAL_MACHINE\SOFTWARE\IO-Link
 3091 Community.

3092 In a first step, the Master Tool gets the Device_Identifier from the IODD of the selected object
 3093 in the Master Tool. Then all sub keys in the system registry path ...DTI\Devices shall be
 3094 compared with this Device_Identifier. If a sub key matches (excepting wildcards), the UUID sub
 3095 key of this key is used to find the PID file name in the registry path
 3096 DTI\Dedicated_Tools\<UUID>. Since the same PID file name can be found in different locations
 3097 in the registry, the context menu of the Master Tool shall only show the Dedicated Tools with
 3098 different PID file names. As a last step, the information in the PID file is used to build the menu
 3099 items of the Master Tool (Figure F.5).

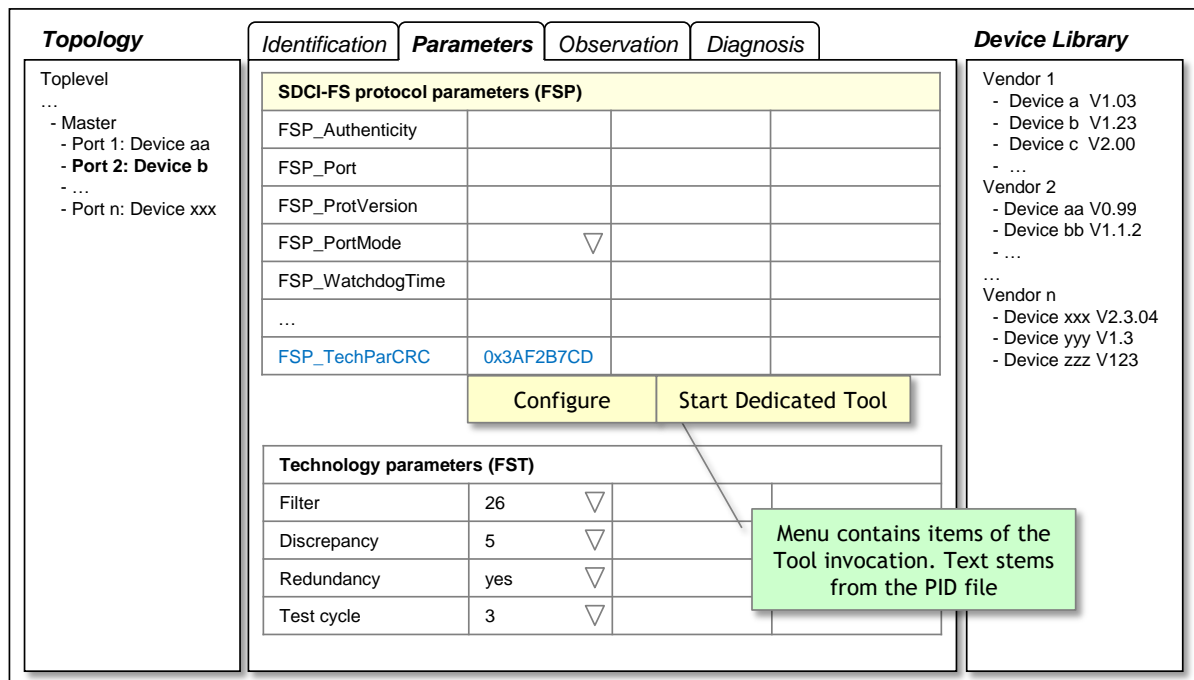
3100 F.3.3 Program Interface Description – PID

3101 F.3.3.1 General

3102 The Program Interface Description (PID) file describes the properties of the Dedicated Tool and
 3103 contains data which are required by the Master Tool to build menu items in its graphical user
 3104 interface (GUI). The PID file is an XML document. The corresponding XML schema is defined
 3105 in F.7.2. UTF-8 shall be used for character encoding.

3106 This PID file shall be provided by the manufacturer of a Device/Dedicated Tool and installed by
 3107 the installation program associated with the Dedicated Tool. This installation program shall also
 3108 insert the name and installation path in the system registry (see F.3.2).

3109 The PID file allows the Master Tool to extend its GUI menu structure by the name of the
 3110 Dedicated Tool such that the user is able to launch the Dedicated Tool for example from the
 3111 context menu of a selected Device as illustrated exemplary in Figure F.5.



3112
3113 **Figure F.5 – Menu for Dedicated Tool invocation**

3114 F.3.3.2 Structure of the PID file

3115 The PID file is an XML based document. Its XML schema is shown in Figure F.8. Namespace
 3116 URI for this file is <http://www.io-link.com/DTI/2024/06/PID>.

3117 The elements of Figure F.8 are specified in Table F.1. The column "SV" indicates the schema
 3118 version a particular attribute has been introduced.

3119 **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 Dedicated 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 Dedicated Tool. A ToolDescription element in English language is mandatory.

Element	Attribute	Type	M/O	SV	Description
	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 Dedicated Tool. This text shall be used to 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 Dedicated Tool.

F.3.3.3 Example PID file

The following XML code shows an example content of a PID file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProgramInterface xmlns="http://www.io-link.com/DTI/2024/06/PID"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
  link.com/DTI/2024/06/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2024/06/PID DTI-
  PID1.0.xsd">
  <DocumentInfo version="V1.0"/>
  <General vendorName="IO-LinkCompany">
    <ToolDescription name="Configure THC" description="IO-Link-16 Safety Device" lang="en"/>
    <ToolDescription name="Konfiguriere THC" description="IO-Link-16 Safety Device" lang="de"/>
  </General>
</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 Dedicated Tool, all required parameters are included into an XML format, called Temporary Parameter File (TPF).

The XML schema for the TPF is defined in F.7.3. 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 Dedicated Tool.

F.3.4.2 Structure of a TPF

The structure of the TPF is defined by the XML schema shown in Figure F.9. 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.

The Master Tool uses the TPF to transfer all readable parameters (read/write and read only) to the Dedicated Tool.

Namespace URI for this file is <http://www.io-link.com/DTI/2024/06/TPF>. The elements of the XML schema in Figure F.9 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 of the TPF/PID file are stored.

Element	Attribute	Type	M/O	SV	Description
					<ul style="list-style-type: none"> • 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 Dedicated Tool to initialize the XML parser. <p>NOTE Even if no schema validation is used, some XML parsers need the location of the schema files for initialization. In this case, a Dedicated 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.</p>
	projectRelatedPath	xsd:string	M	1.0	<p>The attribute "ProjectRelatedPath" contains information about a directory which is assigned to the project context of the Master Tool. A Dedicated Tool should use this path for storage of its Device data. The format and structure of this data is defined by the Dedicated Tool itself. Within this directory, additional subdirectories can be created.</p> <p>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.</p> <p>The attribute "ProjectRelatedPath" contains a unique path (directory) for each combination of Master project and DTI Dedicated 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.</p>

Element	Attribute	Type	M/O	SV	Description
					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 Dedicated Tool can use this name in error messages or user dialogs to provide more understandable texts.
	currentLanguage	xsd:string	M	1.0	Defines which language shall be used by the Dedicated Tool for TPF. The "2-letter coding" or the "3-letter coding" as defined in ISO 639 can be used. If a Dedicated Tool does not support the selected language, the tool shall use its default language.
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 Dedicated Tool 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
	ClientID	xsd:string	M	1.0	ClientID

Element	Attribute	Type	M/O	SV	Description
	DeviceInstanceIdent	xsd:string	M	1.0	uniquely identifies the one instance of a Device.
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

3152

3153 **F.3.4.3 Example of a TPF**

3154 The following XML code shows the content of an exemplary TPF file.

```

3155 <?xml version="1.0" encoding="UTF-8"?>
3156 <InvocationInterface xmlns="http://www.io-link.com/DTI/2024/06/TPF"
3157 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
3158 link.com/DTI/2024/06/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2024/06/TPF DTI-
3159 TPF1.0.xsd">
3160   <General currentLanguage="en" projectRelatedPath="//ServerName\ShareName\Projects"
3161   masterId="444444" masterName="CPU-1" portId="1" portName="P1-4" schemaPath="d:\dti\schem
3162   displayNameEs="MyMTName"/>
3163   <DeviceItem reference="Project1/Network2/Device3/1897212" vendorId="335" ClientID="1"
3164   deviceId="6553616" productId="SafetyDeviceVariant" usedConfigFile="d:\IODDfiles\IO-Link-
3165   SafetyDevice-20170225-IODD1.1.xml" usedConfigFileCRC="1946410459"
3166   DeviceInstanceIdent="DeviceInstant">
3167     <VariableInstanceData>
3168       <Variable variableId="V_DirectParameters_1">
3169         <Item subindex="0" state="valid" value=""/>
3170         <Item subindex="1" state="valid" value=""/>
3171         <Item subindex="2" state="valid" value=""/>
3172         <Item subindex="3" state="valid" value=""/>
3173         <Item subindex="4" state="valid" value=""/>
3174         <Item subindex="5" state="valid" value="17"/>
3175         <Item subindex="6" state="valid" value=""/>
3176         <Item subindex="7" state="valid" value=""/>
3177         <Item subindex="8" state="valid" value=""/>
3178         <Item subindex="9" state="valid" value=""/>
3179         <Item subindex="10" state="valid" value=""/>
3180         <Item subindex="11" state="valid" value=""/>
3181         <Item subindex="12" state="valid" value=""/>
3182         <Item subindex="13" state="valid" value=""/>

```

```
3183     <Item subindex="14" state="valid" value=""/>
3184     <Item subindex="15" state="valid" value=""/>
3185 </Variable>
3186 <Variable variableId="V_DeviceAccessLocks">
3187     <Item subindex="1" state="valid" value="false"/>
3188     <Item subindex="2" state="valid" value="false"/>
3189 </Variable>
3190 <Variable variableId="V_VendorName">
3191     <Item subindex="0" state="valid" value="IO-Link Community"/>
3192 </Variable>
3193 <Variable variableId="V_VendorText">
3194     <Item subindex="0" state="valid" value="http://www.io-link.com"/>
3195 </Variable>
3196 <Variable variableId="V_ProductName">
3197     <Item subindex="0" state="valid" value="SafetyDevice"/>
3198 </Variable>
3199 <Variable variableId="V_ProductID">
3200     <Item subindex="0" state="valid" value="SafetyDeviceVariant"/>
3201 </Variable>
3202 <Variable variableId="V_ProductText">
3203     <Item subindex="0" state="valid" value="Sample IO-Link Safety"/>
3204 </Variable>
3205 <Variable variableId="V_SerialNumber">
3206     <Item subindex="0" state="valid" value=""/>
3207 </Variable>
3208 <Variable variableId="V_HardwareRevision">
3209     <Item subindex="0" state="valid" value=""/>
3210 </Variable>
3211 <Variable variableId="V_FirmwareRevision">
3212     <Item subindex="0" state="valid" value=""/>
3213 </Variable>
3214 <Variable variableId="V_ApplicationSpecificTag">
3215     <Item subindex="0" state="valid" value="IO-Link Safety"/>
3216 </Variable>
3217 <Variable variableId="V_ErrorCount">
3218     <Item subindex="0" state="valid" value=""/>
3219 </Variable>
3220 <Variable variableId="V_DeviceStatus">
3221     <Item subindex="0" state="valid" value=""/>
3222 </Variable>
3223 <Variable variableId="V_DetailedDeviceStatus">
3224     <Item subindex="1" state="valid" value=""/>
3225     <Item subindex="2" state="valid" value=""/>
3226     <Item subindex="3" state="valid" value=""/>
3227     <Item subindex="4" state="valid" value=""/>
3228     <Item subindex="5" state="valid" value=""/>
3229     <Item subindex="6" state="valid" value=""/>
3230     <Item subindex="7" state="valid" value=""/>
3231     <Item subindex="8" state="valid" value=""/>
3232 </Variable>
3233 <Variable variableId="V_ProcessDataInput">
3234     <Item subindex="1" state="valid" value=""/>
3235     <Item subindex="2" state="valid" value=""/>
3236     <Item subindex="3" state="valid" value=""/>
3237     <Item subindex="4" state="valid" value=""/>
3238     <Item subindex="5" state="valid" value=""/>
3239     <Item subindex="6" state="valid" value=""/>
3240     <Item subindex="7" state="valid" value=""/>
3241     <Item subindex="8" state="valid" value=""/>
3242     <Item subindex="9" state="valid" value=""/>
3243     <Item subindex="10" state="valid" value=""/>
3244     <Item subindex="11" state="valid" value=""/>
3245     <Item subindex="12" state="valid" value=""/>
3246     <Item subindex="13" state="valid" value=""/>
3247     <Item subindex="14" state="valid" value=""/>
3248     <Item subindex="127" state="valid" value=""/>
3249     <Item subindex="128" state="valid" value=""/>
3250 </Variable>
3251 <Variable variableId="V_NonSafetyParameter">
3252     <Item subindex="0" state="valid" value="0"/>
3253 </Variable>
3254 <Variable variableId="V_FST_DiscrepancyTime">
3255     <Item subindex="0" state="valid" value="0"/>
3256 </Variable>
3257 <Variable variableId="V_FST_Filter">
3258     <Item subindex="0" state="valid" value="0"/>
3259 </Variable>
3260 <Variable variableId="V_FSP_Authenticity">
```

```

3261     <Item subindex="1" state="valid" value="0"/>
3262     <Item subindex="2" state="valid" value="0"/>
3263     <Item subindex="3" state="valid" value="0"/>
3264     <Item subindex="4" state="valid" value="0"/>
3265   </Variable>
3266   <Variable variableId="V_FSP_Protocol">
3267     <Item subindex="1" state="valid" value="0"/>
3268     <Item subindex="2" state="valid" value="1"/>
3269     <Item subindex="3" state="valid" value="100"/>
3270     <Item subindex="4" state="valid" value="444"/>
3271     <Item subindex="5" state="valid" value="0"/>
3272     <Item subindex="6" state="valid" value="0"/>
3273   </Variable>
3274   <Variable variableId="V_FSP_ParamDescCRC">
3275     <Item subindex="0" state="valid" value="444"/>
3276   </Variable>
3277 </VariableInstanceData>
3278 </DeviceItem>
3279 </InvocationInterface>
3280

```

3281 F.3.5 Temporary Backchannel File – TBF

3282 F.3.5.1 General

3283 The Dedicated Tool sends the TBF via a POST request to the Master Tool either automatically
 3284 or upon User request.

3285 F.3.5.2 Structure of the TBF

3286 The Dedicated Tool uses the TBF to transfer the results to the Master Tool. At least the
 3287 TechParCRC shall be included. In case read/write parameters are changed, they shall be
 3288 included in the TBF. Read only parameters shall not be included.

3289 The XML schema Figure F.10. This schema is built in a generic manner, which means, a new
 3290 parameter does not require the schema itself to be updated. Thus, new parameters can be
 3291 introduced without a new definition of the TBF structure.

3292 Namespace URI for this file is <http://www.io-link.com/DTI/2024/06/TBF>. The elements of Figure
 3293 F.10 are specified in Table F.3. The column "SV" indicates the schema version a particular
 3294 attribute has been introduced.

3295 **Table F.3 – Elements of the TBF**

Element	Attribute	Type	M/O	SV	Description
ReturnInterfaceRequest/ ReturnInterfaceResponse	–	–	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

F.3.5.3 Example of a TBF ReturnInterfaceRequest

The following XML shows the content of an exemplary TBF with a ReturnInterfaceRequest. The ReturnInterfaceRequest is returned from the Master Tool to the Dedicated Tool in the response to a TBF POST. <?xml version="1.0" encoding="UTF-8"?>.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReturnInterfaceRequest xmlns="http://www.io-link.com/DTI/2024/06/TBF"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
  link.com/DTI/2024/06/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2024/06/TBF DTI-
  TBF1.0.xsd">
  <VariableInstanceData>
    <Variable variableId="V_DeviceAccessLocks">
      <Item subindex="1" state="valid" value="false"/>
      <Item subindex="2" state="valid" value="false"/>
    </Variable>
    <Variable variableId="V_ApplicationSpecificTag">
      <Item subindex="0" state="valid" value="IO-Link Safety"/>
    </Variable>
    <Variable variableId="V_NonSafetyParameter">
      <Item subindex="0" state="valid" value="0"/>
    </Variable>
    <Variable variableId="V_FST_DiscrepancyTime">
      <Item subindex="0" state="valid" value="0"/>
    </Variable>
    <Variable variableId="V_FST_Filter">
      <Item subindex="0" state="valid" value="0"/>
    </Variable>
    <Variable variableId="V_FSP_Authenticity">
      <Item subindex="1" state="valid" value="0"/>
      <Item subindex="2" state="valid" value="0"/>
      <Item subindex="3" state="valid" value="0"/>
      <Item subindex="4" state="valid" value="0"/>
    </Variable>
    <Variable variableId="V_FSP_Protocol">
      <!-- (subindex 1) : FSP_ProtVersion -->
      <Item subindex="1" state="valid" value="0x01"/>
      <!-- (subindex 2) : FSP_ProtMode 0..25 octects-->
      <Item subindex="2" state="valid" value="0x02"/>
      <!-- (subindex 3) : FSP_Watchdog -->
      <Item subindex="3" state="valid" value="500"/>
      <!-- (subindex 4) : FSP_IO_Struct_CRC /placeholder value/-->
      <Item subindex="4" state="valid" value="0x87654321"/>
      <!-- (subindex 5) : FSP_TechParCRC /placeholder value/-->
      <Item subindex="5" state="valid" value="0x12345678"/>
      <!-- (subindex 6) : FSP_ProtParCRC /placeholder value/-->
      <Item subindex="6" state="valid" value="0x23456789"/>
    </Variable>
    <Variable variableId="V_FSP_ParamDescCRC">
      <!-- (subindex 0) : FSP_ParamDescCRC /placeholder value/-->
      <Item subindex="0" state="valid" value="0x87654321"/>
    </Variable>
  </VariableInstanceData>
</ReturnInterfaceRequest>
```

F.3.5.4 Example of a TBF ReturnInterfaceResponse

The following XML shows another content of an exemplary TBF. The ReturnInterfaceResponse is returned from the Master Tool to the Dedicated Tool in the response to a TBF POST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReturnInterfaceResponse xmlns="http://www.io-link.com/DTI/2024/06/TBF"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:prim="http://www.io-
  link.com/DTI/2024/06/Primitives" xsi:schemaLocation="http://www.io-link.com/DTI/2024/06/TBF DTI-
  TBF1.0.xsd">
  <Response value="true"/>
</ReturnInterfaceResponse>
```


F.3.6 Invocation behavior

F.3.6.1 Conventions on Dedicated Tool invocation

When the Dedicated Tool is invoked, it should pass named parameters like this:

```
--Port 80 --DeviceInstanceIdent 778
```

In the example --Port specifies the REST Service port number and --DeviceInstanceIdent specifies the unique the Device instance. If the DeviceInstanceIdent contains "blank" characters they should be enclosed in quotes (" ").

It is not required for the invoking Master Tool to monitor the status of the launched Dedicated Tools. Even in case an instance of a Dedicated Tool is already running, the Master Tool will generate a new Dedicated Tool invocation whenever the user launches the same tool again.

Therefore, it is the task of the Dedicated Tool to handle multiple invocations. Table F.4 lists invocation cases and possible behaviors.

Table F.4 – Invocation cases and behaviors

Case	Behavior
Dedicated Tool is launched once	No conflicts
Dedicated Tool is already running and works on the same Device instance as in a prior session.	<ul style="list-style-type: none"> – The tool should be brought to the foreground of the GUI desktop – Invocation of another instance of the Dedicated Tool shall be avoided
Dedicated Tool is already running and works on another Device instance as provided by the DTI call. The provided DeviceInstanceIdent is <i>known</i> in the Dedicated Tool.	<p>The behavior depends on the design of the Dedicated Tool:</p> <ul style="list-style-type: none"> – 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
Dedicated Tool is already running and works on another Device instance as provided by the DTI call. The provided DeviceInstanceIdent is <i>not known</i> in the Dedicated Tool.	<p>The behavior depends on the design of the Dedicated Tool:</p> <ul style="list-style-type: none"> – 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 Dedicated Tool is invoked via DTI, this tool should not call another Dedicated Tool because the Communication Interface cannot interconnect (no nested communication defined for a DTI Communication Interface).

F.3.6.2 Handling of the TPF

When the Dedicated Tool is invoked, it will use the REST service to GET the TPF (as described in the IO-Link-DTI-openapi.yaml file). The Dedicated Tool will handle any error codes from the Web server or when processing the TPF.

F.4 Communication Interface

F.4.1 General

As already explained in Clause F.1, there is no seamless communication solution for stand-alone Dedicated 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 Dedicated 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 Dedicated Tool either can work only with one well-defined communication interface or that the Dedicated 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 Dedicated Tools down to the associated Device (see Figure F.6).

The second major part of DTI solves two problems:

- All Devices/FS-Devices and their Dedicated Tools can rely on one well-defined communication interface.
- The chosen communication technology (REST-Service) solves the routing problem across network boundaries.

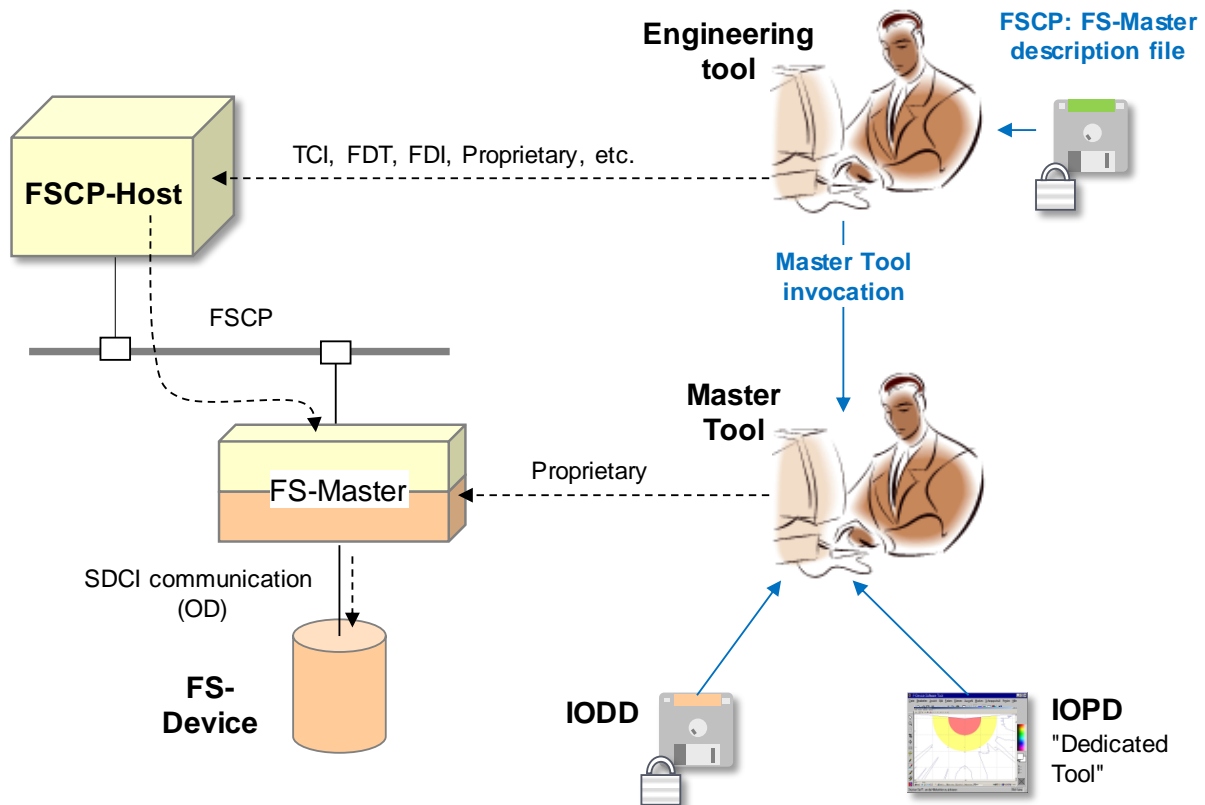


Figure F.6 – Communication routes between Dedicated Tool and Device

F.4.2 Principle of DTI communications

The communication interface consists of a REST service which provides a unique interface (API) to the Dedicated Tool. This REST service can provide communication functionality for different field busses and proprietary network protocols. The Master Tool passes the communication parameters (i.e. DeviceInstanceIdent and REST Service Port number) which are necessary to establish a connection to the Dedicated Tool when it is invoked.

Figure F.6 shows fieldbus or proprietary networks between the PC and the Device. Figure 59 shows the mapping to software and communication drivers for the field bus. This routing information is generated by the Engineering System and transferred.

When the Master Tool invokes the Dedicated tool, it passes a DeviceInstanceIdent as reference to the Dedicated Tool. This reference is forwarded from the Dedicated Tool to the Master Tool in every request to allow the Master Tool to address the corresponding IO-Link Safety Device.

Figure F.7 shows the relationships between the components involved.

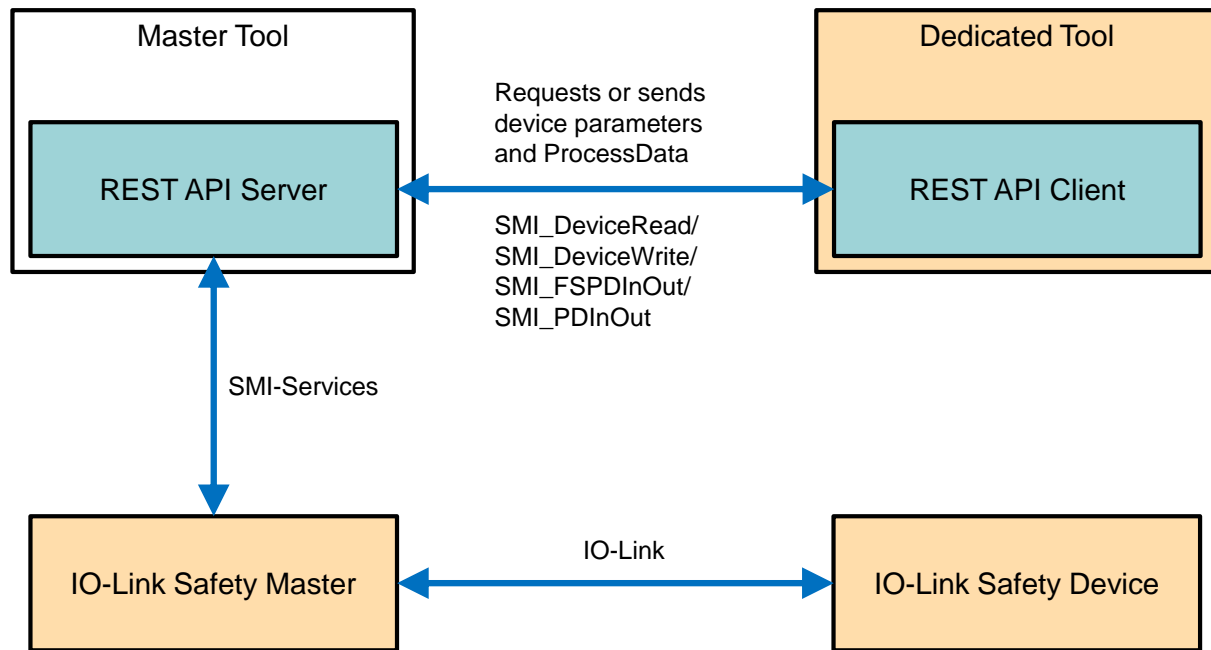


Figure F.7 – REST API Communication

It is always possible for a Dedicated Tool to use its native communication interfaces (for example serial RS232) as an alternative besides the Communication Interface.

F.4.3 Definition of the Communication Interface

The communication interface consists of a REST-API implemented by the Master Tool and described in the 'IO-Link-DTI-openapi.yaml' file (see <http://www.io-link.com/DTI/2024/06/IO-Link-DTI-openapi.yaml>).

F.4.1 Description of the Communication Interface

If a Dedicated Tool needs to read/write parameters to/from its Device, or read the Process Data, the Dedicated Tool can POST a Device Tool Query (DTQ) to the REST Service. The REST Service will respond with a Master Tool Response (MTR) as shown in more in F.7.

F.5 Reaction on incorrect tool behavior

Table F.5 describes the system reaction if a Master Tool or Dedicated Tool works incorrectly.

Table F.5 – Reaction on incorrect tool behavior

Fault	Description	System reaction
XML structure of PID file not valid	The PID file of a Dedicated Tool does not validate with the XML Schema in F.7.1	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.7.3	The Dedicated Tool should only show an error message if required schema elements or attributes are missing. All unknown elements or attributes shall be ignored.
Dedicated 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.
DeviceInstancIdent not valid	The Master Tool issues an incorrect DeviceInstancIdent	The Dedicated Tool should respond with an error message.

Fault	Description	System reaction
Timeout after invocation of the Dedicated Tool	The Dedicated Tool shall request the TPF within a maximum of 10 s.	In case of timeout the Master Tool will report a timeout error and terminate the connection to the Dedicated Tool.
Timeout on REST API requests for TPF, TBF and Process Data requests	The Master Tool shall respond within 2 s.	In case of timeout the Dedicated Tool will report a timeout error and terminate the connection to the Master Tool.
Timeout on REST API requests for ISDU	The Master Tool shall respond within 7s.	In case of timeout the Dedicated Tool will report a timeout error and terminate the connection to the Master Tool.
HTTP errors	Handling of HTTP errors / status	The standard HTTP status/error codes will be used by the REST Service.
SMI based errors	Negative SMI response	Issues in the IO-Link XML content will be handled by the error mechanism of the SMI Services.

F.6 Compatibility

F.6.1 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 Dedicated 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.7 are for information purposes only.

The installation program of the Dedicated Tool can always install the newest PID file version. The Master Tool shall ignore any unknown XML attributes or elements.

F.6.2 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 Dedicated Tool is only able to support the earlier version.

The version attribute in the DocumentInfo Element of the PID file identifies the latest xml version of the Dedicated Tool. See Annex F.3.3 for details.

If a Dedicated Tool supports a newer version than the Master Tool, the Master Tool uses its newest TPF version. In this case the Dedicated Tool shall work with the old schema version.

F.7 Schema definitions for PID, TPF and TBF

F.7.1 General

The schema definitions in this Annex F.7 are for information only (see Annex F.6.1).

F.7.2 Schema of the PID

Figure F.8 shows the XML schema of the Program Interface Description file.

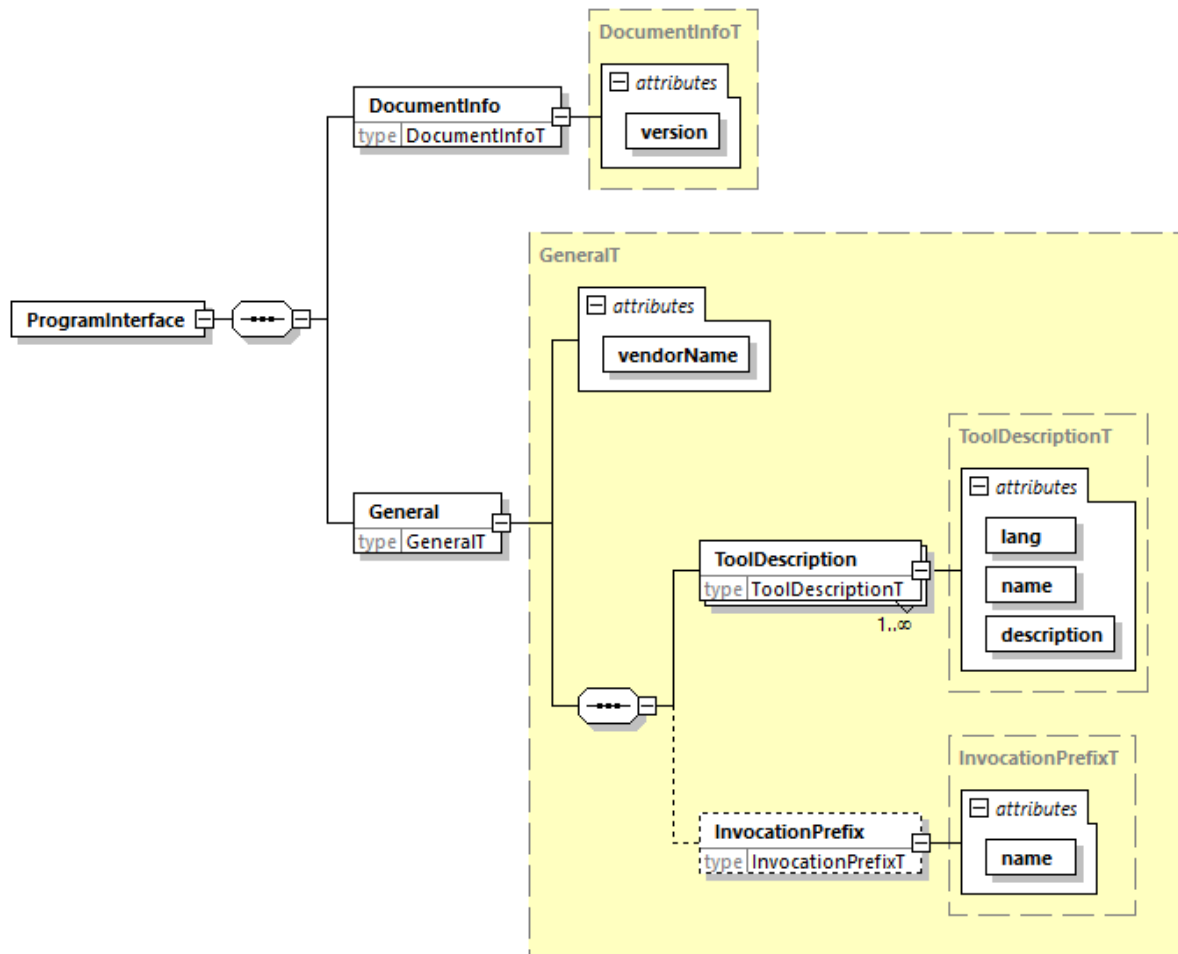


Figure F.8 – XML schema of the PID file

Figure F.8 is based on the following XML schema description code:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2024/06/PID" xmlns:prim="http://www.io-
link.com/DTI/2024/06/Primitives" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.io-link.com/DTI/2024/06/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/2024/06/Primitives" schemaLocation="DTI-
Primitives1.0.xsd"/>
  <xsd:complexType name="DocumentInfoT">
    <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:complexType name="ToolDescriptionT">
    <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:complexType name="GeneralT">
    <xsd:sequence>
      <xsd:element name="ToolDescription" type="ToolDescriptionT" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="vendorName" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:element name="ProgramInterface">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="DocumentInfo" type="DocumentInfoT"/>

```

```

3491         <xsd:element name="General" type="GeneralT"/>
3492     </xsd:sequence>
3493 </xsd:complexType>
3494 </xsd:element>
3495 </xsd:schema>

```

F.7.3 Schema of the TPF

Figure F.9 shows the XML schema of the Temporary Parameter File.

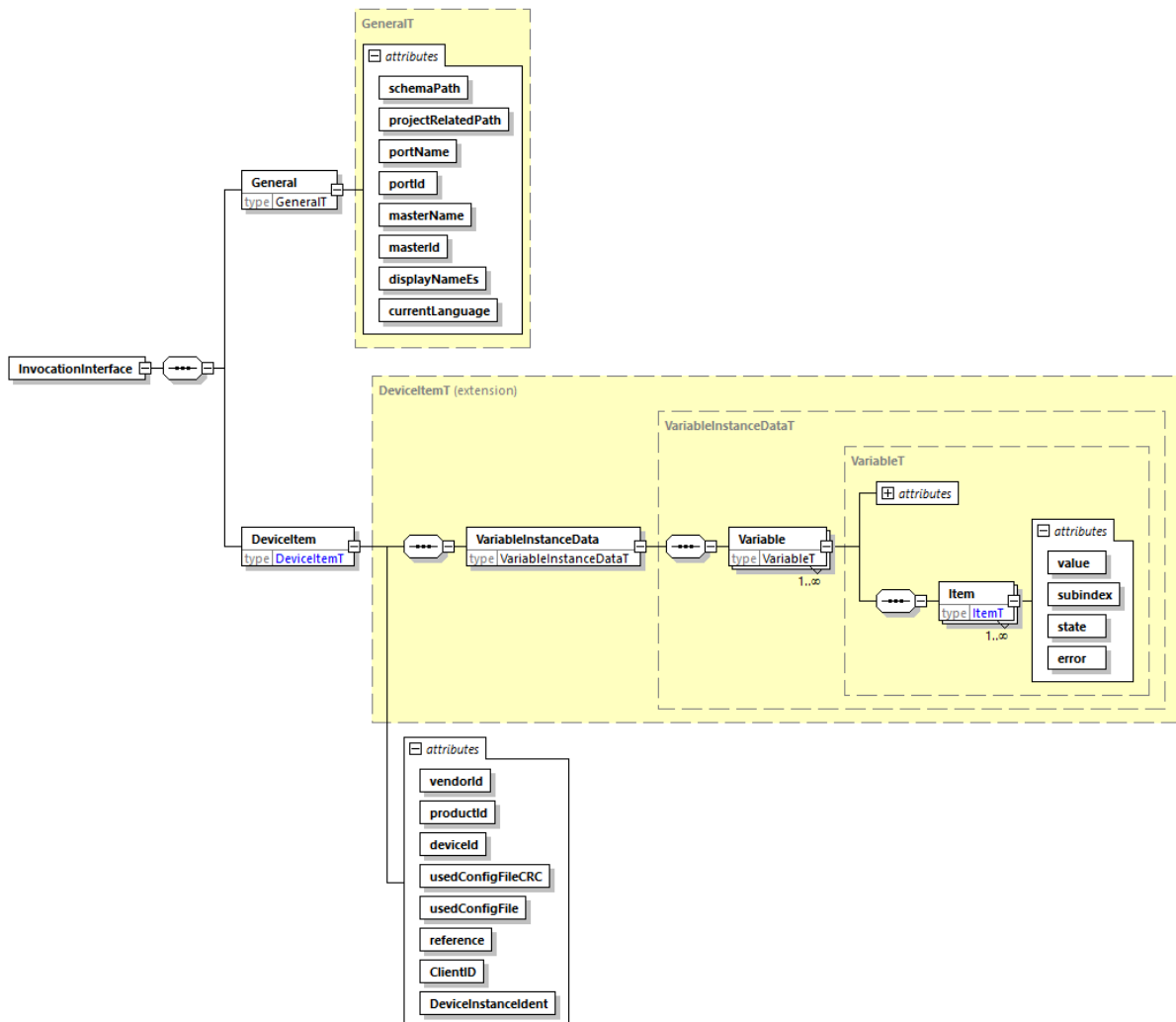


Figure F.9 – XML schema TPF

Figure F.9 is based on the following XML code:

```

3501 <?xml version="1.0" encoding="UTF-8"?>
3502 <xsd:schema xmlns="http://www.io-link.com/DTI/2024/06/TPF" xmlns:prim="http://www.io-
3503 link.com/DTI/2024/06/Primitives" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3504 targetNamespace="http://www.io-link.com/DTI/2024/06/TPF" elementFormDefault="qualified"
3505 attributeFormDefault="unqualified" version="1.0">
3506   <xsd:import namespace="http://www.io-link.com/DTI/2024/06/Primitives" schemaLocation="DTI-
3507 Primitives1.0.xsd"/>
3508   <xsd:complexType name="VariableInstanceDataT">
3509     <xsd:sequence>
3510       <xsd:element name="Variable" type="VariableT" maxOccurs="unbounded"/>
3511     </xsd:sequence>
3512   </xsd:complexType>
3513   <xsd:complexType name="VariableT">
3514     <xsd:sequence>
3515       <xsd:element name="Item" maxOccurs="unbounded">
3516         <xsd:complexType>
3517           <xsd:complexContent>
3518             <xsd:extension base="ItemT">
3519               <xsd:attribute name="value" type="xsd:string" use="required"/>

```

```
3520         <xsd:attribute name="subindex" type="xsd:unsignedByte" use="required"/>
3521         <xsd:attribute name="state" use="required">
3522             <xsd:simpleType>
3523                 <xsd:restriction base="xsd:string">
3524                     <xsd:enumeration value="invalid"/>
3525                     <xsd:enumeration value="valid"/>
3526                 </xsd:restriction>
3527             </xsd:simpleType>
3528         </xsd:attribute>
3529     </xsd:extension>
3530 </xsd:complexContent>
3531 </xsd:complexType>
3532 </xsd:element>
3533 </xsd:sequence>
3534 <xsd:attribute name="variableId" type="prim:IdT" use="required"/>
3535 </xsd:complexType>
3536 <xsd:complexType name="ItemT"/>
3537 <xsd:element name="InvocationInterface">
3538     <xsd:complexType>
3539         <xsd:sequence>
3540             <xsd:element name="General" type="GeneralT"/>
3541             <xsd:element name="DeviceItem"/>
3542         </xsd:complexType>
3543         <xsd:complexContent>
3544             <xsd:extension base="DeviceItemT">
3545                 <xsd:attribute name="vendorId" type="xsd:unsignedShort" use="required"/>
3546                 <xsd:attribute name="productId" type="xsd:string" use="required"/>
3547                 <xsd:attribute name="deviceId" use="required">
3548                     <xsd:simpleType>
3549                         <xsd:restriction base="xsd:unsignedInt">
3550                             <xsd:maxInclusive value="16777215"/>
3551                         </xsd:restriction>
3552                     </xsd:simpleType>
3553                 </xsd:attribute>
3554                 <xsd:attribute name="usedConfigFileCRC" type="xsd:int" use="required"/>
3555                 <xsd:attribute name="usedConfigFile" type="xsd:string" use="required"/>
3556                 <xsd:attribute name="reference" type="xsd:string" use="required"/>
3557                 <xsd:attribute name="ClientID" type="xsd:byte" use="required"/>
3558                 <xsd:attribute name="DeviceInstanceIdent" type="xsd:string" use="required"/>
3559             </xsd:extension>
3560         </xsd:complexContent>
3561     </xsd:complexType>
3562 </xsd:element>
3563 </xsd:sequence>
3564 </xsd:complexType>
3565 </xsd:element>
3566 <xsd:complexType name="GeneralT">
3567     <xsd:attribute name="schemaPath" type="xsd:string" use="required"/>
3568     <xsd:attribute name="projectRelatedPath" type="xsd:string" use="required"/>
3569     <xsd:attribute name="portName" type="xsd:string" use="required"/>
3570     <xsd:attribute name="portId" type="xsd:unsignedByte" use="required"/>
3571     <xsd:attribute name="masterName" type="xsd:string" use="required"/>
3572     <xsd:attribute name="masterId" type="xsd:int" use="required"/>
3573     <xsd:attribute name="displayNameEs" type="xsd:string" use="required"/>
3574     <xsd:attribute name="currentLanguage" type="xsd:string" use="required"/>
3575     <!-- IO-Link -->
3576 </xsd:complexType>
3577 <xsd:complexType name="DeviceItemT">
3578     <xsd:sequence>
3579         <xsd:element name="VariableInstanceData" type="VariableInstanceDataT"/>
3580     </xsd:sequence>
3581 </xsd:complexType>
3582 </xsd:schema>
```

F.7.4 Schema of the TBF

Figure F.10 shows the XML schema of the Temporary Backchannel File.

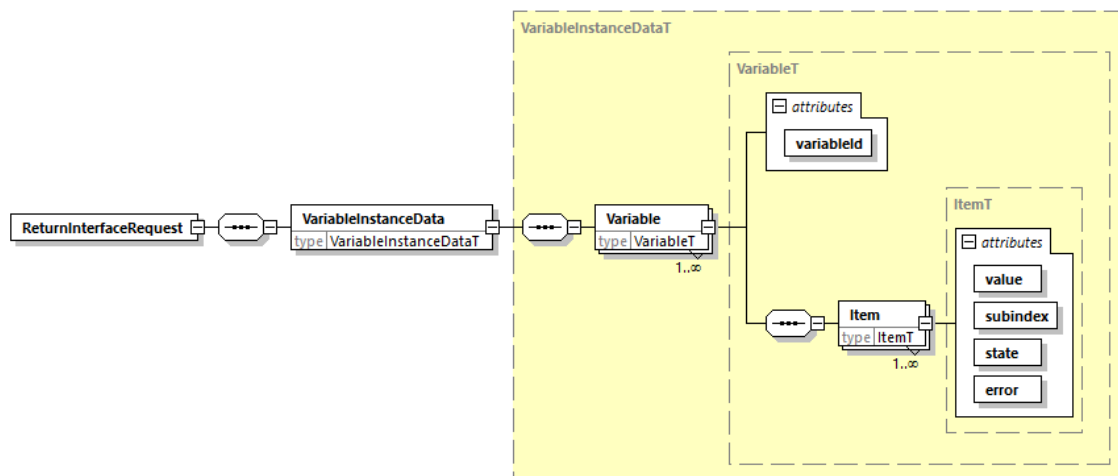


Figure F.10 – XML schema of a TBF

Figure F.10 is based on the following XML code:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2024/06/TBF" xmlns:prim="http://www.io-
link.com/DTI/2024/06/Primitives" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.io-link.com/DTI/2024/06/TBF" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xsd:import namespace="http://www.io-link.com/DTI/2024/06/Primitives" schemaLocation="DTI-
Primitives1.0.xsd"/>
  <xsd:complexType name="VariableInstanceDataT">
    <xsd:sequence>
      <xsd:element name="Variable" type="VariableT" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="VariableT">
    <xsd:sequence>
      <xsd:element name="Item" type="ItemT" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="variableId" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="ItemT">
    <xsd:attribute name="value" type="xsd:string" use="required"/>
    <xsd:attribute name="subindex" type="xsd:unsignedByte" use="required"/>
    <xsd:attribute name="state" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="valid"/>
          <xsd:enumeration value="invalid"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="ResponseT">
    <xsd:attribute name="value" type="xsd:boolean" use="required"/>
  </xsd:complexType>
  <xsd:element name="ReturnInterfaceRequest">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="VariableInstanceData" type="VariableInstanceDataT"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ReturnInterfaceResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Response" type="ResponseT"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```


F.7.5 Schema of DTI primitives

The DTI primitives used for PID, TPF and TBF are defined in the XML code as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.io-link.com/DTI/2024/06/Primitives"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.io-
link.com/DTI/2024/06/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="20240620"/>
    </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:schema>
```

F.8 Schema Definitions of the Communication Interface

The schema definitions of the Communication Interface provide a rule set to convert the 4 SMI Services DeviceWrite, DeviceRead, PDInOut (see IO-Link Interface and System specification, chapter 11) and FSPDInOut (see 10.3.8) into an XML notation.

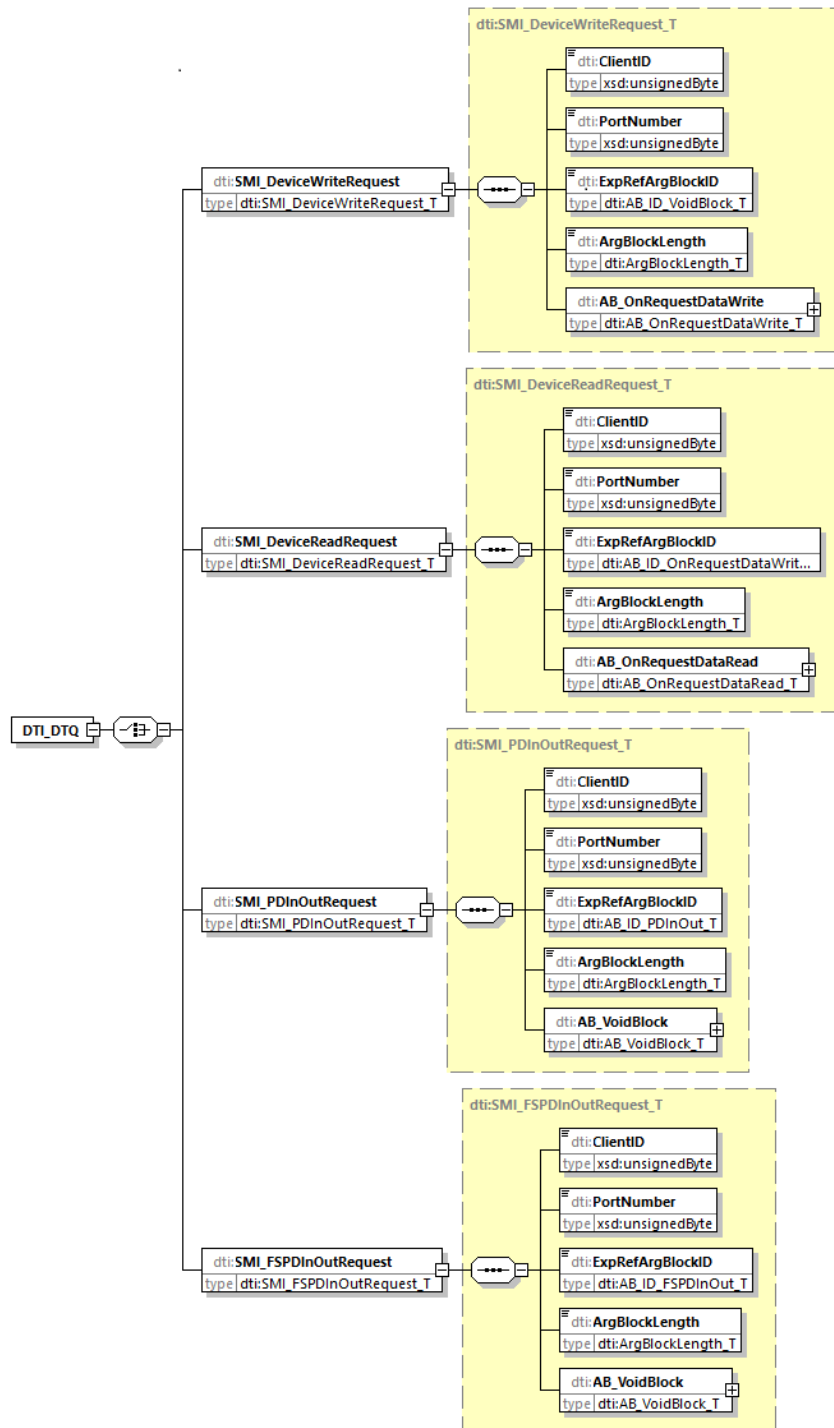


Figure F.11 – XML Schema DTI_DTQ

The Dedicated Tool issues an HTTP POST request to the Master Tool (see F.4) that contains a Device Tool Query element “DTI_DTQ” as shown in Figure F.11. The Master Tool responds with a Master Tool Response element “DTI_MTR” (see Figure F.12).

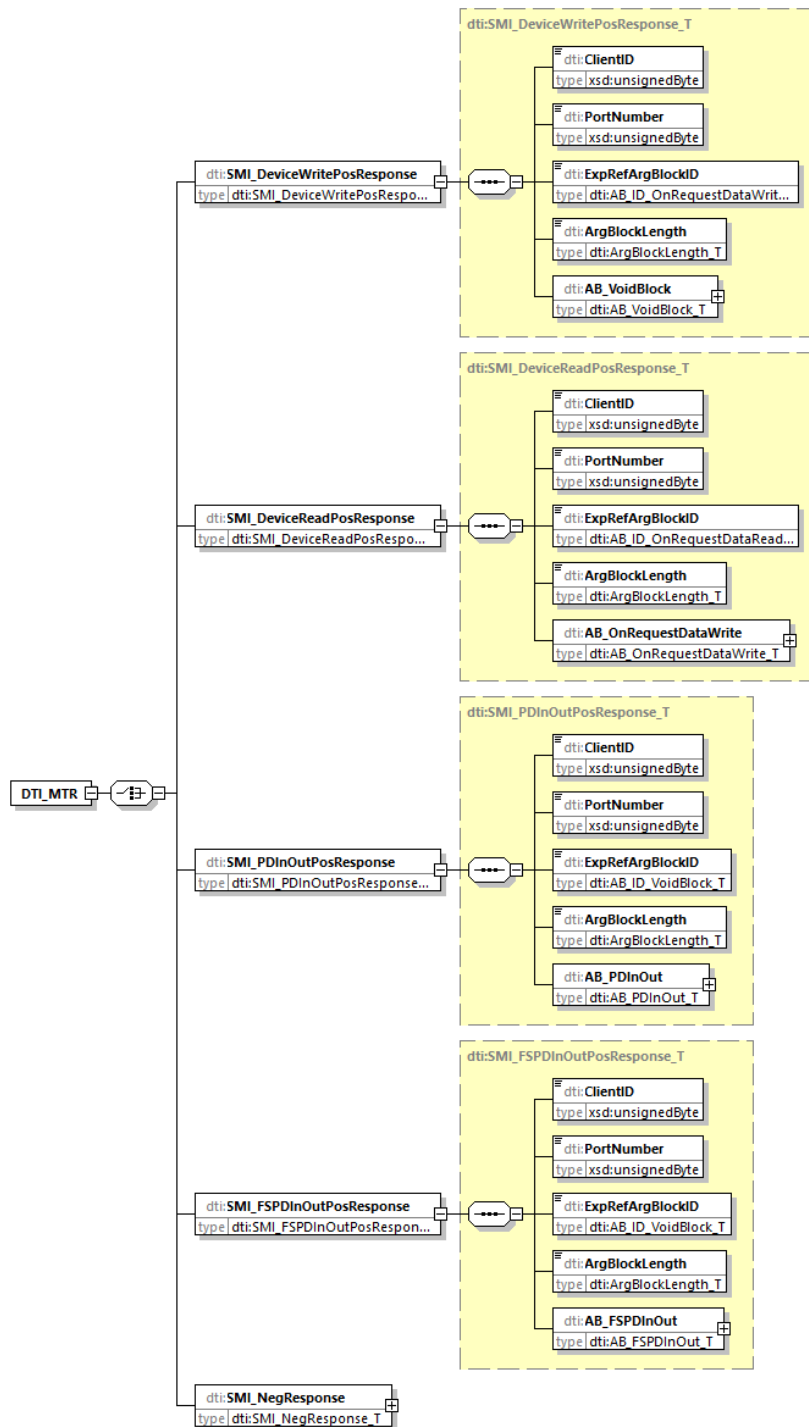


Figure F.12 – DTI_MTR response

Both the DTI_DTQ and the DTI_MTR elements contain the SMI service parameters ClientID, PortNumber, ExpRefArgBlockID, the ArgBlockLength and as last element the ArgBlock representation itself. In case that the Device Tool Query “DTI_DTQ” can’t be answered as expected the DTI_MTR element may contain a negative response “SMI_NegResponse” element (see Figure F.13).

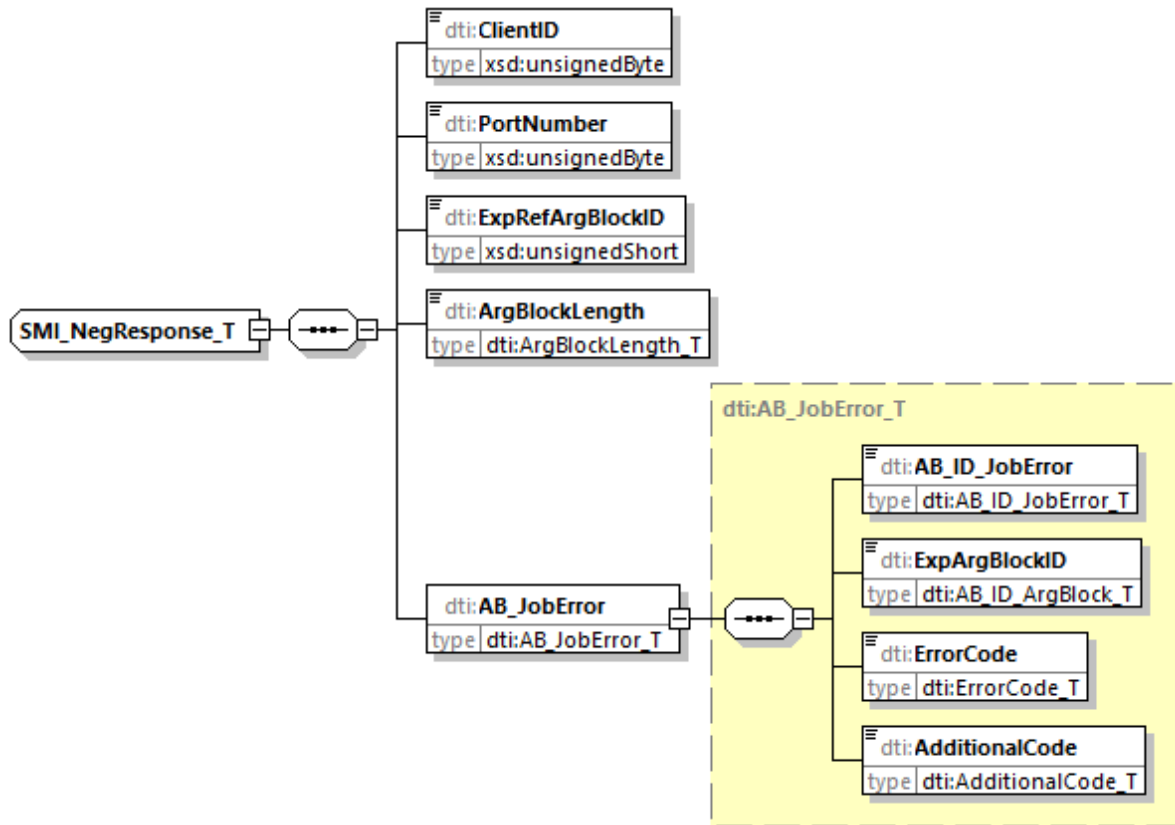


Figure F.13 – DTI_MTR negative response

Details on the XML coding of the DTI_DTQ respectively the DTI_MTR coding are given in the following XML schema definitions.

F.8.1 Schema for DTI_DTQ

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:dti="http://www.io-
link.com/DTI/2024/06/RestCommands" targetNamespace="http://www.io-
link.com/DTI/2024/06/RestCommands" elementFormDefault="qualified">
  <xsd:include schemaLocation="IOL_DTI_BaseTypes.xsd"/>
  <xsd:complexType name="SMI_DeviceWriteRequest_T">
    <xsd:sequence>
      <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
      <xsd:element name="ClientID" type="xsd:unsignedByte"/>
      <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
      <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_VoidBlock_T"/>
      <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
      <xsd:element name="AB_OnRequestDataWrite" type="dti:AB_OnRequestDataWrite_T"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="SMI_DeviceReadRequest_T">
    <xsd:sequence>
      <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
      <xsd:element name="ClientID" type="xsd:unsignedByte"/>
      <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
      <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_OnRequestDataWrite_T"/>
      <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
      <xsd:element name="AB_OnRequestDataRead" type="dti:AB_OnRequestDataRead_T"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="SMI_PDInOutRequest_T">
    <xsd:sequence>
      <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
      <xsd:element name="ClientID" type="xsd:unsignedByte"/>
      <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
      <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_PDInOut_T"/>
      <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
      <xsd:element name="AB_VoidBlock" type="dti:AB_VoidBlock_T"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

3751     </xsd:complexType>
3752     <xsd:complexType name="SMI_FSPDInOutRequest_T">
3753         <xsd:sequence>
3754             <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
3755             <xsd:element name="ClientID" type="xsd:unsignedByte"/>
3756             <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
3757             <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_FSPDInOut_T"/>
3758             <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
3759             <xsd:element name="AB_VoidBlock" type="dti:AB_VoidBlock_T"/>
3760         </xsd:sequence>
3761     </xsd:complexType>
3762     <xsd:element name="DTI_DTQ">
3763         <xsd:complexType>
3764             <xsd:choice>
3765                 <xsd:element name="SMI_DeviceWriteRequest" type="dti:SMI_DeviceWriteRequest_T"/>
3766                 <xsd:element name="SMI_DeviceReadRequest" type="dti:SMI_DeviceReadRequest_T"/>
3767                 <xsd:element name="SMI_PDInOutRequest" type="dti:SMI_PDInOutRequest_T"/>
3768                 <xsd:element name="SMI_FSPDInOutRequest" type="dti:SMI_FSPDInOutRequest_T"/>
3769             </xsd:choice>
3770         </xsd:complexType>
3771     </xsd:element>
3772 </xsd:schema>

```

3773 F.8.2 Schema for DTI_MTR

```

3774 <?xml version="1.0" encoding="UTF-8"?>
3775 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:dti="http://www.io-
3776 link.com/DTI/2024/06/RestCommands" targetNamespace="http://www.io-
3777 link.com/DTI/2024/06/RestCommands" elementFormDefault="qualified">
3778     <xsd:include schemaLocation="IOL_DTI_BaseTypes.xsd"/>
3779     <xsd:complexType name="SMI_DeviceWritePosResponse_T">
3780         <xsd:sequence>
3781             <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
3782             <xsd:element name="ClientID" type="xsd:unsignedByte"/>
3783             <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
3784             <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_OnRequestDataWrite_T"/>
3785             <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
3786             <xsd:element name="AB_VoidBlock" type="dti:AB_VoidBlock_T"/>
3787         </xsd:sequence>
3788     </xsd:complexType>
3789     <xsd:complexType name="SMI_DeviceReadPosResponse_T">
3790         <xsd:sequence>
3791             <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
3792             <xsd:element name="ClientID" type="xsd:unsignedByte"/>
3793             <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
3794             <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_OnRequestDataRead_T"/>
3795             <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
3796             <xsd:element name="AB_OnRequestDataWrite" type="dti:AB_OnRequestDataWrite_T"/>
3797         </xsd:sequence>
3798     </xsd:complexType>
3799     <xsd:complexType name="SMI_PDInOutPosResponse_T">
3800         <xsd:sequence>
3801             <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
3802             <xsd:element name="ClientID" type="xsd:unsignedByte"/>
3803             <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
3804             <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_VoidBlock_T"/>
3805             <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
3806             <xsd:element name="AB_PDInOut" type="dti:AB_PDInOut_T"/>
3807         </xsd:sequence>
3808     </xsd:complexType>
3809     <xsd:complexType name="SMI_FSPDInOutPosResponse_T">
3810         <xsd:sequence>
3811             <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
3812             <xsd:element name="ClientID" type="xsd:unsignedByte"/>
3813             <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
3814             <xsd:element name="ExpRefArgBlockID" type="dti:AB_ID_VoidBlock_T"/>
3815             <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
3816             <xsd:element name="AB_FSPDInOut" type="dti:AB_FSPDInOut_T"/>
3817         </xsd:sequence>
3818     </xsd:complexType>
3819     <xsd:complexType name="SMI_NegResponse_T">
3820         <xsd:sequence>
3821             <!-- Elements ClientID, PortNumber, and ArgBlockLength not checked by schema validator -->
3822             <xsd:element name="ClientID" type="xsd:unsignedByte"/>
3823             <xsd:element name="PortNumber" type="xsd:unsignedByte"/>
3824             <xsd:element name="ExpRefArgBlockID" type="xsd:unsignedShort"/>
3825             <xsd:element name="ArgBlockLength" type="dti:ArgBlockLength_T"/>
3826             <xsd:element name="AB_JobError" type="dti:AB_JobError_T"/>

```

```

3827     </xsd:sequence>
3828 </xsd:complexType>
3829 <xsd:element name="DTI_MTR">
3830   <xsd:complexType>
3831     <xsd:choice>
3832       <xsd:element name="SMI_DeviceWritePosResponse" type="dti:SMI_DeviceWritePosResponse_T"/>
3833       <xsd:element name="SMI_DeviceReadPosResponse" type="dti:SMI_DeviceReadPosResponse_T"/>
3834       <xsd:element name="SMI_PDInOutPosResponse" type="dti:SMI_PDInOutPosResponse_T"/>
3835       <xsd:element name="SMI_FSPDInOutPosResponse" type="dti:SMI_FSPDInOutPosResponse_T"/>
3836       <xsd:element name="SMI_NegResponse" type="dti:SMI_NegResponse_T"/>
3837     </xsd:choice>
3838   </xsd:complexType>
3839 </xsd:element>
3840 </xsd:schema>

```

F.8.3 Schema for DTI Online Channel Base Types

```

3841
3842 <?xml version="1.0" encoding="UTF-8"?>
3843 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:dti="http://www.io-
3844 link.com/DTI/2024/06/RestCommands" targetNamespace="http://www.io-
3845 link.com/DTI/2024/06/RestCommands" elementFormDefault="qualified">
3846   <!-- General types -->
3847   <xsd:simpleType name="ArgBlockLength_T">
3848     <xsd:restriction base="xsd:unsignedShort">
3849       <xsd:minInclusive value="2"/>
3850     </xsd:restriction>
3851   </xsd:simpleType>
3852   <xsd:simpleType name="PDInOutLength_T">
3853     <xsd:restriction base="xsd:unsignedByte">
3854       <xsd:minInclusive value="0"/>
3855       <xsd:maxInclusive value="32"/>
3856     </xsd:restriction>
3857   </xsd:simpleType>
3858   <xsd:simpleType name="FSPDInOutLength_T">
3859     <xsd:restriction base="xsd:unsignedByte">
3860       <xsd:minInclusive value="0"/>
3861       <xsd:maxInclusive value="32"/>
3862     </xsd:restriction>
3863   </xsd:simpleType>
3864   <xsd:complexType name="OctetString_T">
3865     <xsd:sequence minOccurs="0" maxOccurs="unbounded">
3866       <xsd:element name="Octet" type="xsd:unsignedByte"/>
3867     </xsd:sequence>
3868   </xsd:complexType>
3869   <xsd:simpleType name="ErrorCode_T">
3870     <xsd:restriction base="xsd:unsignedByte">
3871       <xsd:enumeration value="64"/>
3872       <xsd:enumeration value="128"/>
3873     </xsd:restriction>
3874   </xsd:simpleType>
3875   <xsd:simpleType name="AdditionalCode_T">
3876     <xsd:restriction base="xsd:unsignedByte"/>
3877   </xsd:simpleType>
3878   <!-- ArgBlock ID types and their restrictions -->
3879   <xsd:simpleType name="AB_ID_ArgBlock_T">
3880     <xsd:restriction base="xsd:unsignedShort">
3881       <xsd:minInclusive value="1"/>
3882     </xsd:restriction>
3883   </xsd:simpleType>
3884   <xsd:simpleType name="AB_ID_VoidBlock_T">
3885     <xsd:restriction base="dti:AB_ID_ArgBlock_T">
3886       <xsd:enumeration value="65520"/>
3887       <!-- 0xFFFF0 -->
3888     </xsd:restriction>
3889   </xsd:simpleType>
3890   <xsd:simpleType name="AB_ID_OnRequestDataWrite_T">
3891     <xsd:restriction base="dti:AB_ID_ArgBlock_T">
3892       <xsd:enumeration value="12288"/>
3893       <!-- 0x3000 -->
3894     </xsd:restriction>
3895   </xsd:simpleType>
3896   <xsd:simpleType name="AB_ID_OnRequestDataRead_T">
3897     <xsd:restriction base="dti:AB_ID_ArgBlock_T">
3898       <xsd:enumeration value="12289"/>
3899       <!-- 0x3001 -->
3900     </xsd:restriction>
3901   </xsd:simpleType>
3902   <xsd:simpleType name="AB_ID_PDInOut_T">

```

```

3903     <xsd:restriction base="dti:AB_ID_ArgBlock_T">
3904       <xsd:enumeration value="4099"/>
3905       <!-- 0x1003 -->
3906     </xsd:restriction>
3907   </xsd:simpleType>
3908   <xsd:simpleType name="AB_ID_FSPDInOut_T">
3909     <xsd:restriction base="dti:AB_ID_ArgBlock_T">
3910       <xsd:enumeration value="4355"/>
3911       <!-- 0x1103 -->
3912     </xsd:restriction>
3913   </xsd:simpleType>
3914   <xsd:simpleType name="AB_ID_JobError_T">
3915     <xsd:restriction base="dti:AB_ID_ArgBlock_T">
3916       <xsd:enumeration value="65535"/>
3917       <!-- 0xFFFF -->
3918     </xsd:restriction>
3919   </xsd:simpleType>
3920   <!-- ArgBlock types -->
3921   <xsd:complexType name="AB_VoidBlock_T">
3922     <xsd:sequence>
3923       <xsd:element name="AB_VoidBlockID" type="dti:AB_ID_VoidBlock_T"/>
3924     </xsd:sequence>
3925   </xsd:complexType>
3926   <xsd:complexType name="AB_OnRequestDataWrite_T">
3927     <xsd:sequence>
3928       <xsd:element name="AB_ID_OnRequestDataWrite" type="dti:AB_ID_OnRequestDataWrite_T"/>
3929       <xsd:element name="Index" type="xsd:unsignedShort"/>
3930       <xsd:element name="Subindex" type="xsd:unsignedByte"/>
3931       <xsd:element name="OnRequestData" type="dti:OctetString_T"/>
3932     </xsd:sequence>
3933   </xsd:complexType>
3934   <xsd:complexType name="AB_OnRequestDataRead_T">
3935     <xsd:sequence>
3936       <xsd:element name="AB_ID_OnRequestDataRead" type="dti:AB_ID_OnRequestDataRead_T"/>
3937       <xsd:element name="Index" type="xsd:unsignedShort"/>
3938       <xsd:element name="Subindex" type="xsd:unsignedByte"/>
3939       <!-- ArgBlock 0x3001 is Index only -->
3940     </xsd:sequence>
3941   </xsd:complexType>
3942   <xsd:complexType name="AB_PDInOut_T">
3943     <xsd:sequence>
3944       <xsd:element name="AB_ID_PDInOut" type="dti:AB_ID_PDInOut_T"/>
3945       <xsd:element name="PQI" type="xsd:unsignedByte"/>
3946       <xsd:element name="OE" type="xsd:unsignedByte"/>
3947       <xsd:element name="InputDataLength" type="dti:PDInOutLength_T"/>
3948       <xsd:element name="PDI" type="dti:OctetString_T"/>
3949       <xsd:element name="OutputDataLength" type="dti:PDInOutLength_T"/>
3950       <xsd:element name="PDO" type="dti:OctetString_T"/>
3951     </xsd:sequence>
3952   </xsd:complexType>
3953   <xsd:complexType name="AB_FSPDInOut_T">
3954     <xsd:sequence>
3955       <xsd:element name="AB_ID_FSPDInOut" type="dti:AB_ID_FSPDInOut_T"/>
3956       <xsd:element name="PQI" type="xsd:unsignedByte"/>
3957       <xsd:element name="OE" type="xsd:unsignedByte"/>
3958       <xsd:element name="SPDUInLength" type="dti:FSPDInOutLength_T"/>
3959       <xsd:element name="PDILength" type="dti:PDInOutLength_T"/>
3960       <xsd:element name="SPDUIn" type="dti:OctetString_T"/>
3961       <xsd:element name="PDI" type="dti:OctetString_T"/>
3962       <xsd:element name="SPDUOutLength" type="dti:FSPDInOutLength_T"/>
3963       <xsd:element name="PDOLength" type="dti:PDInOutLength_T"/>
3964       <xsd:element name="SPDUOut" type="dti:OctetString_T"/>
3965       <xsd:element name="PDO" type="dti:OctetString_T"/>
3966     </xsd:sequence>
3967   </xsd:complexType>
3968   <xsd:complexType name="AB_JobError_T">
3969     <xsd:sequence>
3970       <xsd:element name="AB_ID_JobError" type="dti:AB_ID_JobError_T"/>
3971       <xsd:element name="ExpArgBlockID" type="dti:AB_ID_ArgBlock_T"/>
3972       <xsd:element name="ErrorCode" type="dti:ErrorCode_T"/>
3973       <xsd:element name="AdditionalCode" type="dti:AdditionalCode_T"/>
3974     </xsd:sequence>
3975   </xsd:complexType>
3976 </xsd:schema>

```


F.9 Yaml file IO-Link DTI OPENAPI

The Yaml file below (see <https://yaml.org/> for more information on Yaml) contains the formal description of the REST API.

```

openapi: 3.0.0
info:
  title: IO-Link DTI REST API
  description: |-
    This describes the IO-Link REST API for communication between the Master Tool and a
    Dedicated Tool.

    Security Note: As a Master Tool developer implementing this API, restrict the REST port so
    that it will only communicate with clients on localhost/127.0.0.1. This can be done through the
    selected webserver.

    Invocation interface - pass the REST Service Port number and the DeviceInstanceId as
    command line parameters when invoking the Dedicated Tool.

    The standard HTTP status/error codes will be used by the REST Service.
    Issues in the IO-Link XML content will be handled by the error mechanism of the SMI
    Services.

    Refer to the relevant sections of Annex F or www.io-link.com for the schema definitions for
    the TPF, TBF, DTI_DTQ and DTI_MTR.

  contact:
    email: info@io-link.com
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
  version: 1.0.0
servers:
  - url: http://127.0.0.1

paths:
  /tpf/{DeviceInstanceId}:
    get:
      summary: Get TPF by DeviceInstanceId
      description: Returns a TPF in XML format for the DeviceInstanceId parameter.
      parameters:
        - name: DeviceInstanceId
          in: path
          description: DeviceInstanceId of the device
          required: true
          schema:
            type: string
      responses:
        '200':
          description: successful operation that returns a TPF in XML format
          content:
            application/xml:
              schema:
                type: object
                properties:
                  InvocationInterface:
                    type: object
        '400':
          description: Bad request
  /tbf/{DeviceInstanceId}:
    post:
      summary: posts a TBF
      description: Posts a TBF to the Master Tool REST service with updated parameters and/or
      the TechParCRC.
      parameters:
        - name: DeviceInstanceId
          in: path
          description: DeviceInstanceId of the device
          required: true
          schema:
            type: string
      requestBody:
        required: true
        content:
          application/xml:
            schema:

```



```
4052         type: object
4053         properties:
4054             ReturnInterfaceRequest:
4055                 type: object
4056     responses:
4057         '200':
4058             description: successful operation, TBF posted and response is a
4059 ReturnInterfaceResponse
4060             content:
4061                 application/xml:
4062                     schema:
4063                         type: object
4064                         properties:
4065                             ReturnInterfaceResponse:
4066                                 type: object
4067         '400':
4068             description: Bad request
4069
4070 /OnlineChannel/{DeviceInstanceId}:
4071 post:
4072     summary: Posts xml containing the Device Tool Query (DTQ).
4073     description: Post xml to the Master Tool for writing/reading parameters/process data
4074 to/from the IO-Link device.
4075     parameters:
4076     - name: DeviceInstanceId
4077       in: path
4078       description: DeviceInstanceId of the device
4079       required: true
4080       schema:
4081         type: string
4082     requestBody:
4083     content:
4084         application/xml:
4085             schema:
4086                 type: object
4087                 properties:
4088                     DTI_DTQ:
4089                         type: object
4090     responses:
4091         '200':
4092             description: successful operation, Master Tool Response (MTR) returned
4093             content:
4094                 application/xml:
4095                     schema:
4096                         type: object
4097                         properties:
4098                             DTI_MTR:
4099                                 type: object
4100         '400':
4101             description: Bad request
```

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. The following scenarios and rules apply regardless of the actual mode of the master (standard IO-Link mode (IOL_AUTOSTART or IOL_MANUAL), or SAFETYCOM).

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)	<ol style="list-style-type: none"> 1. Modify FST parameter via tool (e.g. "USB Master"; option; see off-site commissioning in IEC 61131-9) and IODD 2. Adapt FSP_TechParCRC (see 11.8.8) using "Dedicated Tool" 3. FS-Device evaluates validity of technology parameters (FST) via FSP_TechParCRC at STARTUP 4. Plug, validate & play (default)
Back-to-box to Commissioning (monitored operation) See Figure G.1	Authenticity = 0 Port = 0 FSP_TechParCRC ≠ 0 (factory settings)	<ol style="list-style-type: none"> 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. The FS-Device is powered OFF/ON (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 commissioning mode (Verification: FSP_Authenticity record and FSP_Protocol record compared; Data Storage disabled) 7. FST parameter and standard parameter can be modified, the changes take effect immediately without a power cycle. At least the FSP_TechParCRC shall not be stored persistently. The FS-Device will store the parameters persistently. 8. FS-Master Tool and FS-Master are responsible to indicate commissioning mode and/or to prevent from running in commissioning mode w/o tool connection. For this purpose, the FS-Master shall store the Port configuration with FSP_TechParCRC = "0" only in a volatile manner. Thus, after restart of the FS-Master ("power cycle") w/o tool, the FS-Device will not receive a valid FSP_VerifyRecord and will not perform safety communication. Changes of FSP-Parameter require a power cycle of the device to become active.
Commissioning	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC = 0	<ol style="list-style-type: none"> 1. FS-Master transfers FSP_VerifyRecord to FS-Device (see A.2.10 and 10.4.3.1) 2. Run in commissioning mode (Verification: FSP_Authenticity record and FSP_Protocol record compared; Data Storage disabled) 3. FST parameter and standard parameter can be modified, the changes take effect immediately without a power cycle. At least the FSP_TechParCRC shall not be stored persistently. The FS-Device will store the parameters persistently. 4. FS-Master Tool and FS-Master are responsible to indicate commissioning mode and/or to prevent from running in commissioning mode w/o tool connection. For this purpose, the FS-Master shall store the Port configuration with FSP_TechParCRC = "0" only in a volatile manner. Thus, after restart of the FS-Master ("power cycle") w/o tool, the FS-Device will not receive a valid FSP_VerifyRecord and will not

Scenario	Initial parameters	System activities
		perform safety communication. Changes of FSP-Parameter require a power cycle of the device to become active.
Commissioning to Armed and Validate See Figure G.1	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC = 0	<ol style="list-style-type: none"> 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: FSP_Authenticity record and FSP_Protocol record compared), see 11.8.6 7. Validation according to safety manual of FS-Device. 8. Standard parameter could be changed in armed mode, but no FSP or FST parameter. (FST parameter cannot be changed. Standard parameter can be changed and the changes take effect immediately without a power cycle. FSP parameter can be changed but take effect only after a power cycle.)
Armed	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC ≠ 0	<ol style="list-style-type: none"> 1. FS-Master transfers FSP_VerifyRecord to FS-Device (see A.2.10) 2. Run in armed mode (Verification: FSP_Authenticity record and FSP_Protocol record compared), see 11.8.6 3. Standard parameter could be changed in armed mode, but no FSP or FST parameter.
Armed to Commissioning	Authenticity = FSCP ("A-Code", NOTE) Port = Port number FSP_TechParCRC ≠ 0	<ol style="list-style-type: none"> 1. Set FSP_TechParCRC = 0 in FSPortConfigList for FS-Master via FS-Master Tool and FSP_TechParCRC = 0 in FS-Device. It is allowed to change the FST_TechParCRC to 0 together with a possible change of FSP_Watchdog and the according FSP_ProtParCRC. These values shall be updated in the FSP_PortConfiguration and stored temporarily in the FSPortConfiguration of the FS-Master port. The FS-Device will store the parameters persistently. 2. Follow activity 4. to activity 8. in scenario "Commissioning" <p>NOTE: The change of the FSPortConfiguration will lead to a PowerCycle of the FS-Device which will activate these parameters.</p>
Replacement by FS-Device with factory settings w/o tools	Authenticity = 0 Port = 0 FSP_TechParCRC ≠ 0	<ol style="list-style-type: none"> 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: FSP_Authenticity record and FSP_Protocol record compared), see 11.8.6 and A.2.10 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	<ol style="list-style-type: none"> 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: FSP_Authenticity record and FSP_Protocol record compared), see 11.8.6 and A.2.10 3. Error message: "Misconnection" (0xB003 or 0xB004, see Annex B).
Connection to a master in standard IO-Link mode (without transmission of FSP_Verify-Record)	Don't care	<ol style="list-style-type: none"> 1. Parameter rules do not depend on the IO-Link mode. For this reason, all of the above rules apply, and parameter can be changed under these restrictions.
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.

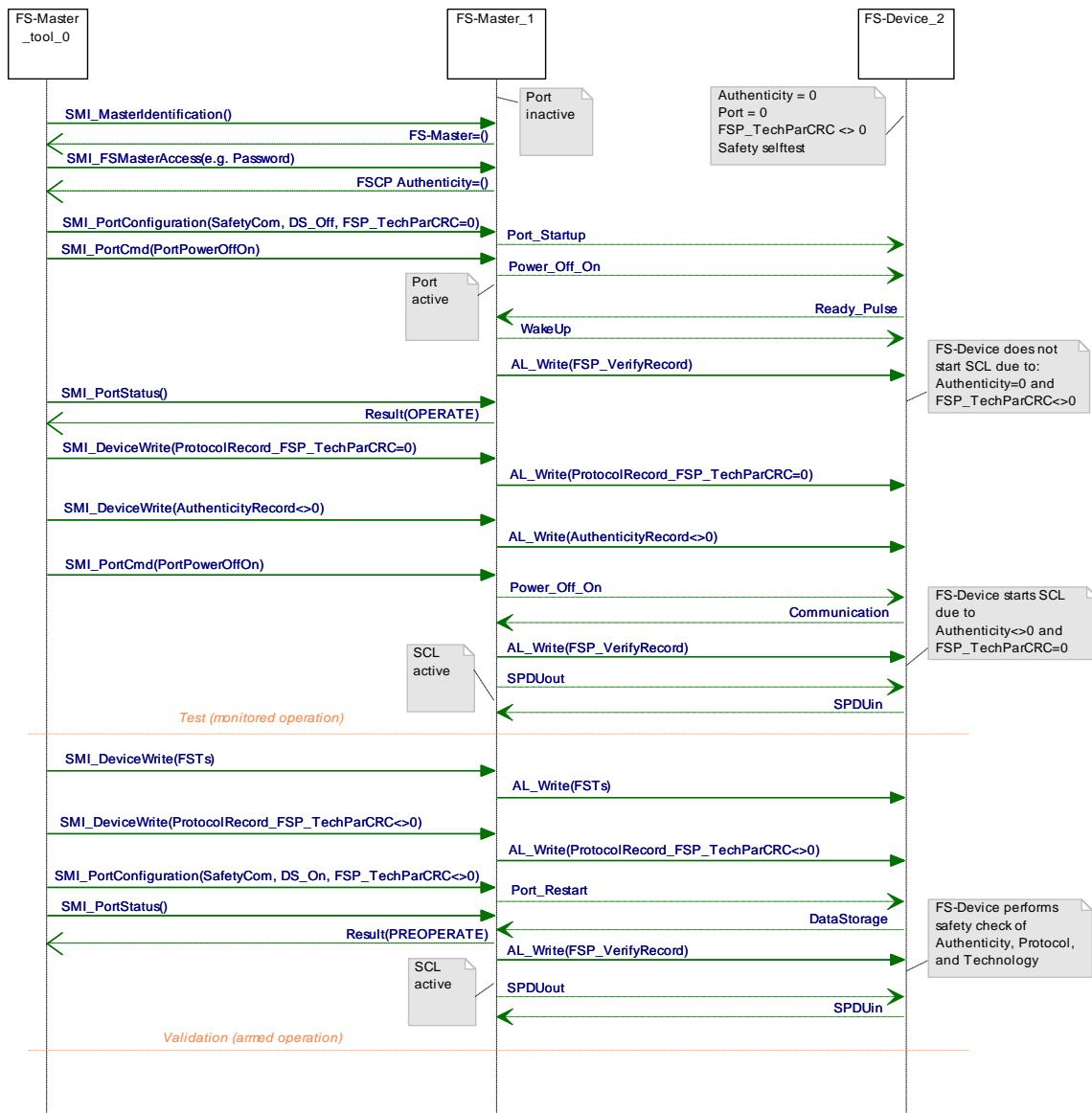


Figure G.1 – Commissioning with test and armed operation

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.

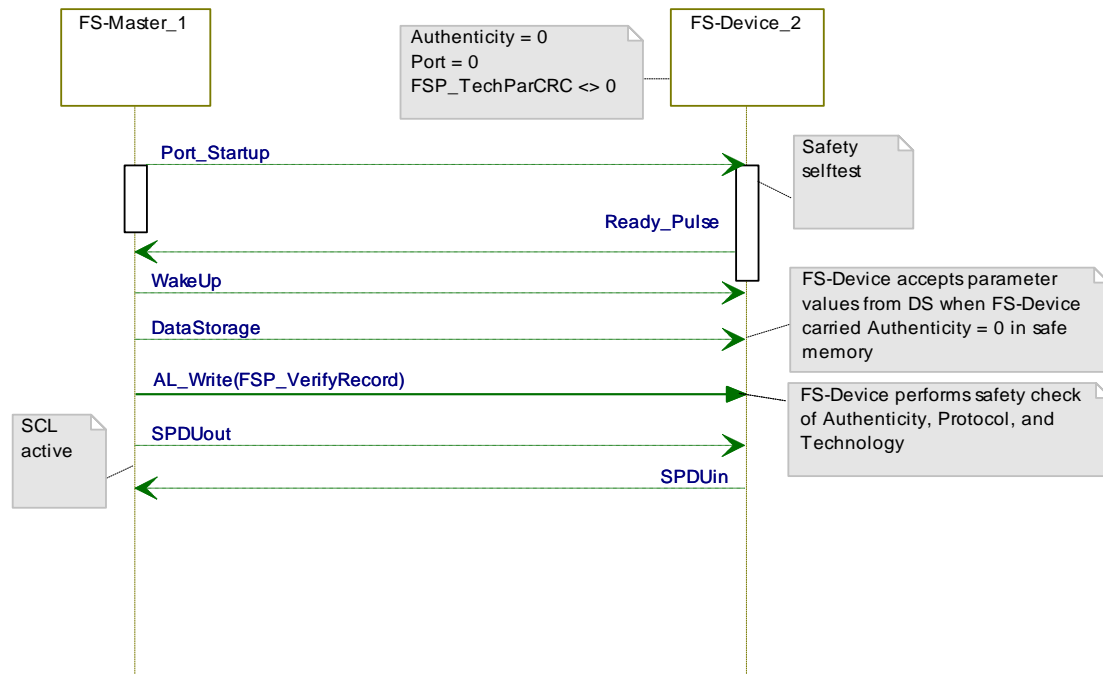


Figure G.2 – FS-Device replacement

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.

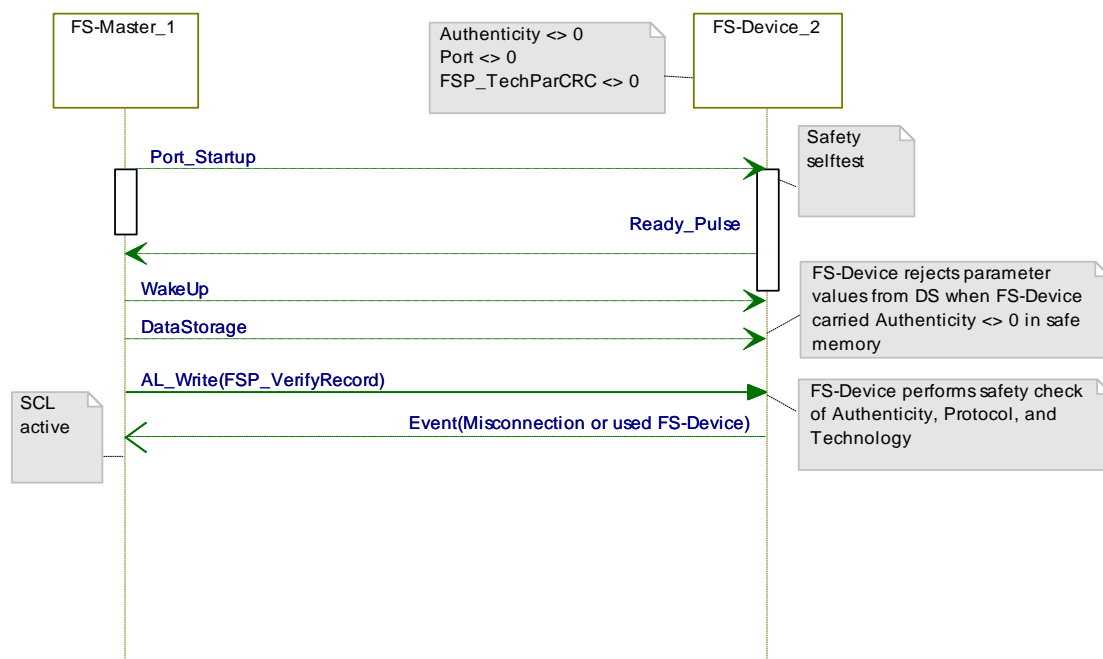


Figure G.3 – FS-Device misconnection

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 FS-DI

In case an FS-Master Port is running in FS-DI 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 FS-Master Tool

In case an FS-Device/FS-Master port runs in commissioning mode, the FS-Master Tool shall show this.

H.1.5 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 11.12.4 and 11.12.5. 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 [19]).

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.

4174 The zones and conduit concept of IEC 62443 applies for security and/or the rules of the
4175 applicable FSCP system.

4176 **H.3 Safety function response time**

4177 Safety manuals of FS-Master shall provide information on how to determine the safety function
4178 response time for FS-DI and for communication modes (see Clause H.6).

4179 **H.4 Duration of demands**

4180 Short demands of FS-Devices may not trip a safety function due to its chain of independent
4181 communication cycles across the network. Therefore, a demand shall last for at least two SCL
4182 (SPDU) cycles.

4183 **H.5 Maintenance and repair**

4184 FS-Devices can be replaced at runtime. Restart of the corresponding safety function is only
4185 permitted if there is no hazardous process state, after validation of the safety function(s), and
4186 after an operator acknowledgment.

4187 **H.6 Safety manual**

4188 FS-Masters and FS-Devices shall provide safety manuals according to the relevant national
4189 and international standards, for example IEC 61784-3:2021.

4190 Manufacturer/vendor of FS-Masters and/or FS-Devices shall specify appropriate mitigation
4191 means in the safety manual for the deployment of SDCI-FS components in harsh industrial
4192 environment such as in EMC zones B and C according to IEC 61131-2.

4193 Manufacturer/vendor of FS-Masters and/or FS-Devices shall define all constraints for the
4194 operation of OSSD devices or FS-Devices in OSSDe mode within their safety manuals.

4195 Manufacturer/vendor of FS-Masters and/or FS-Devices shall define all constraints for the
4196 operation of FS-Devices in communication mode within their safety manuals such as limitations
4197 with respect to storing elements, inductive or optical couplers, and alike.

4198 Manufacturer/vendor of FS-Masters and/or FS-Devices shall define the maintenance rules with
4199 respect to the PFH-Monitor (see Table 40).

4200 Manufacturer/vendor of FS-Devices shall provide the "worst case delay time" (WCDT) value.

4201 Manufacturer/vendor of FS-Devices shall provide the "one fault delay time" (OFDT) value.

4202 Manufacturer/vendor of FS-Masters shall provide information on how to determine the safety
4203 function response time as specified in IEC 61784-3:2021 using WCDTs and considering
4204 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
Ohiostraße 8
76149 Karlsruhe
GERMANY

Phone: +49 721 9861 970
Fax: +49 721 9861 9711
E-Mail: info@io-link.com
URL: <https://www.io-link.com/>

Bibliography

- 4223
- 4224 [1] ISO/IEC 19505-2:2012, *Information technology – Object Management Group Unified*
4225 *Modeling Language (OMG UML) – Part 2: Superstructure*
- 4226 [2] Bruce P. Douglass, *Real Time UML – Advances in the UML for Real-Time Systems*, 3rd
4227 Edition, Addison-Wesley, ISBN 0-321-16076-2
- 4228 [3] Chris Rupp, Stefan Queins, die SOPHISTen, *UML 2 glasklar – Praxiswissen für die UML-*
4229 *Modellierung*. Hanser-Verlag, 2012, ISBN 978-3-446-43057-0
- 4230 [4] IEC/TR 62390, *Common Automation Device – Profile Guideline*
- 4231 [5] IEC 60947-5-5, *Low-voltage switchgear and controlgear – Part 5-5: Control circuit*
4232 *devices and switching elements – Electrical emergency stop device with mechanical*
4233 *latching function*
- 4234 [6] ZVEI|RECOMMENDATION 2022.01 Position Paper CB24I, Classification of Binary 24V
4235 Interfaces – Functional Safety aspects covered by dynamic testing:
4236 https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2022/Ja
4237 [nuar/24_V_INTERFACES/24V-Interfaces.pdf](https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2022/Ja)
- 4238 [7] IEC 60947-5-2, *Low-voltage switchgear and controlgear – Part 5-2: Control circuit*
4239 *devices and switching elements – Proximity switches*
- 4240 [8] IO-Link Community, *IO-Link Safety (Single Platform) – Requirements, Use Cases, and*
4241 *Concept Baseline*, V1.0, November 2014, Order No. 10.062
- 4242 [9] IEC 62769 series: *Field Device Integration (FDI)*
- 4243 [10] FDT Joint Interest Group, *FDT 2.0 – Specification*, V1.0, Order No. 0001-0008-000
- 4244 [11] IEC 60947-5-6, *Low-voltage switchgear and controlgear – Part 5-6: Control circuit*
4245 *devices and switching elements – DC interface for proximity sensors and switching*
4246 *amplifiers (NAMUR)*
- 4247 [12] Philip Koopman, *Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded*
4248 *Networks*, The International Conference on Dependable Systems and Networks, DSN-
4249 2004
- 4250 [13] Philip Koopman, Best CRC Polynomials, <https://users.ece.cmu.edu/~koopman/crc/>
4251 (Date: 29-Jan-2017)
- 4252 [14] CRC signature calculator for a seed value of "0":
4253 <https://www.ghsi.de/pages/subpages/Online%20CRC%20Calculation/index.php?Polyno>
4254 [5339%20m=10100111010101011&Message=1%0D%0A](https://www.ghsi.de/pages/subpages/Online%20CRC%20Calculation/index.php?Polyno) (Date: 05-Apr-2018)
- 4255 [15] IO-Link Community, *IO Device Description (IODD)*, V1.1.5, October 2025, Order No.
4256 10.012
- 4257 [16] IO-Link Community, *IO-Link Common Profile*, V1.2.1, October 2025, Order No. 10.072
- 4258 [17] IEC 62453 series: *Field device tool (FDT) interface specification*
- 4259 [18] FDT Joint Interest Group, *FDT for IO-Link – Annex to FDT Specification – Based on FDT*
4260 *Specification Version 1.2.1*, V1.0, Order No. 0002-0013-000
- 4261 [19] IO-Link Community, *IO-Link Design Guideline*, V1.0, November 2016, Order No. 10.912
- 4262 [20] IEC 60947-5-3, *Low-voltage switchgear and controlgear – Part 5-3: Control circuit*
4263 *devices and switching elements – Requirements for proximity devices with defined*
4264 *behavior under fault conditions*

- 4265 [21] IEC 61076-2-113, *Connectors for electronic equipment – Product requirements – Part*
4266 *2-113: Circular connectors – Detail specification for connectors with M12 screw locking*
4267 *with power and signal contacts for data transmission with frequency up to 100 MHz*
- 4268 [22] IEC 61508-4:2010, *Functional safety of electrical/electronic/programmable electronic*
4269 *safety-related systems – Part 4: Definitions and abbreviations*
- 4270 [23] ISO 14119:2013, *Safety of machinery – Interlocking devices associated with guards –*
4271 *Principles for design and selection*
- 4272 [24] IEC 61131-9:2022, *Programmable controllers – Part 9: Single-drop digital*
4273 *communication interface for small sensors and actuators (SDCI)*
- 4274 [25] IO-Link Community, *IO-Link Interface and System*, V1.1.5, October 2025, Order No.
4275 10.002
- 4276 [26] PI Profile Specification, *IO-Link Safety Integration in PROFI-safe*, V1.0, February 2022,
4277 Order No. 3.312
- 4278 [27] IEC 61784-3:2021/AMD1:2024, Amendment 1 - Industrial communication networks -
4279 Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions

4280

© Copyright by:

IO-Link Community

Ohiostraße 8

76149 Karlsruhe

Germany

Phone: +49 (0) 721 / 98 61 97 0

Fax: +49 (0) 721 / 98 61 97 11

e-mail: info@io-link.com

<http://www.io-link.com/>



IO-Link